A Demonstration of 4D Digital Avatar Infrastructure for Access of Complete Patient Information

Project acronym: MyHealthAvatar

Deliverable No. D3.7 v2.0
Integrated Platform with an evaluation report

Grant agreement no: 600929

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	MyHealthAvatar
Project Full Name:	A Demonstration of 4D Digital Avatar Infrastructure for Access of Complete Patient Information
Deliverable No.:	D3.7
Document name:	Integrated platform with an evaluation report
Nature (R, P, D, O)[1]	R
Dissemination Level (PU, PP, RE, CO)[2]	PU
Version:	2.0
Actual Submission Date:	2/06/2016
Editor: Institution: E-Mail:	Manolis Tsiknakis TEI-C tsiknaki@ie.teicrete.gr

[1] **R**=Report, **P**=Prototype, **D**=Demonstrator, **O**=Other

[2] **PU**=Public, **PP**=Restricted to other programme participants (including the Commission Services), **RE**=Restricted to a group specified by the consortium (including the Commission Services), **CO**=Confidential, only for members of the consortium (including the Commission Services)

ABSTRACT:

This deliverable provide information about the integration of tools and services that support the 4D digital avatar. More specifically, the infrastructure needs to support the integration of data/model utilities, the ICT toolboxes and the model/data repository, and demonstrate capacities of linking towards the external data resource. The integration will provide a front-end web interface for users to access the system, e.g. use of the avatar toolboxes; and a back-end workflow mechanism that manages data access and storage, model processing, and model composition. The integrated platform pays special attention to the systems that reside in different geographic area and use different implementation platforms, which mimics the real systems deployments. Also, links to computing clusters and supercomputing facilities are also integrated to support the computing demands of the avatar, e.g. running of the simulation models.

KEYWORD LIST:

Integration, API, User Requirements, Standards, Guidelines, Protocols, IT, State-Of-the-Art

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement ° 600929.

MODIFICATION CONTROL			
Version	Date	Status	Author
1.0	06/10/2015	Draft	Manolis Tsiknakis
1.1	05/12/2015	Draft	Stelios Sfakianakis, Kostas Marias
1.2	05/01/2016	Draft	Emmanouil G. Spanakis, Stelios Sfakianakis
1.4	25/01/2016	Draft	Feng Dong, Nikolaus Forgo, Sarah Jensen, Zhikun Deng, Haridimos Kondilakis, Nikolaos Christodoulou, Eleni Georgiadi, Nikolaos Tousert, Georgios Stamatakos
1.0	15/02/2016	Final	Manolis Tsiknakis , Emmanouil G. Spanakis, Kostas Marias
2.0	2/06/2016	Final	Manolis Tsiknakis , Emmanouil G. Spanakis, Kostas Marias, Feng Dong, Stelios Sfakianakis, Nikolaos Christodoulou, Haridimos Kondylakis

List of contributors

– Tsiknakis Manolis (TEI-CRETE)
– N. Bidakis (TEI-CRETE)
– V. Fragopoulou (TEI-CRETE)
– Emmanouil G. Spanakis (FORTH-ICS)
– Haridimos Kondilakis (FORTH-ICS)
– Kostas Marias (FORTH-ICS)
– Stelios Sfakianakis (FORTH-ICS)
– Nikolaus Forgó (LUH)
– Feng Dong (BED)
– Sarah Jensen (LUH)
– Zhikun Deng (BED)
– Nikolaos Christodoulou (ICCS-NTUA)
– Eleni Georgiadi (ICCS-NTUA)
– Nikolaos Tousert (ICCS-NTUA)
– Georgios Stamatakos(ICCS-NTUA)

Contents

Table of Contents

1 Executive Summary

MyHealthAvatar proposes a solution for access, collection and sharing of long term and consistent personal health status data through an integrated environment, which will allow more sophisticated clinical data analysis, prediction, prevention and *in silico* treatment simulations tailored to the individual citizen. The ambitious work programme of MyHealthAvatar presents great challenges from the technological point of view in order for the produced system to be a coherent set of interoperable software components. Therefore the aim of this deliverable is to specify the guidelines that are needed for the integration of all MyHealthAvatar architectural elements into an integrated, coherent and consistent platform. In addition, we describe the methodologies and the tools that are required to monitor the specific architectural components and the integration process as a whole.

This deliverable elaborates on the MyHealthAvatar architecture that is defined in deliverable D3.2 and presents the integration status of the platform using the select user scenarios and applications. As stated in the Description of Work *these guidelines will be created based on the integration experiences and platform evolution based on the architecture definition of the platform*. MHA architecture description is refined on an iterative approach with annual updates and revisions thus this deliverable present the final view of the platform design and final integration of MHA platform.

In the rest of this document we outline the relevant key points of the MyHealthAvatar architecture, we examine practices and guidelines from related projects regarding integration procedures, we elaborate on the quality management guidelines and how they affect the integration process, we present a certification process to monitor and measure the compliance of architectural elements to the integration guidelines, we propose best practices for the development of REST web services and we propose guidelines on integrating the architectural elements with the privacy framework of p-medicine.

Finally we present the results of the evaluation process of the high level functionalities and applications of the platform both from the end-user point of view and from the technical perspective.

2 Introduction

2.1 *Purpose of this document*

The purpose of the MyHealthAvatar project is to deliver an IT infrastructure, both as an open architecture and as a set of architectural elements which realize this architecture. This infrastructure enables and enhances the provision of personalized medical treatment, at all phases such as diagnosis, treatment, patient empowerment, clinical decision support, etc. In order to be able to provide MHA services are defined and implemented as a modular, extensible integrated architectural platform able to adapt and incorporate the advances in science, cover the interactions between different types of stakeholders and scientists, integrate health and health related data, to actually provide a useful platform and not just a set of isolated and independent tools and services which do not cooperate and interact as much as it would be desirable. The purpose of this document is to provide the integration procedure and implementation from each developer of all architectural elements, in order to deliver solid components that will be able to be integrated smoothly with the rest of the architectural elements while taking into account the constraints and the recommendations set by the reference architecture.

3 MyHealthAvatar Integration Methodology

The methods for integration provides the required steps for integration & validation of informal software engineering building blocks. The software design and development team and test engineers developed a strategy for planning, design, execution, data collection, and testing for quality evaluation. The integration activities are informal and flexible for software checkout to prepare for the software and systems integration phase for the work product. MHA methodology for integration describes the steps that were conducted as part of the implementation of necessary software elements and integration activities to implement the final MHA system platform. The methodology was flexible and promote an approach were all relevant partners conducted effective technical reviews in each different step of the implementation and the final step of integration, presented different integration techniques and software approaches through designers involved from the start to the finish. The following picture presents a schematic of our approach. The technical manager with the WP3 leader (responsible for the architecture design) and the coordinator guided the process through the project. Below we analyse the most important issues for integration activities followed.

3.1 Design

External software interfaces are defined as part of derived software requirements. To support systems design, graphical representations are prepared and take the form of data flow, collaboration and communications, and component diagrams. Proper systems design is needed by the product team and the requirements personnel that works with users to ensure an accurate and complete understanding of the restrictions of a system or subsystem that affects work products. Systems design allows you to analyse customer requirements, satisfy specified requirements, and develop a software design and development migration plan for defining the architecture, components, modules, interfaces, and necessary data[3]. This is important for communication, knowledge, the visibility into the software life-cycle, and integration operations. The system and subsystem requirements reviewed by program and project personal ensure accurate and complete understanding of the restrictions of systems design and applied work products. If program or project plans include reusable software interfaces, you can identify and evaluate the requirements. The term “reusable software” is a common term used in military and aerospace programs and/or projects. External software interfaces are defined as part of derived software requirements. To support systems design, graphical representations are prepared and take the form of data flow, collaboration and communications, and component diagrams. The requirements for a system design definition are reviewed with applicable users to ensure an accurate and complete understanding of the restrictions of a system or subsystem that affects work products. The external software interface is defined at those levels and verified for completeness. The program and project plans at times include reusable software and identified

[3] Summers BL. Effective methods for software and systems integration. CRC Press; 2012 Jun 1.

interface requirements for use. The external interfaces based on software architecture definitions are part of derived software requirements also.

3.2 Requirements

Defined and documented software requirements provide a systematic approach to developing software requirements derived from multiple resources. The combination of functional software interfaces, performance, verification, and production with required plans, documentation, and procedures make up the basis for software design or development. This discipline can be applied to the initial development of software requirements and any changes to requirement baselines.

3.3 Development

System development is a consistent approach and method to the development of software requirements in defined designs of a work product. The software architecture definition provides a framework for the creation of the product design and, at times, can provide constrictions. The software design definition implements details about a software product architecture, components, and interfaces. Software designers use element traceability of the design and the software requirements are used by software designers. The traceability data and software design definitions are documented according to program and project plans, ideas, processes, procedures, and applicable internal work instructions.

3.4 Quality metrics

The ISO (International organization for standardization, http://www.iso.org/iso/home.html) SQuaRE (Software product Quality Requirements and Evaluation) [4], will be used as reference model; it lists standards in terms of: General Guidance: ISO/IEC 25000, Particular Guidance: ISO/IEC 25040 (ISO/IEC 9126-1 and ISO/IEC 14598-1) and Execution: ISO/IEC 25041 (ISO/IEC 14598-6), ISO/IEC 25042 (ISO/IEC 14598-3), ISO/IEC 25043 (ISO/IEC 14598-4).

We point out the different point of view for quality evaluation: Evaluation process for developers (ISO/IEC 14598-3), to use when the evaluation is conducted in parallel with the development and Evaluation process for acquirers (ISO/IEC 14598-4), to use during modifications to existing software products. The general reference is contained in the evaluation reference model and guide (ISO/IEC 9126-1 and 14598-1) and structure and content of the documentation to be used is listed in (ISO/IEC 14598-6).

User needs specify also the required level of quality from the end user point of view. The defined requirements have to be seen from an external and an internal view as defined below:

[4] ISO/IEC 25000:2005 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE. Geneva, Switzerland: ISO/IEC.

- External SW quality requirements: are used as the target for technical verification and validation of the software product
- Internal SW quality requirements: quality from the internal view of the product, they are used to specify properties of intermediate software products.

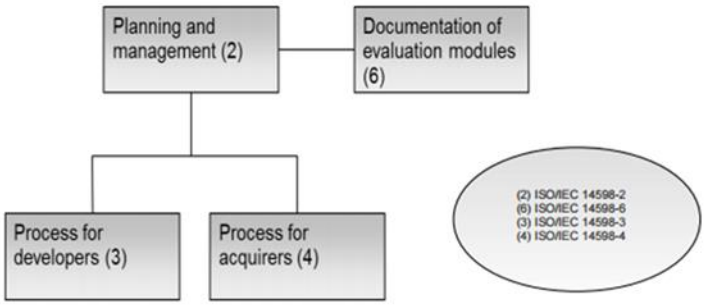

Planning and management (2)
Documentation of evaluation modules (6)
Process for developers (3)
Process for acquirers (4)
(2) ISO/IEC 14598-2
(6) ISO/IEC 14598-6
(3) ISO/IEC 14598-3
(4) ISO/IEC 14598-4

Figure 3-1: Adapted from ISO/IEC 14598 modules organization.
The numbers in the round brackets represent the modules: i.e. if (2), then ISO/IEC 14598-2, etc.

To assess quality levels, the end user and/or evaluator have to receive a list of metrics that she/he can measure. Internal metrics are associated to the software product architecture and allow to predict the final product quality; external metrics are measurable when the product is under operation. The ISO standard identifies major quality characteristics, each one having some sub-properties which specialize in more depth that characteristic. Below we outline their essential purpose and properties[5].

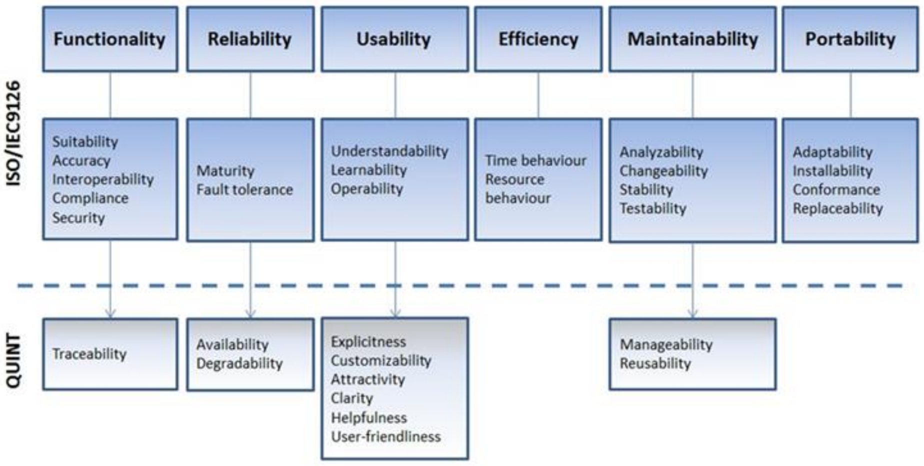


Figure 3-2: ISO/IEC 9126 metrics and the QUINT extension.

3.4.1.1 Functionality

Functionality is the core property of a system, which summarizes the essential functions that the system must provide. The functionality depends on the requirements from that system and if the

[5] http://www.sqa.net/iso9126.html

system is not functional, then the rest of the characteristics, such as usability or reliability, do not really matter. Functionality is measured by the degree of conformance to the user requirements, and is better expressed by decomposing it to the following properties:

- **Suitability**. The appropriateness of the functions to the specification of the system **Accuracy**. The correctness of the functions.
- **Interoperability**. The ability of a system or component to interact with other systems or components.
- **Compliance**. The compliance with laws or guidelines, when such a requirement exists.
- **Security**. The ability to prevent unauthorized access or intervention with the system.
- **Traceability**. The ability to keep trace, for auditing purposes, of the operation of the system.

3.4.1.2 Reliability

Once the system is *functional* then the reliability characteristic defines and measures the capability of the system to maintain its functionality under certain conditions. To further analyze the reliability of the system, we define the following sub-characteristics:

- **Maturity**. The frequency of failure of the system.
- **Fault tolerance**. The ability to withstand a failure.
- **Recoverability**. The ability to recover a failed system to fill operation.
- **Availability**. The ability to maintain a prearranged level of operational performance. **Degradability**. The ability to continue operating in case of a failure, with a graceful decrease of operation proportional to the severity of failure.

3.4.1.3 Usability

Similar to the reliability characteristic, the usability of a system is meaningful once the system is functional and it refers to the easiness to learn and perform a function. The sub-characteristics of usability are:

- **Understandability**. The easiness to understand the system functions.
- **Learnability**. Learning effort required (for different users, such as novice, expert etc.)
- **Operability**. The ability to operate the system by a given user in a given environment.

3.4.1.4 Efficiency

The efficiency of a system concerns the needed resources to provide the required functionality, such as memory, storage space, network, etc. The efficiency characteristic many times influences the usability of a system. Sub-characteristics of efficiency are:

- **Time behavior**. The response time to a given action in certain conditions.
- **Resource behavior**. The resources used in certain conditions.

3.4.1.5 Portability

The portability of a system refers to its ability to adapt to changes in its operational environment or its requirements. Sub-characteristics of portability are:

- **Adaptability**. The ability of the system to change to new specifications or environments.
- **Installability**. The effort required to install the system.
- **Conformance**. A characteristic similar to compliance (functionality) in relation to portability.
- **Replaceability**. How easy it is to exchange or replace a specific component by another in a specified environment.

3.4.1.6 Maintainability

The maintainability of a system refers to its ability to keep performing and the effort required to fix, change, test, reuse it etc. Sub-characteristics of maintainability are:

- **Analyzability**. The ability to identify the cause of a failure in the software.
- **Changeability**. The effort required to change the system.
- **Stability**. The sensitivity to change of a system and the degree of negative impact that a change may cause.
- **Testability**. The effort required to verify a system change.
- **Manageability**. The effort required to monitor the system and keep it performing.
- **Reusability**. The suitability of the system to be used again under slight or no modifications.

For each characteristic listed above, an appropriate measure which quantifies the corresponding metric must be specified. For some of these metrics, a numerical or quantifiable measure is harder to be applied than others because of their subjective nature. For this matter, an alternative scale has been proposed by the ISO standard where each metric can be compared with a minimum level or target range which sets the measured quality as acceptable or not.

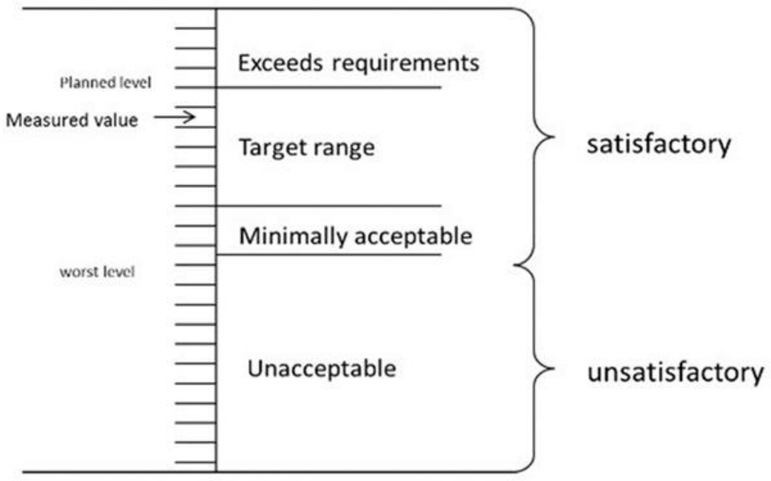


Figure 3-3: Degrees of satisfaction and interpretation (ISO/IEC)

It is a hard task to set the satisfactory on unsatisfactory ranges of satisfaction for each architectural element (AE) individually, and to measure them for all of the characteristics listed above, since that would require close examination of the requirements of all components. So, what we propose is to

require for each component to provide test clients and appropriate source code or test scripts, which will enable developers and users to examine the above characteristics by their own.
Monitoring of services

3.4.1.7 Testing clients

An essential part of the integration process is the monitoring of all components. Monitoring involves continuous tests in many different aspects such as stability, reliability, fault tolerance, efficiency, security etc. In our view, most of these aspects are covered by the corresponding key domains of the ISO 9126 metrics and its extensions. Thus, monitoring of the tools and services of MyHealthAvatar is a process embedded in the quality management of the whole project and can be part of the same procedure.

One of the quality management metrics of ISO 9126 is *testability*. There are various artefacts required during the evaluation of the compatibility and interoperability of a system/component. These artefacts include also test scripts, source code or example clients for the evaluated components, along with any other relevant documentation. We propose that the same kind of artefacts be required for the integration process of a various MyHealthAvatar component into the platform for testability reasons.

Each component provider is responsible to provide the testing client, either in the form of source code snippets or a complete client, depending on the nature of the evaluated component. The testing clients can be provided in whatever format or language is considered appropriate by their developers, but for interoperability and reusability reasons we strongly recommend that the testing clients be in a form that can be easily integrated, without major revisions, to the MyHealthAvatar portal.

These clients will be used to support many different needs, both during and after the integration process:

- To evaluate the functionality of a component during the integration process with other components.
- To validate the conformance to protocols, standards or integration guidelines.
- To validate its compliance with the security guidelines.
- To evaluate the component in terms of efficiency and performance.
- To evaluate and monitor the component in terms of stability, reliability and fault tolerance. To be used as exemplar use case while integrating with another component or service.

3.4.1.8 Testbed

For the needs of the integration process and for the monitoring of the components, we shall set-up a test-bed infrastructure. This test-bed shall be composed either from physical or virtual (based on virtualization software) computers, to provide for each use case the necessary environment and configurations such as operating system, libraries, databases, firewalls, security mechanisms, network configuration etc.

The need for a test-bed results from the distributed nature of the participating components. Although the development of the MyHealthAvatar components is distributed, due to the geographical distribution of the involved partners, there is the need to test the components in a deterministic process which can better be supported by a centrally installed system.

By using a test-bed:

- We shall reproduce and verify the documentation of a component, as to what are the necessary installation and operation environment, and the step-by-step installation procedure.
- We shall use it as a Sandbox where we can test and validate the security of the components, in terms of compliance with the security guidelines.
- We shall document the necessary steps, from installation to operation and maintenance, of a use case.
- We shall replicate and demonstrate a use case / scenario.
- We shall use it to verify interconnectivity, interoperability and compliance between different architectural components.
- We shall reproduce and debug reported errors or problems.

In order to set-up the test-bed, each component shall clearly state in its documentation its requirements and its dependencies, such as the needed operating system, runtime environment, network configuration, etc. with specific versions or releases with which it has been tested. The private cloud of MHA has provided the test bed facilities. All details are presented in deliverables D3.2, and D3.2 version 2.0 and shortly described in the Integration environment/infrastructure of this deliverable.

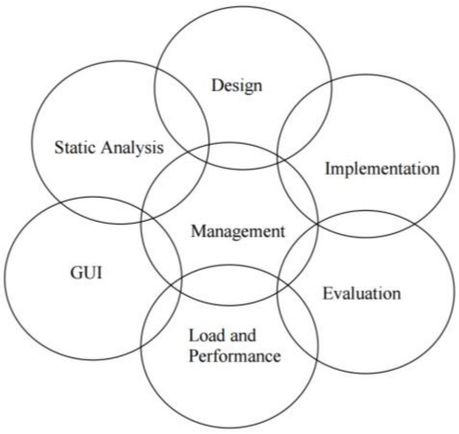
3.5 Platform testing methodologies

In this section we present the integration approach and the software testing methodologies for the MyHealthAvatar platform. Typically the integration process deals with two distinct topics: the integration tests and the results showing the outcome of the execution of the tests during integration. Software testing is conducted to provide and objective, independent view about the quality of the product or service under test. Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test: meets the requirements that guided its design and development, responds correctly to all kinds of inputs, performs its functions within an acceptable time, is sufficiently usable, can be installed and run in its intended environments, and achieves the general result its stakeholders desire. As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources. As a result, software testing typically (but not exclusively) attempts to execute a program or application with the intent of finding bugs errors or other defects. The job of testing is an iterative process as when one bug is fixed, it can illuminate other, deeper bugs, or can even create new ones.

Software testing management tools are used to store information on how testing is to be done, plan testing activities and report the status of quality assurance activities. The tools have different approaches and thus have different sets of features. Generally they are used to maintain and plan manual testing, run or gather execution data from automated tests, manage multiple environments and to enter information about found defects. These tools offer the prospect of streamlining the testing process and allow quick access to data analysis, collaborative tools and easy communication across multiple project teams. Many test management tools incorporate requirements management capabilities to streamline test case design from the requirements. Tracking of defects and project tasks are done within one application to further simplify the testing. Test Management encompasses anything and everything that we do as testers: creating and maintaining release /project cycle /component information, test artifacts specific to each release /cycle; establishing traceability and coverage between the test assets; test execution support – test suite creation, test execution status capture, etc; metric collection/report-graph generation for analysis; and finally bug tracking/defect management. The above are broadly some of the tasks that involve what we call, the test management process. This process is critical, detail-oriented and instrumental is making sure that the entire testing effort is successful.

Some of the most well known and used management tools are listed below (this is an indicative list and we have to acknowledge that much more management tools – designed for specific requirements exist[6]):

- **Atera** Remote Monitoring Management (RMM), Professional Service Automation (PSA) and remote control - customer happiness in one place.
- **Telecommunications Management** Calero is a leading provider of IT Telecom Management solutions with a deep commitment to innovation & customer service.
- **JIRA Service Desk** Redefine what IT means for your business with JIRA Service Desk
- **Desktop Central** Helps administrators to automate, standardize, secure, and audit their windows network. Desktop Central now supports MDM also.



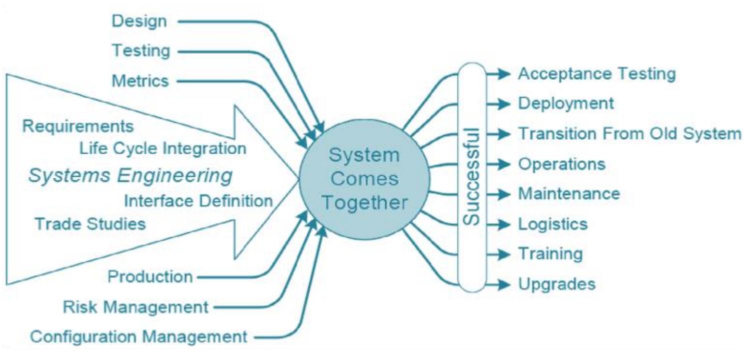
An extensive comparison of various similar tools can be found here[7]. This study divides and categorizes a big number of tools based on Design, GUI (Graphical User Interface), Load and Performance, Management, Implementation, Evaluation, Static Analysis and outside of inspection: Defect Tracking, Web Sites and Miscellaneous.

3.5.1 Software and platform testing

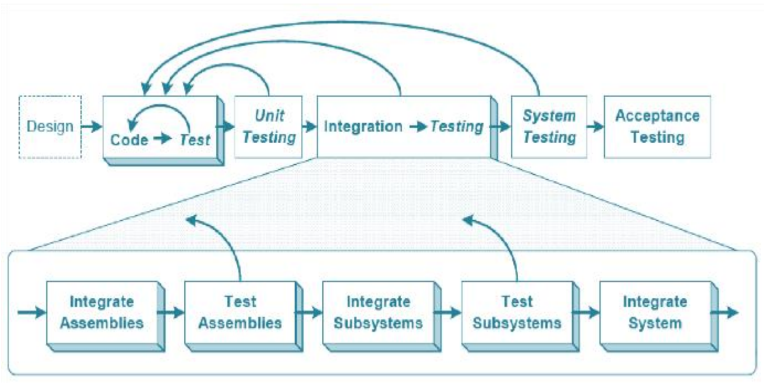
A definition of software testing can be given as "*the process that is followed in order to verify that the system being tested conforms to the requirements that have been specified*". We followed several testing phases during the development of the final platform: unit and component testing, integration testing and system (or user acceptance) testing and evaluation. Unit and component tests mainly focus on functional testing, finding and eliminating bugs in the components that comprise the system. The final integration was the process of successfully putting together various components, assemblies, and subsystems to assemble all different resources in to a complex system/platform making sure that all functionalities are operating successfully. System integration in MyHelathAvatar followed the coding/implementation phase in the development life cycle, starting with the requirement gathering phase, the design the implementation and unit test, the *integration and testing* of the platform, the user validation and the deployment of the final solution.

[6] http://www.softwaretestinghelp.com/15-best-test-management-tools-for-software-testers/
[7] http://www.cs.uef.fi/tutkimus/Teho/SoftwareTestingTools.pdf

Integration tests usually deal with functional and non-functional system behaviour identifying problems in the relationships and interfaces between incrementally integrated components. Finally, acceptance tests pay explicit attention in conformance to requirements at the fully integrated system. This section focuses mainly on unit and integration tests and user acceptance tests. Since different components are developed by different partners, integration plays an important role as each partner is able to test only the individual component the partner is involved in, but not the integrated backend system. Some typical component tests are carried out to verify that the various components have the expected functionality, but component testing is considered to have already been done by the developer of each component. Unit tests are not covered in the test plan at all, since they are specifically valuable only during implementation (in fact, in agile software development methodologies unit testing is at the centre of the implementation philosophy, which is thus appropriately named "test-driven development").



The diagram on the right explains the plans for the integration in conjunction with the testing in detail. Integration testing within MHA was used to assure developers that the integrated product is properly functional. Integrated modules that fail testing are sent back for debugging and rework.



During the final year of the project all tests results were analyzed to determine erroneous situations components/services and software interfaces in order to be sent back for recoding by the developing teams. The technical manager and WP3 leader were responsible for this. If a problem appears with the design, the modules involved are sent back for redesign, in order to resolve the problem. This process loop continued until all problems are solved. After rework, all lower-level integration and test cycles are repeated prior to repeating the higher-level integration. This is part of the regression testing process and is used to detect any new errors that may slip in due to "fixing" other problems.

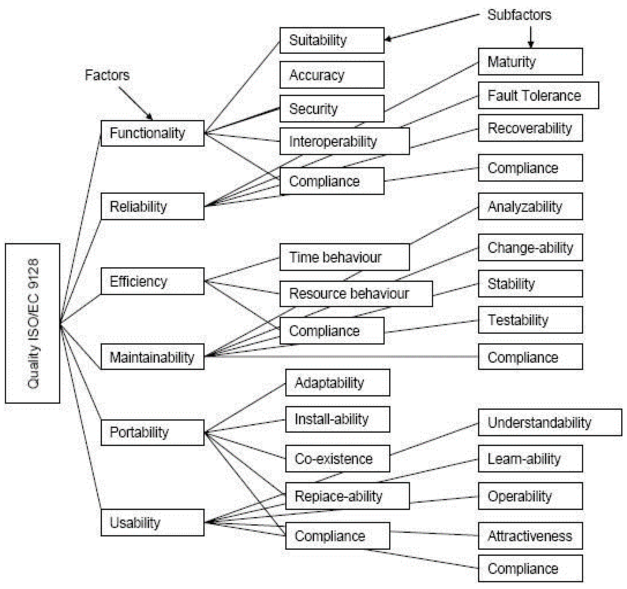
3.5.2 Software Testing and ISO Standards

Testing is the process of evaluating a system or its component(s) with the intent to find that whether it satisfies the specified requirements or not. This activity results in the actual, expected and difference between their results. In simple words testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements. According to ANSI/IEEE 1059 standard, Testing can be defined as “A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item”. Many organizations around the globe are developing and implementing different Standards to improve the quality needs of their Software.

The next section briefly describes some of the widely used standards related to Quality Assurance and Testing. Here is a definition of some of them:

ISO/IEC 9126: This standard deals with the following aspects to determine the quality of a software application: Quality model; External metrics; internal metrics; Quality in use metrics.

This standard presents some set of quality attributes for any Software such as: Functionality; Reliability; Usability; Efficiency; Maintainability; and Portability. These quality attributes are further divided into sub-factors. These sub characteristics can be measured by internal or external metrics as shown in the graphical depiction on the right of ISO-9126 model.



ISO/IEC 9241-11: Part 11 of this standard deals with the extent to which a product can be used by specified users to achieve specified goals with Effectiveness, Efficiency and Satisfaction in a specified context of use. This standard proposed a framework which describes the usability components and relationship between them. In this standard usability is considered in terms of user performance and satisfaction. According to ISO 9241-11 usability depends on context of use and the level of usability will change as the context changes.

ISO/IEC 25000: ISO/IEC 25000:2005 is commonly known as the standard which gives the guidelines for Software product Quality Requirements and Evaluation (SQuaRE). This standard helps in organizing and enhancing the process related to Software quality requirements and their evaluations. In reality, ISO-25000 replaces the two old ISO standards i.e. ISO-9126 and ISO-14598. ***SQuaRE*** is divided into sub parts such as: ISO 2500n - Quality Management Division; ISO 2501n - Quality Model Division; ISO 2502n - Quality Measurement Division; ISO 2503n - Quality Requirements Division; and ISO 2504n - Quality Evaluation Division. The main contents of ***SQuaRE*** are: Terms and definitions, Reference Models, General guide, Individual division guides and Standard related to Requirement Engineering (i.e. specification, planning, measurement and evaluation process

ISO/IEC 12119: This standard deals with Software packages delivered to the client. It does not focus or deal with the client's (the person/organization whom Software is delivered) production process. The main contents are related to the following items: Set of Requirements for Software packages; Instructions for testing the delivered Software package against the requirements;

ISO/IEC/IEEE 29119: The purpose of the ISO/IEC/IEEE 29119 series of software testing standards is to define an internationally-agreed set of standards for software testing that can be used by any organization when performing any form of software testing. This standard includes templates and examples of test documentation. The templates are arranged within clauses reflecting the overall test process description structure in ISO/IEC/IEEE 29119-2, i.e. by the test process in which they are being produced. Annex A contains outlines of the contents of each document. Annex B contains mappings of ISO/IEC/IEEE 29119-2. Annex C contains an overview of the examples. Annexes D to S contain examples of the application of the templates. Annex T provides mappings to existing standards. The Bibliography for ISO/IEC/IEEE 29119-3:2013 is at the end of the document. ISO/IEC/IEEE 29119-3:2013 supports dynamic testing, functional and non-functional testing, manual and automated testing, and scripted and unscripted testing. The documentation templates defined in ISO/IEC/IEEE 29119-3:2013 can be used in conjunction with any software development lifecycle model.

Some of the other standards related to QA and Testing processes are:

IEEE 829: A standard for the format of documents used in different stages of software testing.

IEEE 1061: A methodology for establishing quality requirements, identifying, implementing, analyzing, and validating the process and product of software quality metrics is defined.

IEEE 1059: Guide for Software Verification and Validation Plans.

IEEE 1008: A standard for unit testing.

IEEE 1012: A standard for Software Verification and Validation.

IEEE 1028: A standard for software inspections.

IEEE 1044: A standard for the classification of software anomalies.

IEEE 1044-1: A guide to the classification of software anomalies.

IEEE 830: A guide for developing system requirements specifications.

IEEE 730: A standard for software quality assurance plans.

IEEE 1061: A standard for software quality metrics and methodology.

IEEE 12207: A standard for software life cycle processes and life cycle data.

BS 7925-1: A vocabulary of terms used in software testing.

BS 7925-2: A standard for software component testing.

3.5.3 Integration testing check list

In MyHealthAvatar we created a checklist to be used as a guideline for understanding general system integration issues of the platform. If a question cannot be answered affirmatively, the associated issue was carefully examined and appropriate action was taken.

Before starting

- *Have you implemented your system taking into account the development life cycle approach?*
- *Do your test plans include and support integration efforts?*
- *Does your development plan allocate adequate time and resources for system integration efforts, including rework time?*
- *Are the interfaces between components, assemblies, subsystems, and systems defined in adequate detail?*
- *Will hardware be available for testing software during integration (i.e. activity sensor, mobile phones)?*
- *Is there a contingency plan if the schedule slips or if the integration schedule is compressed?*
- *Are all elements of the system included in the integration plan?*
- *Is all necessary documentation current and available for reference?*

During the integration phase

- *Is there an efficient rework cycle in place to fix problems found during integration testing?*
- *Are "fixed" modules or components integrated and retested at all levels of integration up to the level where the problem was found?*
- *Is the people element (operators, maintainers, logisticians, trainers, etc.) being prepared to work with the system when it is deployed?*
- *Are you following an iterative, progressive integration process?*
- *Are experienced integrators involved with the integration?*
- *Are area/subject matter experts involved with the integration?*
- *Is adequate time being allowed for integration, testing, rework, reintegration, and retesting?*
- *Are all necessary resources being made available for integration?*
- *Is adequate testing being performed on integrated units (assemblies, subsystems, elements, system) to ensure that there are no surprises during acceptance testing?*
- *Are you updating documentation during rework?*
- *Are integration and system test errors being traced back to requirements and design? And if so, are the requirements and design being updated?*

3.5.4 Methodologies used for testing the MHA software components

There are different methods which can be used for software testing. This section briefly describes those methods and describes the ones employed for the testing of the SW componnets developed in MyHealthAvatar.

Black Box Testing: testing without having any knowledge of the interior workings of the application is Black Box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon. *Advantages*: Well suited and efficient for large code segments; Code Access not required; Clearly separates user's perspective from the developer's perspective through visibly defined roles; and Large numbers of moderately skilled testers can test the application with no knowledge of implementation, programming language or operating systems. *Disadvantages*: Limited Coverage since only a selected number of test scenarios are actually performed; Inefficient testing, due to the fact that the tester only has limited knowledge about an application; Blind Coverage, since the tester cannot target specific code segments or error prone areas; but test cases are difficult to design.

White Box Testing: the detailed investigation of internal logic and structure of the code. White box testing is also called glass testing or open box testing. In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code. The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately. *Advantages*: As the tester has knowledge of the source code, it becomes very easy to find out which type of data can help in testing the application effectively: in optimizing the code; parts of code can be removed which can bring in hidden defects.; Due to the tester's knowledge about the code, maximum coverage is attained during test scenario writing. *Disadvantages*: Due to the fact that a skilled tester is needed to perform white box testing, the costs are increased; Sometimes it is impossible to look into every nook and corner to find out hidden errors that may create problems as many paths will go untested; It is difficult to maintain white box testing as the use of specialized tools like code analysers and debugging tools are required.

Grey Box Testing: a technique to test the application with limited knowledge of the internal workings of an application. In software testing, the term "the more you know the better" carries a lot of weight when testing an application. Mastering the domain of a system always gives the tester an edge over someone with limited domain knowledge. Unlike black box testing, where the tester only tests the application's user interface, in grey box testing, the tester has access to design documents and the database. Having this knowledge, the tester is able to better prepare test data and test scenarios when making the test plan. *Advantages*: Offers combined benefits of black box and white box testing wherever possible; Grey box testers don't rely on the source code; instead they rely on interface definition and functional specifications; Based on the limited information available, a grey box tester can design excellent test scenarios especially around communication protocols and data type handling; The test is done from the point of view of the user and not the designer. *Disadvantages*: Since the access to source code is not available, the ability to go over the code and test coverage is limited; the tests can be redundant if the software designer has already run a test case; testing every possible input stream is unrealistic because it would take an unreasonable amount of time; therefore, many program paths will go untested.

Table 1: Comparison between the Three Testing Types

Black Box Testing	Grey Box Testing	White Box Testing
The Internal Workings of an application are not required to be known	Somewhat knowledge of the internal workings are known	Tester has full knowledge of the Internal workings of the application

Also known as closed box testing, data driven testing and functional testing	Another term for grey box testing is translucent testing as the tester has limited knowledge of the insides of the application	Also known as clear box testing, structural testing or code based testing
Performed by end users and also by testers and developers	Performed by end users and also by testers and developers	Normally done by testers and developers
-Testing is based on external expectations -Internal behavior of the application is unknown	Testing is done on the basis of high level database diagrams and data flow diagrams	Internal workings are fully known and the tester can design test data accordingly
This is the least time consuming and exhaustive	Partly time consuming and exhaustive	The most exhaustive and time consuming type of testing
Not suited to algorithm testing	Not suited to algorithm testing	Suited for algorithm testing
This can only be done by trial and error method	Data domains and Internal boundaries can be tested, if known	Data domains and Internal boundaries can be better tested

3.5.4.1 Functional Testing

Functional testing involves testing the application against the business/user/case requirements. It is performed using the required functional specifications provided by the users of the application (gathered during the design specifications phase). In MyHealthAvatar we had specific set of use cases during the beginning of the project to collect the necessary user requirements of the system.

The functional testing can be broken down into four components

- Unit testing: testing of individual software modules or components that make up an application or system.
- Integration testing: the testing of the different modules/components that have been successfully unit tested when integrated together to perform specific tasks and activities (also known as scenario testing). In MyHealthAvatar the platform has been tested and validated through specific scenarios.
- System testing: testing the entire system for errors and bugs
- Acceptance testing: the final phase of functional software testing and involves making sure that all the product/project requirements have been met and that the end-users and customers have tested the system to make sure it operates as expected and meets all their defined requirements.

3.5.4.2 Non-Functional Testing

Non-functional testing involves testing the platform against non-functional requirements, which typically involve measuring/testing the system against defined technical qualities because for example: vulnerability, scalability, usability. Some examples of non-functional testing are described below:

Performance, Load, Stress Testing: measuring how a system behaves under an increasing load (both numbers of users and data volumes), load testing is verifying that the system can operate at the required response times when subjected to its expected load, and stress testing is finding the failure point(s) in the system when the tested load exceeds that which it can support.

Security, Vulnerability Testing: tests the platform for confidentiality, integrity, authentication, availability, and non-repudiation. Individual tests are conducted to prevent any unauthorized access to the software code. MHA has a security and legal framework which was followed during the implementation and evaluation of the platform.

Usability Testing: looks at the end-user usability aspect of the software. The ease with which a user can access the product forms the main testing point. Usability testing looks at five aspects of testing, - learnability, efficiency, satisfaction, memorability, and errors. The usability testing is discussed in Section XX in this deliverable for the four high end use case scenarios.

Compatibility Testing: tests that the platform is compatible with all the specified operating systems, hardware platforms, web browsers, mobile devices, and other designed third-party programs (e.g. browser plugins). Compatibility tests check that the product works as expected across all the different hardware/software combinations and that all functionality is consistently supported.

3.5.5 MHA tests reports and results

Integration is an iterative, progressive process that has components integrated into assemblies, then assemblies are tested for functionality and successful testing is followed by integration of subsystems from assemblies, which are also tested for correct functionality to be able to integrate subsystems into the complete platform that must be tested for functionality by the end users. An absolute essential to the integration process is the detailed knowledge of the interfaces among components, assemblies, subsystems, and between the system and other external systems it will need to work with. Defining interfaces and maintaining those definitions is a primary responsibility of the systems engineering and design process and is mandatory for all developing partners and actors (all related info are include in deliverable D3.4). In addition to the usability results a number of functional tests were also presented in D6.3, D6.4 and D8.2.

Successful system integration involved testing every module of the MHA platform and this was an on-going procedure throughout the system's development cycle. In MyHealthAvatar the starting point for successful integration was the definition of the initial user requirement and the implementation of the functional analysis (reported in D9.3 Report on the clinical acceptability and evaluation of MyHealthAvatar and future recommendation), synthesis, user case studies, interface definition, etc. In addition it is also necessary to be able to ensure activities such as configuration, design, and testing that are essential to ensure all the pieces for integration.

The MyHeathAvatar's partners responsible for the development of the platform selected different approaches and ways of ensuring that the integrated platform is fully tested..

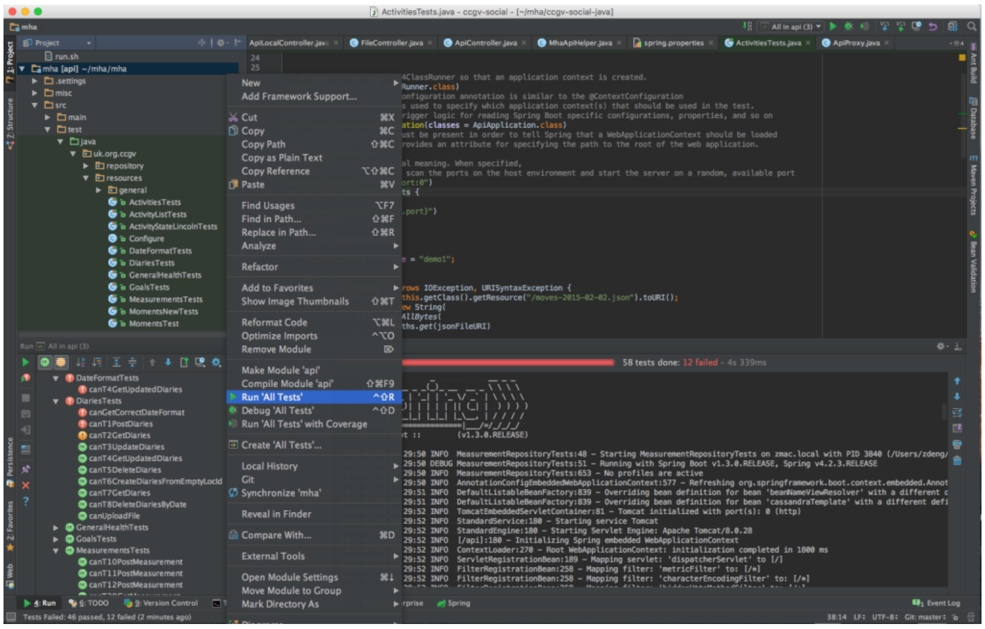
MHA is a complex environment that includesa large number of different software components and services that need to be tested and their functioning validated. It was thus important to follow a robust testing methodology for making sure that software products/systems being developed have been fully tested, to make sure they meet their specified requirements and guarantee that they can successfully operate in all the anticipated environments with the required usability and security. As a result the testing of the MHA platform's software components encompass everything from unit testing of individual modules, integration testing of the entire system to specialized forms of testing such as security and performance. Below we present indicative results/reports on various software testing performed within MyHealthAvatar

3.5.5.1 MyHealthAvatar Platform/API/Application-service tests

MHA web and API are coded in Java and automatically unit tested, mainly based upon following libraries and frameworks:

- Maven: A build automation tool, which also manages project dependencies.
- JUnit: A popular Java unit-testing framework.
- REST-assured: A Java DSL for easy testing of REST services.
- Hamcrest: A framework for writing matcher objects allowing the rules to be defined declarively.
- Spring JUnit: Spring framework's support for running unit-tests for Spring projects in JUnit.

Following is in everyday development of bug fixes, refactoring of existing codes and new features, unit tests could be run directly from the development environment as part of the coding.



```
4ClassRunner so that an application context is created.
Runner.class)
onfiguration annotation is similar to the @ContextConfiguration
s used to specify which application context(s) that should be used in the test.
rigger logic for reading Spring Boot specific configurations, properties, and so on
ation(classes = ApiApplication.class)
ust be present in order to tell Spring that a WebApplicationContext should be loaded
rovides an attribute for specifying the path to the root of the web application.

al meaning. When specified,
 scan the ports on the host environment and start the server on a random, available port
ort:0")
ts {

.port}")


e = "demo1";


rows IOException, URISyntaxException {
this.getClass().getResource("/moves-2015-02-02.json").toURI();
ew String(
AllBytes(
ths.get(jsonFileURI)
```

58 tests done: 12 failed - 4s 339ms

```
t ::         (v1.3.0.RELEASE)

29:50 INFO  MeasurementRepositoryTests:48 - Starting MeasurementRepositoryTests on zmac.local with PID 3840 (/Users/zdeng/
29:50 DEBUG MeasurementRepositoryTests:51 - Running with Spring Boot v1.3.0.RELEASE, Spring v4.2.3.RELEASE
29:50 INFO  MeasurementRepositoryTests:653 - No profiles are active
29:50 INFO  AnnotationConfigEmbeddedWebApplicationContext:577 - Refreshing org.springframework.boot.context.embedded.Annot
29:51 INFO  DefaultListableBeanFactory:839 - Overriding bean definition for bean 'beanNameViewResolver' with a different c
29:51 INFO  DefaultListableBeanFactory:839 - Overriding bean definition for bean 'cassandraTemplate' with a different defi
29:52 INFO  TomcatEmbeddedServletContainer:81 - Tomcat initialized with port(s): 0 (http)
29:52 INFO  StandardService:180 - Starting service Tomcat
29:52 INFO  StandardEngine:180 - Starting Servlet Engine: Apache Tomcat/8.0.28
29:52 INFO  [/api]:180 - Initializing Spring embedded WebApplicationContext
29:52 INFO  ContextLoader:270 - Root WebApplicationContext: initialization completed in 1800 ms
29:52 INFO  ServletRegistrationBean:189 - Mapping servlet: 'dispatcherServlet' to [/]
29:52 INFO  FilterRegistrationBean:258 - Mapping filter: 'metricFilter' to: [/*]
29:52 INFO  FilterRegistrationBean:258 - Mapping filter: 'characterEncodingFilter' to: [/*]
```

Figure 3-4: Run test during development from IDE's context menu

Unit tests cases are mapped in a one to one relationship to the source code Java class file, for example UserProfileController.java file from src/ folder will have its corresponding test UserProfileControllerTests.java in test/ folder with same package names (for Java package access).

We take one test case for example; in the first place meta-programming style annotations are used to configure the test class:

```
// We need the SpringJUnit4ClassRunner so that an application context is created.
@RunWith(SpringJUnit4ClassRunner.class)
// The @SpringApplicationConfiguration annotation is similar to the @ContextConfiguration
// annotation in that it is used to specify which application context(s) that should be used in the test.
// Additionally, it will trigger logic for reading Spring Boot specific configurations, properties, and so on
@SpringApplicationConfiguration(classes = ApiApplication.class)
//  @WebAppConfiguration must be present in order to tell Spring that a WebApplicationContext should be loaded
// for the test. It also provides an attribute for specifying the path to the root of the web application.
@WebAppConfiguration
// the value 0 has a special meaning. When specified,
// it tells Spring Boot to scan the ports on the host environment and start the server on a random, available port
@IntegrationTest("server.port:0")
public class UserProfileControllerTests {

    // …
}
```

Within the class, the stage for the test cases are set up, which configure the sample data files to be used, as well as username to test, etc.

```

@Value("${local.server.port}")
int port;

String activitiesJson;

private String username = "demo1";

@Before
public void setUp() throws IOException, URISyntaxException {
    URI jsonFileURI = this.getClass().getResource("/moves-2015-02-02.json").toURI();
    activitiesJson = new String(
            Files.readAllBytes(
                    Paths.get(jsonFileURI)
            )
        );

    RestAssured.port = port;
    RestAssured.registerParser("text/plain", String.class);
}
```

For unit test cases themselves, they are normally fairly short and self-explained by reading the source code.

For example, following test cases will test post sample JSON profile to API will end with http status of 201 (created) and result of 'ok':

```
@Test
public void canT10PostProfile() {
    given()
            .contentType("application/json").body(fullProfileJson)
            .when().post(Configure.contextPath + "/user/full_profile?username=" + username)
```

```
        .then()
        .statusCode(HttpStatus.SC_CREATED)
        .content(Matchers.containsString("ok"));
}
```

It is extremely important to unit test code written and MHA aim to cover the code as much as the resource is able to reach. It currently has the line coverage of 45%, which is considered as fair amount of coverage for its feasibility research nature.

MHA require the developers to make sure all tests passes before code are pushed to repository, so that new changes from one organization and/or developer will not break the code of others.

Maven Surefire Plugin is used to generate report of test results for easier documentation and observation. Take the API project for example, member of team can view which test cases takes the most time by query the Surefire generated results from command line:

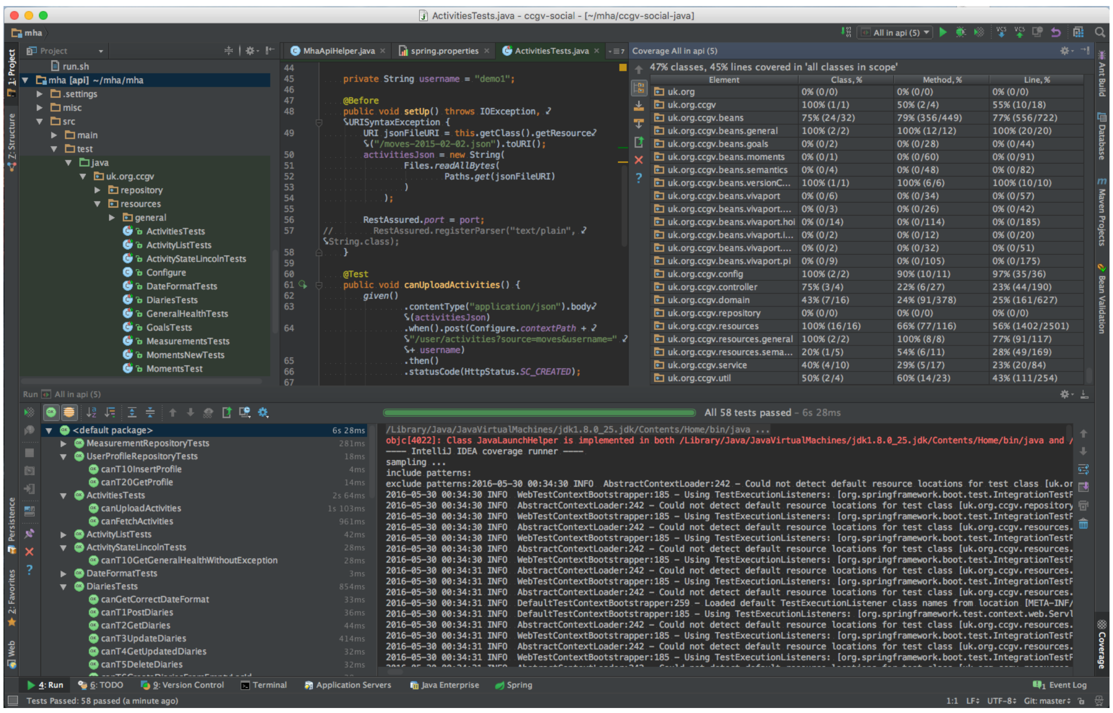
```
$ grep -h "<testcase" `find . -iname "TEST-*.xml"` | sed 's/<testcase name="\(.*\)"
classname="\(.*\)" time="\(.*\)".*/\3 | \1 | \2/' | sort –rn
```

```
0.619 | canUploadActivities | uk.org.ccgv.resources.ActivitiesTests

0.411 | canFetchActivities | uk.org.ccgv.resources.ActivitiesTests

0.337 | canT20GetAlerts | uk.org.ccgv.resources.general.AlertsTests

0.298 | canT15PostMomentsData | uk.org.ccgv.resources.MomentsNewTests

0.296 | canT21GetForthAlerts | uk.org.ccgv.resources.general.AlertsTests

0.133 | canT10InsertMeasurement | uk.org.ccgv.repository.MeasurementRepositoryTests

0.103 | canT40GetProfile1108 | uk.org.ccgv.resources.ProfileTests

0.062 | canUploadFile | uk.org.ccgv.resources.DiariesTests

0.06 | canT23GetMeasurementByDateRange | uk.org.ccgv.resources.MeasurementsTests

0.05 | canT22GetMeasurementFromDate | uk.org.ccgv.resources.MeasurementsTests

0.048 | canT30PostProfile1108 | uk.org.ccgv.resources.ProfileTests

0.043 | canT21GetMeasurementByDate | uk.org.ccgv.resources.MeasurementsTests

0.037 | canT10PostAlerts | uk.org.ccgv.resources.general.AlertsTests

0.036 | canT8DeleteDiariesByDate | uk.org.ccgv.resources.DiariesTests

0.032 | canT22GetGeneralHealth | uk.org.ccgv.resources.general.VersionControlTests

0.028 | canT20GetProfile | uk.org.ccgv.repository.MeasurementRepositoryTests
```

```
0.023 | canT2GetDiaries | uk.org.ccgv.resources.DiariesTests

0.023 | canT23GetProfileInsurance | uk.org.ccgv.resources.ProfileTests

0.022 | canFetchActivityList | uk.org.ccgv.resources.ActivityListTests

0.02 | canT24GetLatestMeasurement | uk.org.ccgv.resources.MeasurementsTests

0.02 | canT21GetProfilePersonalInformation | uk.org.ccgv.resources.ProfileTests

0.02 | canT20GetMeasurement | uk.org.ccgv.resources.MeasurementsTests

0.019 | canT10PostGoals | uk.org.ccgv.resources.GoalsTests

0.019 |canT10GetGeneralHealthWithoutException | uk.org.ccgv.resources.ActivityStateLincolnTests

0.018 | canT4GetUpdatedDiaries | uk.org.ccgv.resources.DiariesTests

0.018 | canT23GetGeneralHealth | uk.org.ccgv.resources.GeneralHealthTests

0.018 | canT20GetRisks | uk.org.ccgv.resources.general.RiskAssessmentsTests

0.018 | canT11PostForthAlerts | uk.org.ccgv.resources.general.AlertsTests

0.018 | canGetCorrectDateFormat | uk.org.ccgv.resources.DiariesTests

0.016 | canT7GetDiaries | uk.org.ccgv.resources.DiariesTests

0.016 | canT5DeleteDiaries | uk.org.ccgv.resources.DiariesTests

0.016 | canT20GetGeneralHealth | uk.org.ccgv.resources.GeneralHealthTests

0.016 | canT1PostDiaries | uk.org.ccgv.resources.DiariesTests

0.015 | canT30GetLatestAlerts | uk.org.ccgv.resources.general.AlertsTests

0.015 | canT20GetVersionControl | uk.org.ccgv.resources.general.VersionControlTests

0.014 | canT6CreateDiariesFromEmptyLocId | uk.org.ccgv.resources.DiariesTests

0.014 | canT3UpdateDiaries | uk.org.ccgv.resources.DiariesTests

0.014 | canT10PostRisks | uk.org.ccgv.resources.general.RiskAssessmentsTests

0.013 | canT30GetLatestRisks | uk.org.ccgv.resources.general.RiskAssessmentsTests

0.013 | canT21GetVersionControl | uk.org.ccgv.resources.general.VersionControlTests

0.013 | canT21GetGeneralHealth | uk.org.ccgv.resources.GeneralHealthTests

0.013 | canT12PostMeasurement | uk.org.ccgv.resources.MeasurementsTests

0.013 | canT11PostMeasurement | uk.org.ccgv.resources.MeasurementsTests
```

```
0.013 | canT10PostMeasurement | uk.org.ccgv.resources.MeasurementsTests

0.012 | canT22GetProfileVitalSigns | uk.org.ccgv.resources.ProfileTests

0.012 | canT15PostVersionControl | uk.org.ccgv.resources.general.VersionControlTests

0.012 | canT11PostGeneralHealth | uk.org.ccgv.resources.GeneralHealthTests

0.011 | canT22GetGeneralHealth | uk.org.ccgv.resources.GeneralHealthTests

0.011 | canT10PostGeneralHealth | uk.org.ccgv.resources.GeneralHealthTests

0.007 | canT20GetProfile | uk.org.ccgv.resources.ProfileTests

0.004 | canT20GetProfile | uk.org.ccgv.repository.UserProfileRepositoryTests

0.004 | canT10PostProfile | uk.org.ccgv.resources.ProfileTests

0.002 | testDateFormat | uk.org.ccgv.resources.GoalsTests

0.001 | canT4GetUpdatedDiaries | uk.org.ccgv.resources.DateFormatTests

0 | canT20GetMomentsData | uk.org.ccgv.resources.MomentsTest

0 | canT15PostMomentsData | uk.org.ccgv.resources.MomentsTest

0 | canT11PostersonalInformation | uk.org.ccgv.resources.ProfileTests

0 | canT10InsertProfile | uk.org.ccgv.repository.UserProfileRepositoryTests
```

MHA adopt the methodology of Continuously Integration and unit tests are part of the build process, the build will only success if all unit tests pass. As mentioned previously all test cases can be executed directly from within Java IDE of developer’s choice.



```
ActivitiesTests.java - ccgv-social - [~/mha/ccgv-social-java]
```

```
mha
  run.sh
  mha [api] ~/mha/mha
    .settings
    misc
    src
      main
      test
        java
          uk.org.ccgv
            repository
            resources
              general
              ActivitiesTests
              ActivityListTests
              ActivityStateLincolnTests
              Configure
              DateFormatTests
              DiariesTests
              GeneralHealthTests
              GoalsTests
              MeasurementsTests
              MomentsNewTests
              MomentsTest
```

```java
    private String username = "demo1";

    @Before
    public void setUp() throws IOException, ↲
    ↳URISyntaxException {
        URI jsonFileURI = this.getClass().getResource↲
        ↳("/moves-2015-02-02.json").toURI();
        activitiesJson = new String(
                Files.readAllBytes(
                        Paths.get(jsonFileURI)
                )
        );

        RestAssured.port = port;
//          RestAssured.registerParser("text/plain", ↲
↳String.class);
    }

    @Test
    public void canUploadActivities() {
        given()
                .contentType("application/json").body↲
                ↳(activitiesJson)
                .when().post(Configure.contextPath + ↲
                ↳"/user/activities?source=moves&username=" ↲
                ↳+ username)
                .then()
                .statusCode(HttpStatus.SC_CREATED);
```

Coverage All in api (5)

47% classes, 45% lines covered in 'all classes in scope'

Element	Class, %	Method, %	Line, %
uk.org	0% (0/0)	0% (0/0)	0% (0/0)
uk.org.ccgv	100% (1/1)	50% (2/4)	55% (10/18)
uk.org.ccgv.beans	75% (24/32)	79% (356/449)	77% (556/722)
uk.org.ccgv.beans.general	100% (2/2)	100% (12/12)	100% (20/20)
uk.org.ccgv.beans.goals	0% (0/2)	0% (0/28)	0% (0/44)
uk.org.ccgv.beans.moments	0% (0/1)	0% (0/60)	0% (0/91)
uk.org.ccgv.beans.semantics	0% (0/4)	0% (0/48)	0% (0/82)
uk.org.ccgv.beans.versionC...	100% (1/1)	100% (6/6)	100% (10/10)
uk.org.ccgv.beans.vivaport	0% (0/6)	0% (0/34)	0% (0/57)
uk.org.ccgv.beans.vivaport....	0% (0/3)	0% (0/26)	0% (0/42)
uk.org.ccgv.beans.vivaport.hoi	0% (0/14)	0% (0/114)	0% (0/185)
uk.org.ccgv.beans.vivaport.i...	0% (0/2)	0% (0/12)	0% (0/20)
uk.org.ccgv.beans.vivaport....	0% (0/2)	0% (0/32)	0% (0/51)
uk.org.ccgv.beans.vivaport.pi	0% (0/9)	0% (0/105)	0% (0/175)
uk.org.ccgv.config	100% (2/2)	90% (10/11)	97% (35/36)
uk.org.ccgv.controller	75% (3/4)	22% (6/27)	23% (44/190)
uk.org.ccgv.domain	43% (7/16)	24% (91/378)	25% (161/627)
uk.org.ccgv.repository	0% (0/0)	0% (0/0)	0% (0/0)
uk.org.ccgv.resources	100% (16/16)	66% (77/116)	56% (1402/2501)
uk.org.ccgv.resources.general	100% (2/2)	100% (8/8)	77% (91/117)
uk.org.ccgv.resources.sema...	20% (1/5)	54% (6/11)	28% (49/169)
uk.org.ccgv.service	40% (4/10)	29% (5/17)	23% (20/84)
uk.org.ccgv.util	50% (2/4)	60% (14/23)	43% (111/254)

Run All in api (5)

All 58 tests passed - 6s 28ms

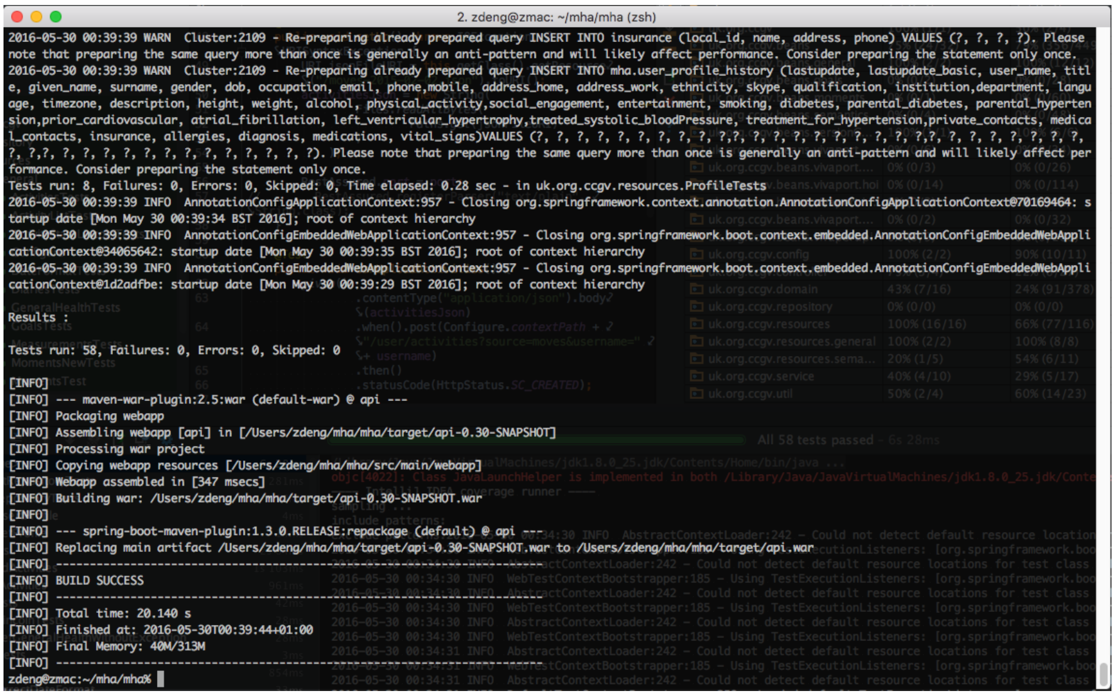
```
<default package>                                    6s 28ms
  MeasurementRepositoryTests                          281ms
  UserProfileRepositoryTests                           18ms
    canT10InsertProfile                                 4ms
    canT20GetProfile                                   14ms
  ActivitiesTests                                    2s 64ms
    canUploadActivities                             1s 103ms
    canFetchActivities                                961ms
  ActivityListTests                                    42ms
  ActivityStateLincolnTests                            28ms
    canT10GetGeneralHealthWithoutException             28ms
  DateFormatTests                                       3ms
  DiariesTests                                        854ms
    canGetCorrectDateFormat                            33ms
    canT1PostDiaries                                   36ms
    canT2GetDiaries                                    44ms
    canT3UpdateDiaries                                414ms
    canT4GetUpdatedDiaries                             32ms
    canT5DeleteDiaries                                 32ms
```

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_25.jdk/Contents/Home/bin/java ...
objc[4022]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachines/jdk1.8.0_25.jdk/Contents/Home/bin/java and /
---- IntelliJ IDEA coverage runner ----
sampling ...
include patterns:
exclude patterns:2016-05-30 00:34:30 INFO  AbstractContextLoader:242 - Could not detect default resource locations for test class [uk.or
2016-05-30 00:34:30 INFO  WebTestContextBootstrapper:185 - Using TestExecutionListeners: [org.springframework.boot.test.IntegrationTestF
2016-05-30 00:34:30 INFO  AbstractContextLoader:242 - Could not detect default resource locations for test class [uk.org.ccgv.repository
2016-05-30 00:34:30 INFO  WebTestContextBootstrapper:185 - Using TestExecutionListeners: [org.springframework.boot.test.IntegrationTestF
2016-05-30 00:34:30 INFO  AbstractContextLoader:242 - Could not detect default resource locations for test class [uk.org.ccgv.resources.
2016-05-30 00:34:30 INFO  WebTestContextBootstrapper:185 - Using TestExecutionListeners: [org.springframework.boot.test.IntegrationTestF
2016-05-30 00:34:30 INFO  AbstractContextLoader:242 - Could not detect default resource locations for test class [uk.org.ccgv.resources.
2016-05-30 00:34:30 INFO  WebTestContextBootstrapper:185 - Using TestExecutionListeners: [org.springframework.boot.test.IntegrationTestF
2016-05-30 00:34:31 INFO  AbstractContextLoader:242 - Could not detect default resource locations for test class [uk.org.ccgv.resources.
2016-05-30 00:34:31 INFO  WebTestContextBootstrapper:185 - Using TestExecutionListeners: [org.springframework.boot.test.IntegrationTestF
2016-05-30 00:34:31 INFO  AbstractContextLoader:242 - Could not detect default resource locations for test class [uk.org.ccgv.resources.
2016-05-30 00:34:31 INFO  DefaultTestContextBootstrapper:259 - Loaded default TestExecutionListener class names from location [META-INF/
2016-05-30 00:34:31 INFO  DefaultTestContextBootstrapper:185 - Using TestExecutionListeners: [org.springframework.test.context.web.Servl
2016-05-30 00:34:31 INFO  AbstractContextLoader:242 - Could not detect default resource locations for test class [uk.org.ccgv.resources.
2016-05-30 00:34:31 INFO  WebTestContextBootstrapper:185 - Using TestExecutionListeners: [org.springframework.boot.test.IntegrationTestF
2016-05-30 00:34:31 INFO  AbstractContextLoader:242 - Could not detect default resource locations for test class [uk.org.ccgv.resources.
2016-05-30 00:34:31 INFO  WebTestContextBootstrapper:185 - Using TestExecutionListeners: [org.springframework.boot.test.IntegrationTestF
```

```
Tests Passed: 58 passed (a minute ago)                    1:1  LF÷  UTF-8÷  Git: master÷
```

Figure 3-5: All tests passed with code coverage analyses result of 47% classes and 45% lines

And during the build process, maven will also execute the test cases in the build server from command line.



```
2. zdeng@zmac: ~/mha/mha (zsh)
```

```
2016-05-30 00:39:39 WARN  Cluster:2109 - Re-preparing already prepared query INSERT INTO insurance (local_id, name, address, phone) VALUES (?, ?, ?, ?). Please
note that preparing the same query more than once is generally an anti-pattern and will likely affect performance. Consider preparing the statement only once.
2016-05-30 00:39:39 WARN  Cluster:2109 - Re-preparing already prepared query INSERT INTO mha.user_profile_history (lastupdate, lastupdate_basic, user_name, titl
e, given_name, surname, gender, dob, occupation, email, phone,mobile, address_home, address_work, ethnicity, skype, qualification, institution,department, langu
age, timezone, description, height, weight, alcohol, physical_activity,social_engagement, entertainment, smoking, diabetes, parental_diabetes, parental_hyperten
sion,prior_cardiovascular, atrial_fibrillation, left_ventricular_hypertrophy,treated_systolic_bloodPressure, treatment_for_hypertension,private_contacts, medica
l_contacts, insurance, allergies, diagnosis, medications, vital_signs)VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?,?, ?, ?, ?, ?, ?, ?, ?, ?, ?,?, ?, ?, ?, ?, ?, ?, ?,
?, ?,?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?). Please note that preparing the same query more than once is generally an anti-pattern and will likely affect per
formance. Consider preparing the statement only once.
Tests run: 8, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.28 sec - in uk.org.ccgv.resources.ProfileTests
2016-05-30 00:39:39 INFO  AnnotationConfigApplicationContext:957 - Closing org.springframework.context.annotation.AnnotationConfigApplicationContext@70169464: s
tartup date [Mon May 30 00:39:34 BST 2016]; root of context hierarchy
2016-05-30 00:39:39 INFO  AnnotationConfigEmbeddedWebApplicationContext:957 - Closing org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebAppli
cationContext@34065642: startup date [Mon May 30 00:39:35 BST 2016]; root of context hierarchy
2016-05-30 00:39:39 INFO  AnnotationConfigEmbeddedWebApplicationContext:957 - Closing org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebAppli
cationContext@1d2adfbe: startup date [Mon May 30 00:39:29 BST 2016]; root of context hierarchy

Results :

Tests run: 58, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-war-plugin:2.5:war (default-war) @ api ---
[INFO] Packaging webapp
[INFO] Assembling webapp [api] in [/Users/zdeng/mha/mha/target/api-0.30-SNAPSHOT]
[INFO] Processing war project
[INFO] Copying webapp resources [/Users/zdeng/mha/mha/src/main/webapp]
[INFO] Webapp assembled in [347 msecs]
[INFO] Building war: /Users/zdeng/mha/mha/target/api-0.30-SNAPSHOT.war
[INFO]
[INFO] --- spring-boot-maven-plugin:1.3.0.RELEASE:repackage (default) @ api ---
[INFO] Replacing main artifact /Users/zdeng/mha/mha/target/api-0.30-SNAPSHOT.war to /Users/zdeng/mha/mha/target/api.war
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 20.140 s
[INFO] Finished at: 2016-05-30T00:39:44+01:00
[INFO] Final Memory: 40M/313M
[INFO] ------------------------------------------------------------------------
zdeng@zmac:~/mha/mha%
```

Figure 3-6: all tests are run during build process before deploy on server

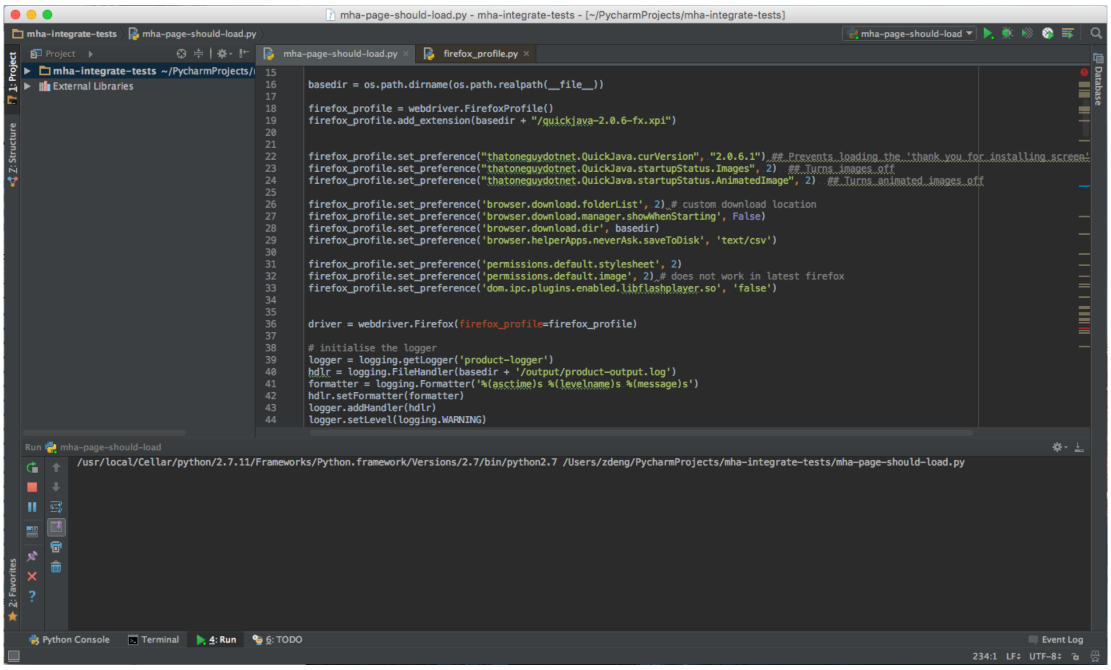
For integration tests, following tools and frameworks are utilized:

- Python: A programming language, which is quicker in writing and running integration tests.
- Selenium-WebDriver: A browser automation tools, using each browser's native support for automation.
- Firefox / Chrome: the browser drive by Selenium, which automatically executes integration test tasks.
- Splinter: An open source tool for testing web application use Python, which run on top of above tools.
- Selenium-Server: allows distribute integration tests over multiple machine.

Docker is an open platform, which could run same versions of applications from different development environments, whether on laptop or desktop, Windows, Linux or Mac. In order to create a unified environment for integration tests, Docker is utilized to allow developers and testers run the same version of MySQL, Cassandra, and Tomcat etc. Following script are used for initial run of Docker Cassandra and Tomcat images:

```
docker run -p 9042:9042 -v /home/ccgv/cassandra-data:/var/lib/cassandra --name api-cassandra -d cassandra:2.2.4

docker run -d -p 8080:8080 -p 8009:8009 --link api-cassandra:cassandra --name api-tomcat -v /home/ccgv/tomcat-api:/opt/tomcat/webapps dordoka/tomcat
```

The Cassandra and Tomcat are linked by Docker to allow interactions in between. As you can see that Cassandra 2.2.4 version is used in above scripts, which make sure everyone from development and testing team are on the same page.



```
basedir = os.path.dirname(os.path.realpath(__file__))

firefox_profile = webdriver.FirefoxProfile()
firefox_profile.add_extension(basedir + "/quickjava-2.0.6-fx.xpi")


firefox_profile.set_preference("thatoneguydotnet.QuickJava.curVersion", "2.0.6.1") ## Prevents loading the 'thank you for installing screen'
firefox_profile.set_preference("thatoneguydotnet.QuickJava.startupStatus.Images", 2)  ## Turns images off
firefox_profile.set_preference("thatoneguydotnet.QuickJava.startupStatus.AnimatedImage", 2)  ## Turns animated images off

firefox_profile.set_preference('browser.download.folderList', 2) # custom download location
firefox_profile.set_preference('browser.download.manager.showWhenStarting', False)
firefox_profile.set_preference('browser.download.dir', basedir)
firefox_profile.set_preference('browser.helperApps.neverAsk.saveToDisk', 'text/csv')

firefox_profile.set_preference('permissions.default.stylesheet', 2)
firefox_profile.set_preference('permissions.default.image', 2) # does not work in latest firefox
firefox_profile.set_preference('dom.ipc.plugins.enabled.libflashplayer.so', 'false')


driver = webdriver.Firefox(firefox_profile=firefox_profile)

# initialise the logger
logger = logging.getLogger('product-logger')
hdlr = logging.FileHandler(basedir + '/output/product-output.log')
formatter = logging.Formatter('%(asctime)s %(levelname)s %(message)s')
hdlr.setFormatter(formatter)
logger.addHandler(hdlr)
logger.setLevel(logging.WARNING)
```

```
/usr/local/Cellar/python/2.7.11/Frameworks/Python.framework/Versions/2.7/bin/python2.7 /Users/zdeng/PycharmProjects/mha-integrate-tests/mha-page-should-load.py
```

Figure 3-7: python script which test if the MHA home page will load with content

MHA has automated integration tests run by developers against development server to make sure that essential functionalities are working as expected. Manual integration tests are also curries out by individuals after new version is published.

Following items are normally smoke tested when a staging version of integrated platform are published online:

- Login page loads with all content, images and styles
- Login use demo user should success
- Dashboard page should load with all information and data
- Life->Diary page should load with correct content and route should be available when a specific data is chosen
- Life->lifeTracker should load and interact the way expected
- Well-being -> Weight should load and tools should work properly
- Disease menu and models should execute
- Chart should work with drag and drop functional
- Tools menu should work with Nephroblastoma tools execute with result display properly
- Administrator menu should only visible to admin role users and functions
- Questionnaire should load and post answer should work
- 3D Avatar model should load with all controls function
- Data Input should allow link to third party and social network
- Synchronization of data should work with linked devices and applications
- Profile should work and PDF generation should work
- Help menu should load help documents and videos normally, sound should play

When major changes happen more detailed tests are carried out with tasks distributed to different people from the team and other researchers from within partner organizations.

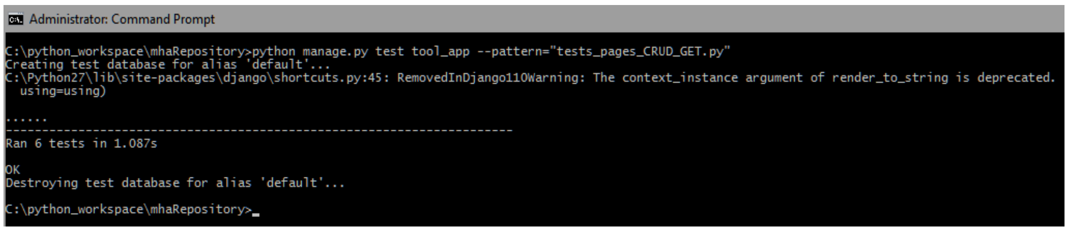
3.5.5.1 MyHealthAvatar IAPETUS tests

Test #: IAPETUS Views/API		**Type:** IAPETUS Views/API – unit test	
Date:	25/5/2016-30/5/2016	**Duration:**	1 hour
Objective: IAPETUS application will be tested. This is achieved by testing potential user actions against the application's views (and by extension the related forms), and the API. The system is tested with respect to the end user's inputs and outputs per application function. This means that the only criterion is determining whether the application responds to HTTP requests in the appropriate way and delivers the requested data to the user.			
Necessary hardware: Windows Server Machine			
Necessary software: According to the python/Django/Tastypie testing procedures, separate modules following the pattern tests_*.py are developed, in which the desired tests are implemented. For the IAPETUS application, each possible response to the end user when making requests via direct URL-typing in the browser, or navigating through the application's GUI, are tested. This is achieved by making GET/POST requests and simultaneously request or upload data. As a data security measure applied by python/Django tests to keep the production version of the database and the stored data secured, all tests and operations are tested across test databases,			

copying the schema of the original ones. This calls for creating test data to upload or retrieve, prior to executing the test functions
Preconditions: As a Web application, IAPETUS's testing a is a complex task, because a Web application is made of several layers of logic – from HTTP-level request handling, to form validation and processing, to template rendering. In addition, tests must not harm the already stored data. Tests are implemented and executed by IAPETUS's developer To produce and execute the tests for IAPETUS, knowledge of basic Python coding skills and the Django framework is required, along with knowledge of a command line interface (Windows command prompt, Linux bash shell, etc.) Tests can be run even in a single laptop, since they usually take under 10 seconds per batch to execute.
Set-up: Tests are developed and implemented by calling the original views/URL's of the application, similar to the production version of IAPETUS. The responses and any accompanying results are returned by the same mechanism as in the production version, as well. All the various data handling between the different layers is carried out by the same Django server, which is used for the production version. Finally, all data used in tests are created within the testing modules, prior to running the test functions and the testing database is a copy of the original, keeping the same schema with no stored data. Django testing framework inherently answers both key issues
Test Steps:
Step 1: Make an HTTP request (GET/POST) to the view's URL with some accompanying data for upload where necessary.
Step 2: Assume specific HTTP response codes and compare with the actual outcomes. Assume specific values of variables or cardinality/kind/etc. of results
Expected Test Results: Test Pass / Fail Criteria: In the tests_*.py files, a set of assertions are made for certain item values pertaining to the outcome of the described actions. A test is successful if the assertion is true. This means that the program behaves in the predicted and desirable way. Otherwise, the test is considered to have failed. Test Entry / Exit Criteria: Since test functions are grouped into separate python files callable from within the Django framework, unit testing can be executed anytime with no special preparations. To achieve maximum effectiveness, we have decided to run the tests after the completion of the first prototype that covers all the desired user functions. Tests will stop before each IAPETUS upgrade and resume afterwards, following the proper alterations. Test Suspension / Resumption Criteria: A potential factor for suspending any testing is the implicit application of test-driven software development procedures. Although IAPETUS is not build using this approach, there is always the possibility that through running the tests, new user requirements and/or ways to improve the implementations and the returning results may emerge. To incorporate such new material into IAPETUS would call for the suspension of any testing. This is facilitated by the way that the Django testing framework operates

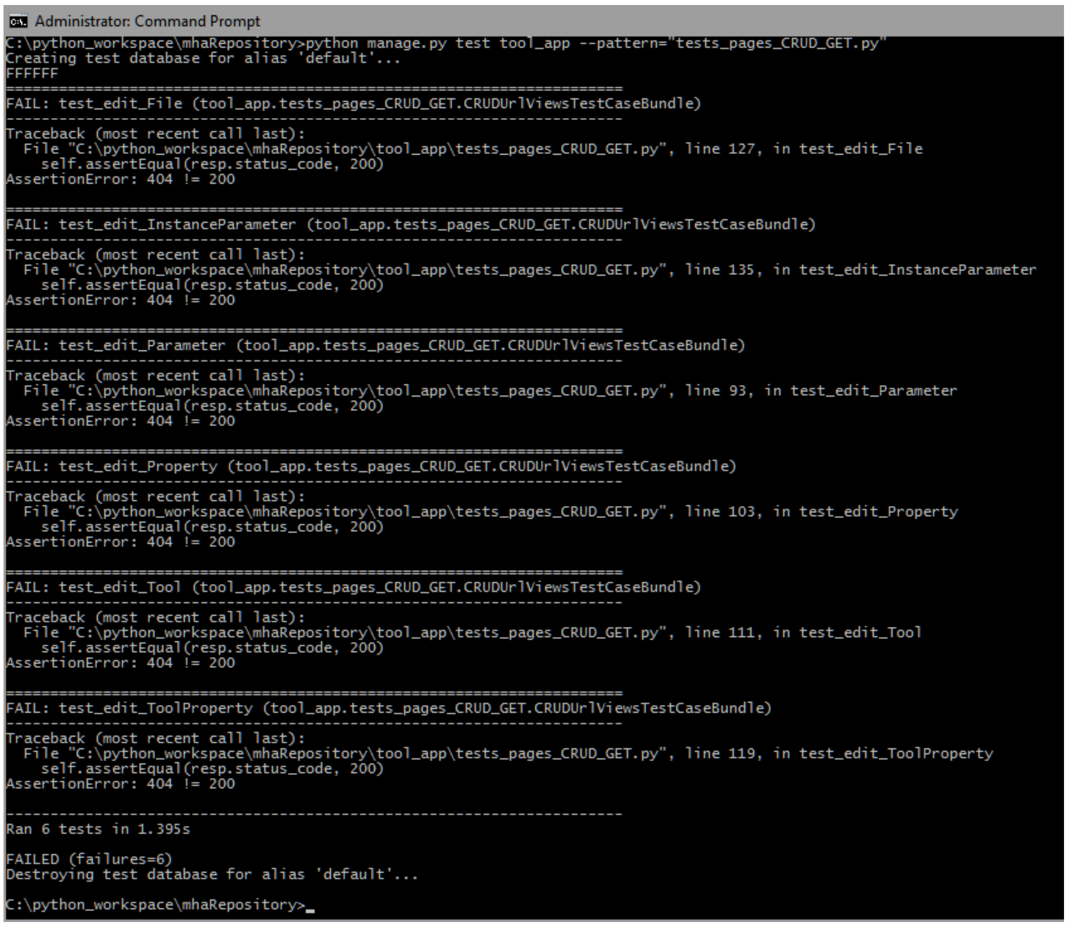
Test Results:

Due to the nature of testing in Django, the only results available to the tester are the command line interface outputs of pass/fail for each of the functions included in the called testing module (python source code file). Indicative examples are given in figures 1 and 2, which describe the testing of some HTTP GET requests from the WP5 tool/model repository of IAPETUS.



```
C:\python_workspace\mhaRepository>python manage.py test tool_app --pattern="tests_pages_CRUD_GET.py"
Creating test database for alias 'default'...
C:\Python27\lib\site-packages\django\shortcuts.py:45: RemovedInDjango110Warning: The context_instance argument of render_to_string is deprecated.
  using=using)

......
----------------------------------------------------------------------
Ran 6 tests in 1.087s

OK
Destroying test database for alias 'default'...

C:\python_workspace\mhaRepository>_
```

Output of successful tests



```
C:\python_workspace\mhaRepository>python manage.py test tool_app --pattern="tests_pages_CRUD_GET.py"
Creating test database for alias 'default'...
FFFFFF
======================================================================
FAIL: test_edit_File (tool_app.tests_pages_CRUD_GET.CRUDUrlViewsTestCaseBundle)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\python_workspace\mhaRepository\tool_app\tests_pages_CRUD_GET.py", line 127, in test_edit_File
    self.assertEqual(resp.status_code, 200)
AssertionError: 404 != 200

======================================================================
FAIL: test_edit_InstanceParameter (tool_app.tests_pages_CRUD_GET.CRUDUrlViewsTestCaseBundle)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\python_workspace\mhaRepository\tool_app\tests_pages_CRUD_GET.py", line 135, in test_edit_InstanceParameter
    self.assertEqual(resp.status_code, 200)
AssertionError: 404 != 200

======================================================================
FAIL: test_edit_Parameter (tool_app.tests_pages_CRUD_GET.CRUDUrlViewsTestCaseBundle)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\python_workspace\mhaRepository\tool_app\tests_pages_CRUD_GET.py", line 93, in test_edit_Parameter
    self.assertEqual(resp.status_code, 200)
AssertionError: 404 != 200

======================================================================
FAIL: test_edit_Property (tool_app.tests_pages_CRUD_GET.CRUDUrlViewsTestCaseBundle)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\python_workspace\mhaRepository\tool_app\tests_pages_CRUD_GET.py", line 103, in test_edit_Property
    self.assertEqual(resp.status_code, 200)
AssertionError: 404 != 200

======================================================================
FAIL: test_edit_Tool (tool_app.tests_pages_CRUD_GET.CRUDUrlViewsTestCaseBundle)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\python_workspace\mhaRepository\tool_app\tests_pages_CRUD_GET.py", line 111, in test_edit_Tool
    self.assertEqual(resp.status_code, 200)
AssertionError: 404 != 200

======================================================================
FAIL: test_edit_ToolProperty (tool_app.tests_pages_CRUD_GET.CRUDUrlViewsTestCaseBundle)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\python_workspace\mhaRepository\tool_app\tests_pages_CRUD_GET.py", line 119, in test_edit_ToolProperty
    self.assertEqual(resp.status_code, 200)
AssertionError: 404 != 200

----------------------------------------------------------------------
Ran 6 tests in 1.395s

FAILED (failures=6)
Destroying test database for alias 'default'...

C:\python_workspace\mhaRepository>_
```

Indicative output of failed tests.

CONFORMITY:	☒ **OK** *(test results in accordance with expected results)*	☐ **NOK** *(test results not in accordance with expected results)* **Comments:**

Test #: IAPETUS tool/model repository		**Type:** IAPETUS tool/model repository– integration test	
Date:	25/5/2016-30/5/2016	**Duration:**	1 hour

Objective: Being the starting point in the Nephroblastoma use case as well as the “core” of the IAPETUS application, it is important to test the tool/model repository’s ability to interact with the tool execution platform, and provide all pertinent data for a chosen model in order to create an ad-hoc form on the IAPETUS wizard.

Necessary hardware: Windows Server Machine

Necessary software: The IAPETUS testing procedure in the context of the Nephroblastoma clinical scenario was the source for the Use Case demonstration as described in D9.2. Combination of Unit tests can provide an execution of the demonstrator through the Django Testing framework, but with different end results (pass/fail, instead of viewing and retrieving stored data and execution results). As an integration test, though, the end user can test the application by its visual outputs as well.

Preconditions: The system is tested with respect to the end user’s inputs and outputs as a whole. Depending on the approach of testing as described in the above section, the criteria to determine the success are either the command line interface outputs as described in unit testing or the visual outputs delivered to the user. In the case of directly user-testing via the execution of the demonstrator, no significant computer skills are required, other than familiarity with web browsers. In the case of individual combined unit testing, the same skills mentioned in the corresponding section of unit testing are required.

Set-up: Both modules of IAPETUS were tested for the Nephroblastoma Use case. The Wilms Oncosimulator is uploaded to the tool/model repository (basic descriptive information, a set of files and a set of parameters). These data are used by IAPETUS’s tool execution platform, to produce the necessary input form for the Oncosimulator.

Test Steps:

Step 1: Use IAPETUS GUIs to upload the Wilms Oncosimulator data and info.

Step 2: Retrieve files and data as needed for model execution by the tool execution platform

Expected Test Results:

After building a dedicated python file with a sequence of the individual unit tests, in the Django testing console (Windows command prompt), a successful test screen should be produced. On their side, the end users should be able to view their uploaded results and when calling the Oncosimulation module of IAPETUS, the proper form must be created.

Test Pass / Fail Criteria

According to the scenario's development and described demonstrator, the test is successful if the following conditions are met:

1. The stored Oncosimulator data can be viewed by the user and retrieved by the tool execution platform.

2. Upon choosing a patient, the proper input form is created from the model's stored parameter information

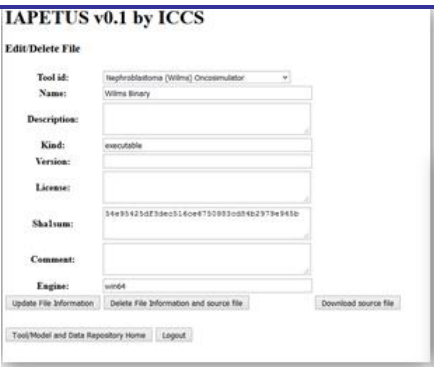
As it pertains to a combined unit testing, the standard assertion mechanism can be utilized.

Test Entry / Exit Criteria

Similar to the individual unit testing, the Nephroblastoma clinical scenario can be executed anytime with no special preparations. To achieve maximum effectiveness, we have decided to run the tests after the completion of the first prototype that covers all the desired user functions. Tests will stop before each IAPETUS upgrade and resume afterwards, following the proper alterations.

Test Results:

Results of the stored Oncosimulator data in the tool/model repository

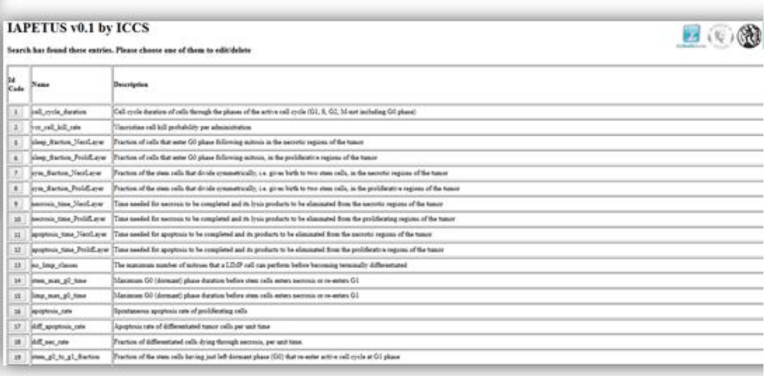


IAPETUS v0.1 by ICCS

Edit/Delete File

Tool id:	Nephroblastoma (Wilms) Oncosimulator
Name:	Wilms Binary
Description:	
Kind:	executable
Version:	
License:	
Sha1sum:	54e95425df3dec516ce4750983cd84b2979e945b
Comment:	
Engine:	win64

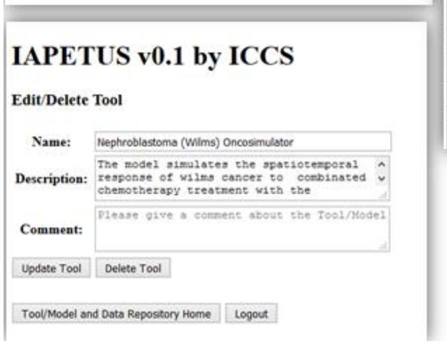
Update File Information
Delete File Information and source file
Download source file
Tool/Model and Data Repository Home
Logout



IAPETUS v0.1 by ICCS

Search has found these entries. Please choose one of them to edit/delete

Id Code	Name	Description
1	cell_cycle_duration	Cell cycle duration of cells through the phases of the active cell cycle (G1, S, G2, M-not including G0 phase)
2	vcr_cell_kill_rate	Vincristine cell kill probability per administration
5	sleep_fraction_NecrLayer	Fraction of cells that enter G0 phase following mitosis in the necrotic regions of the tumor
6	sleep_fraction_ProlifLayer	Fraction of cells that enter G0 phase following mitosis, in the proliferative regions of the tumor
7	sym_fraction_NecrLayer	Fraction of the stem cells that divide symmetrically, i.e. gives birth to two stem cells, in the necrotic regions of the tumor
8	sym_fraction_ProlifLayer	Fraction of the stem cells that divide symmetrically, i.e. gives birth to two stem cells, in the proliferative regions of the tumor
9	necrosis_time_NecrLayer	Time needed for necrosis to be completed and its lysis products to be eliminated from the necrotic regions of the tumor
10	necrosis_time_ProlifLayer	Time needed for necrosis to be completed and its lysis products to be eliminated from the proliferating regions of the tumor
11	apoptosis_time_NecrLayer	Time needed for apoptosis to be completed and its products to be eliminated from the necrotic regions of the tumor
12	apoptosis_time_ProlifLayer	Time needed for apoptosis to be completed and its products to be eliminated from the proliferative regions of the tumor
13	no_limp_classes	The maximum number of mitoses that a LIMP cell can perform before becoming terminally differentiated
14	stem_max_g0_time	Maximum G0 (dormant) phase duration before stem cells enters necrosis or re-enters G1
15	limp_max_g0_time	Maximum G0 (dormant) phase duration before stem cells enters necrosis or re-enters G1
16	apoptosis_rate	Spontaneous apoptosis rate of proliferating cells
17	diff_apoptosis_rate	Apoptosis rate of differentiated tumor cells per unit time
18	diff_nec_rate	Fraction of differentiated cells dying through necrosis, per unit time
19	stem_g0_to_g1_fraction	Fraction of the stem cells having just left dormant phase (G0) that re-enter active cell cycle at G1 phase

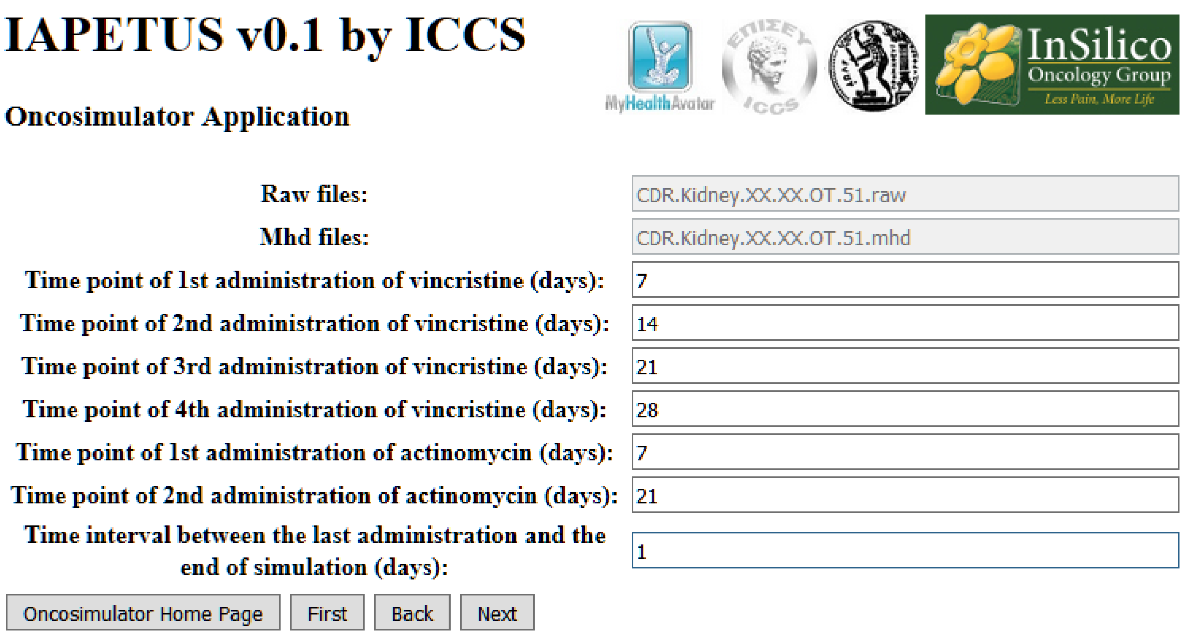


IAPETUS v0.1 by ICCS

Edit/Delete Tool

Name:	Nephroblastoma (Wilms) Oncosimulator
Description:	The model simulates the spatiotemporal response of wilms cancer to combinated chemotherapy treatment with the
Comment:	Please give a comment about the Tool/Model

Input form for the Nephroblastoma Oncosimulator. The variable names (in the left column), are retrieved from the tool/model repository



IAPETUS v0.1 by ICCS

Oncosimulator Application

Raw files:	CDR.Kidney.XX.XX.OT.51.raw
Mhd files:	CDR.Kidney.XX.XX.OT.51.mhd
Time point of 1st administration of vincristine (days):	7
Time point of 2nd administration of vincristine (days):	14
Time point of 3rd administration of vincristine (days):	21
Time point of 4th administration of vincristine (days):	28
Time point of 1st administration of actinomycin (days):	7
Time point of 2nd administration of actinomycin (days):	21
Time interval between the last administration and the end of simulation (days):	1

CONFORMITY:	☒ OK *(test results in accordance with expected results)*	☐ NOK *(test results not in accordance with expected results)* **Comments:**

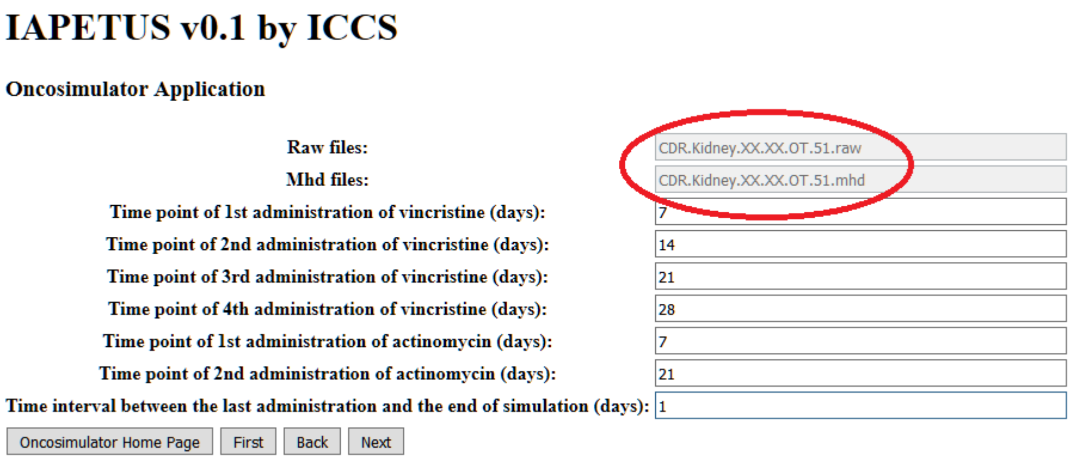
Test #: IAPETUS tool execution platform – Communication with 3rd party APIs		**Type:** IAPETUS tool execution platform – Communication with 3rd party APIs – integration test	
Date:	25/5/2016-30/5/2016	**Duration:**	1 hour
Objective: For this step, integration was achieved by API communications between IAPETUS, the MHA platform and the CHIC repository. Potential risks include the exchange of data which would alter the stored data of all three parties and security issues n communications. These issues are resolved by the involved application's APIs, which make sure that the correct forms of data are exchanged and their authentication mechanisms along with the use of the https protocol which prevent unauthorized access**.**			
Necessary hardware: Windows Server Machine			
Necessary software: Django Testing framework, Tastypie, reportlab, ImageJ and Gnuplot (for result visualization)			
Preconditions: The system is tested with respect to the end user's inputs and outputs as a whole. Depending on the approach of testing as described in the above section, the criteria to determine the success are either the command line interface outputs as described in unit testing or the visual outputs delivered to the user. In the case of directly user-testing via the execution of the demonstrator, no significant computer skills are required, other than familiarity with web browsers. In the case of individual combined unit testing, the same skills mentioned in the corresponding section of unit testing are required.			
Set-up: Both modules of IAPETUS were tested for the Nephroblastoma clinical scenario. The "border" between the two steps is the Oncosimulator input page. The user input data along with information from the CHIC data repository and MHA platform are utilized to produce treatment simulation results and store them back to the MHA platform.			
Test Steps:			
Step 1: Retrieve from CHIC repository the synthetic patient's imaging data to be used for the Oncosimulator's execution			
Step 2: Interact with the MHA platform to store and retrieve Oncosimulator execution result reports			
Expected Test Results: Similar to the previous step, there is a python file which contains the composing unit tests. The users can determine the test's success from the visual results of the Oncosimulator and the storing and download the report from MHA platform. **Test Pass / Fail Criteria**			

According to the scenario's development and described demonstrator, the test is successful if the following conditions are met:
1. Having chosen a patient, the proper input files are delivered from the CHIC data repository and shown in the GUI
2. After the Oncosimulator Execution the user can view the results
3. After storing the results to the MHA platform, the user should be able to download them and view it with their PDF browser.
As it pertains to a potential combined unit testing, the standard assertion mechanism can be utilized.

Test Entry / Exit Criteria

Similar to the individual unit testing, the Nephroblastoma clinical scenario can be executed anytime with no special preparations. To achieve maximum effectiveness, we have decided to run the tests after the completion of the first prototype that covers all the desired user functions. Tests will stop before each IAPETUS upgrade and resume afterwards, following the proper alterations.

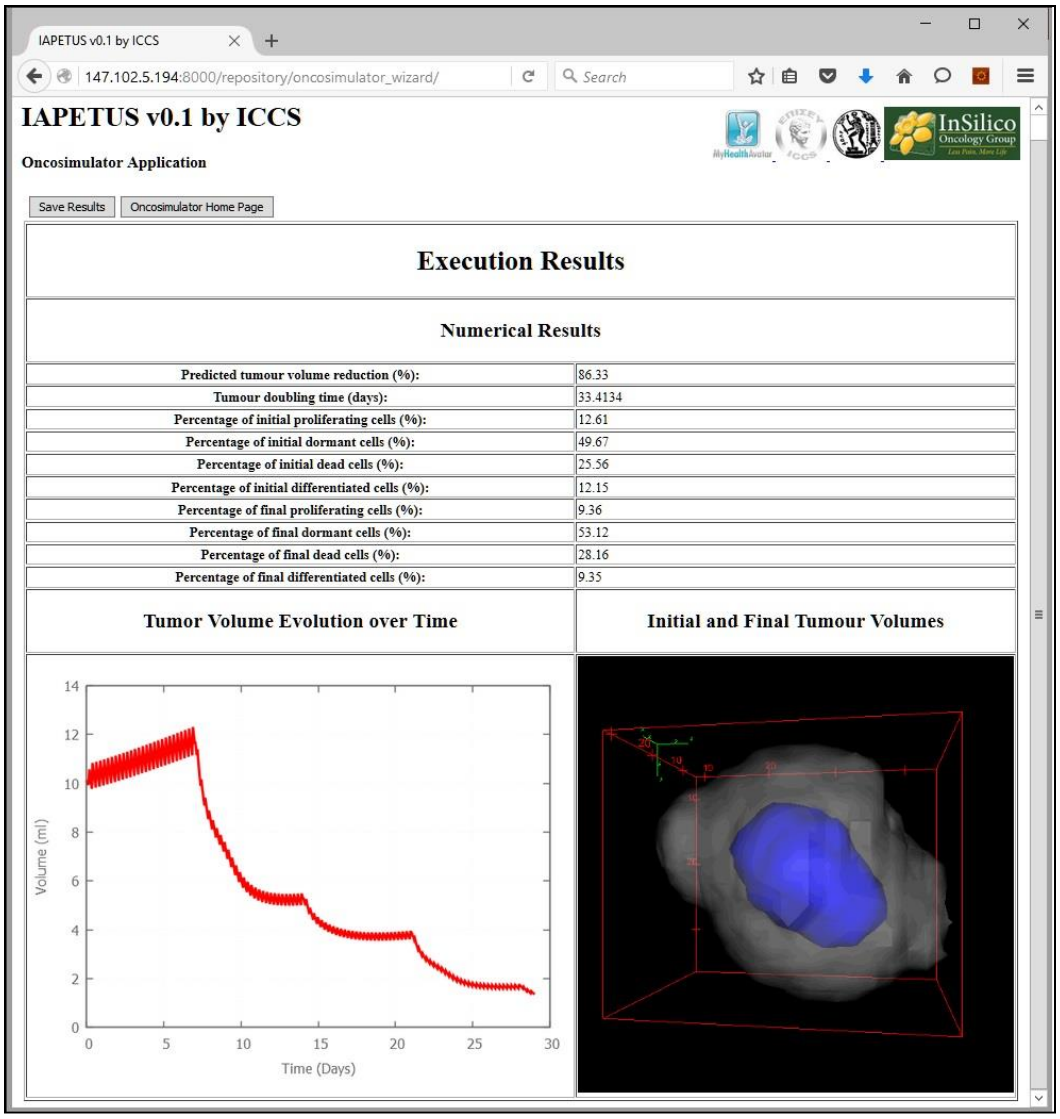
Test Results:


IAPETUS v0.1 by ICCS

Oncosimulator Application

Raw files:	CDR.Kidney.XX.XX.OT.51.raw
Mhd files:	CDR.Kidney.XX.XX.OT.51.mhd
Time point of 1st administration of vincristine (days):	7
Time point of 2nd administration of vincristine (days):	14
Time point of 3rd administration of vincristine (days):	21
Time point of 4th administration of vincristine (days):	28
Time point of 1st administration of actinomycin (days):	7
Time point of 2nd administration of actinomycin (days):	21
Time interval between the last administration and the end of simulation (days):	1

Input form for the Nephroblastoma Oncosimulator. The raw imaging files (in the red circle), are downloaded from the CHIC repository, after providing the latter with the patient's ID number



IAPETUS v0.1 by ICCS

Oncosimulator Application

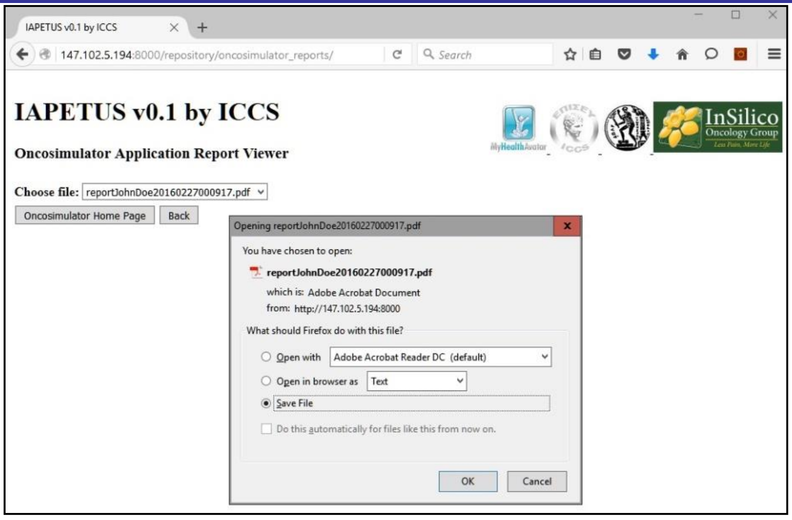
Execution Results

Numerical Results

Predicted tumour volume reduction (%):	86.33
Tumour doubling time (days):	33.4134
Percentage of initial proliferating cells (%):	12.61
Percentage of initial dormant cells (%):	49.67
Percentage of initial dead cells (%):	25.56
Percentage of initial differentiated cells (%):	12.15
Percentage of final proliferating cells (%):	9.36
Percentage of final dormant cells (%):	53.12
Percentage of final dead cells (%):	28.16
Percentage of final differentiated cells (%):	9.35

Tumor Volume Evolution over Time | **Initial and Final Tumour Volumes**

Oncosimulator results



IAPETUS v0.1 by ICCS

147.102.5.194:8000/repository/oncosimulator_reports/

IAPETUS v0.1 by ICCS

Oncosimulator Application Report Viewer

Choose file: reportJohnDoe20160227000917.pdf

Oncosimulator Home Page | Back

Opening reportJohnDoe20160227000917.pdf

You have chosen to open:

reportJohnDoe20160227000917.pdf

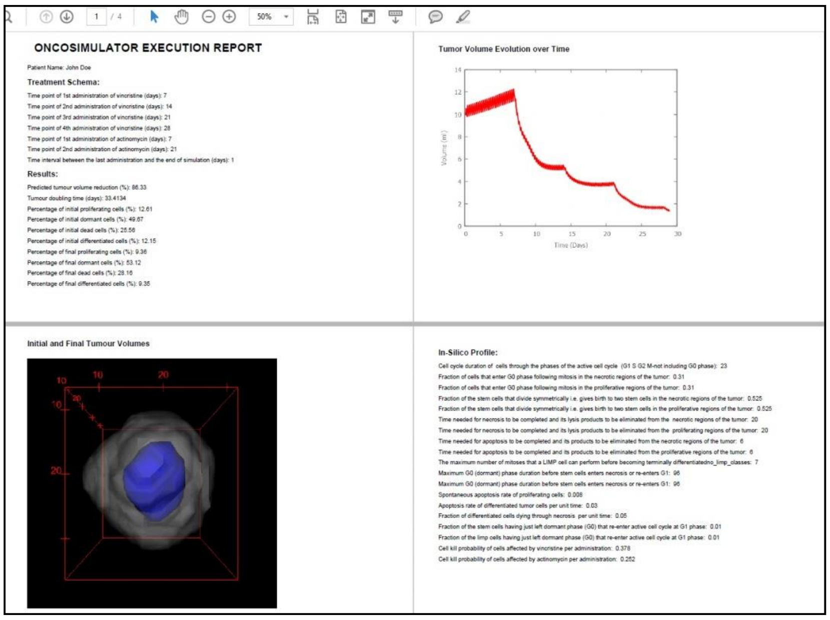
which is: Adobe Acrobat Document
from: http://147.102.5.194:8000

What should Firefox do with this file?

- ○ Open with: Adobe Acrobat Reader DC (default)
- ○ Open in browser as: Text
- [✓ Save File]
- ☐ Do this automatically for files like this from now on.

OK | Cancel

Result report download



ONCOSIMULATOR EXECUTION REPORT

Patient Name: John Doe

Treatment Schema:

Time point of 1st administration of vincristine (days): 7
Time point of 2nd administration of vincristine (days): 14
Time point of 3rd administration of vincristine (days): 21
Time point of 4th administration of vincristine (days): 28
Time point of 1st administration of actinomycin (days): 7
Time point of 2nd administration of actinomycin (days): 21
Time interval between the last administration and the end of simulation (days): 1

Results:

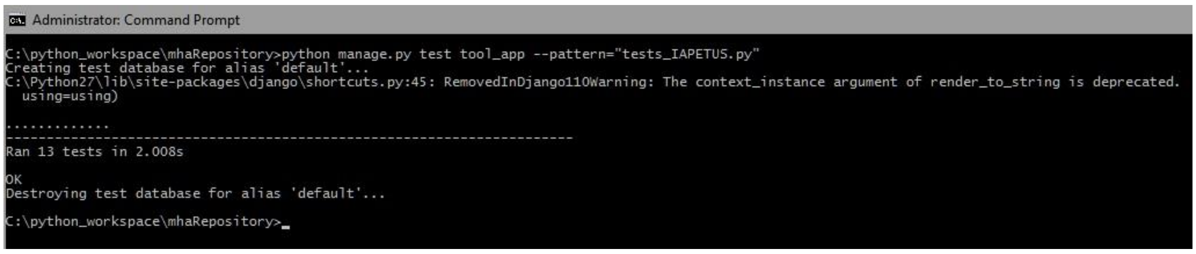
Predicted tumour volume reduction (%): 86.33
Tumour doubling time (days): 33.4134
Percentage of initial proliferating cells (%): 12.61
Percentage of initial dormant cells (%): 49.67
Percentage of initial dead cells (%): 25.56
Percentage of initial differentiated cells (%): 12.15
Percentage of final proliferating cells (%): 9.36
Percentage of final dormant cells (%): 53.12
Percentage of final dead cells (%): 28.16
Percentage of final differentiated cells (%): 9.35

In-Silico Profile:

Cell cycle duration of cells through the phases of the active cell cycle (G1 S G2 M-not including G0 phase): 23
Fraction of cells that enter G0 phase following mitosis in the necrotic regions of the tumor: 0.31
Fraction of cells that enter G0 phase following mitosis in the proliferative regions of the tumor: 0.31
Fraction of the stem cells that divide symmetrically i.e. gives birth to two stem cells in the necrotic regions of the tumor: 0.525
Fraction of the stem cells that divide symmetrically i.e. gives birth to two stem cells in the proliferative regions of the tumor: 0.525
Time needed for necrosis to be completed and its lysis products to be eliminated from the necrotic regions of the tumor: 20
Time needed for necrosis to be completed and its lysis products to be eliminated from the proliferating regions of the tumor: 20
Time needed for apoptosis to be completed and its products to be eliminated from the necrotic regions of the tumor: 6
Time needed for apoptosis to be completed and its products to be eliminated from the proliferative regions of the tumor: 6
The maximum number of mitoses that a LIMP cell can perform before becoming terminally differentiatedno_limp_classes: 7
Maximum G0 (dormant) phase duration before stem cells enters necrosis or re-enters G1: 96
Maximum G0 (dormant) phase duration before stem cells enters necrosis or re-enters G1: 96
Spontaneous apoptosis rate of proliferating cells: 0.008
Apoptosis rate of differentiated tumor cells per unit time: 0.03
Fraction of differentiated cells dying through necrosis per unit time: 0.05
Fraction of the stem cells having just left dormant phase (G0) that re-enter active cell cycle at G1 phase: 0.01
Fraction of the limp cells having just left dormant phase (G0) that re-enter active cell cycle at G1 phase: 0.01
Cell kill probability of cells affected by vincristine per administration: 0.378
Cell kill probability of cells affected by actinomycin per administration: 0.252

1 / 4
50%
Tumor Volume Evolution over Time
Volume (ml)
Time (Days)
0
5
10
15
20
25
30
Initial and Final Tumour Volumes
10
20

PDF result report



Administrator: Command Prompt

```
C:\python_workspace\mhaRepository>python manage.py test tool_app --pattern="tests_IAPETUS.py"
Creating test database for alias 'default'...
C:\Python27\lib\site-packages\django\shortcuts.py:45: RemovedInDjango110Warning: The context_instance argument of render_to_string is deprecated.
  using=using)

.............
----------------------------------------------------------------------
Ran 13 tests in 2.008s

OK
Destroying test database for alias 'default'...

C:\python_workspace\mhaRepository>_
```

Result of running a testing sequence of multiple unit tests for IAPETUS

CONFORMITY:	☒ OK *(test results in accordance with expected results)*	☐ NOK *(test results not in accordance with expected results)* **Comments:**

Performance Testing

The most significant issue in the performance test of IAPETUS is that the final results have to be correlated with the average patient-doctor session. Given that the average visit time ranges from 10 to 25 minutes, it is important that the doctor is left with plenty of time to evaluate the results, run more executions and determine the proper treatment strategy.

Item to Test	Test Description	Test Date	Responsibility
Wilms Oncosimulator Executable	Measurement of the Wilms Oncosimulator execution time called by IAPETUS and the formulation of results. Cross-reference with execution times in command line interface	4/2016-5/2016	Testing conducted by IAPETUS developers
Tool execution module	Measurement of the time from logging in to the retrieval of the stored results from the MHA platform		
Tool execution engine	In the indicative case of up to three simultaneous Oncosimulator executions, measurement of the speed-up percentage against a set of serial executions		

These tests are a simple case of time measurements. The developers can easily include a few lines of code to determine the time duration of a component's execution. These execution times are produced in the result backend command line console (handles the django views output and by extension, the output of the called Oncosimulator model).

A "loose" criterion mentioned above is that the execution times must be low enough to be only a small fraction of the doctor-patient session duration (avg. 17,5 min). As these tests have to do with execution times, there are no strictly defined pass/fail boundaries. The goal is to keep these times as low as possible. These tests actually require the components to run as usual, so there are no specific criteria to start/stop the tests.

Test results: the following table has been produced with all the above described measurements:

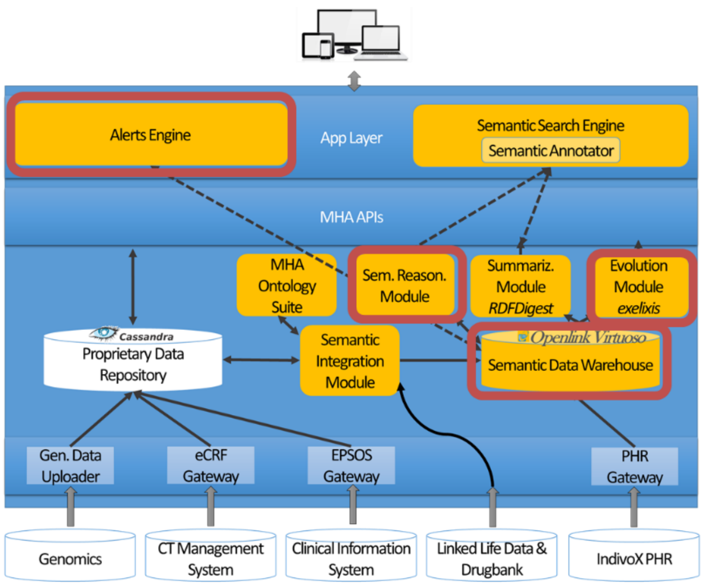
Number of executions / Execution mode	Single execution	2 simultaneous executions	3 simultaneous executions

Command prompt only	19.12s	20.43s	21.81s
Using the infrastructure (with execution engine)	25.67s	28.05s	35.64s
Using the infrastructure (without execution engine)	26.82s	53.62s	1min 20.88s
Workflow 2 overall time (login to report retrieval, single execution only)	1min 16.74s	1min 19.12s (estimated)	1min 26.71s (estimated)

The average time to produce the graph and the gif image was measured to be 7.48 seconds. Looking at the table, it can be deducted that the engine helps the infrastructure produce results 48% and 56% faster for two and three simultaneous executions respectively

3.5.5.2 MyHealthAvatar Semantic Layer /Clinical data unit/integration tests

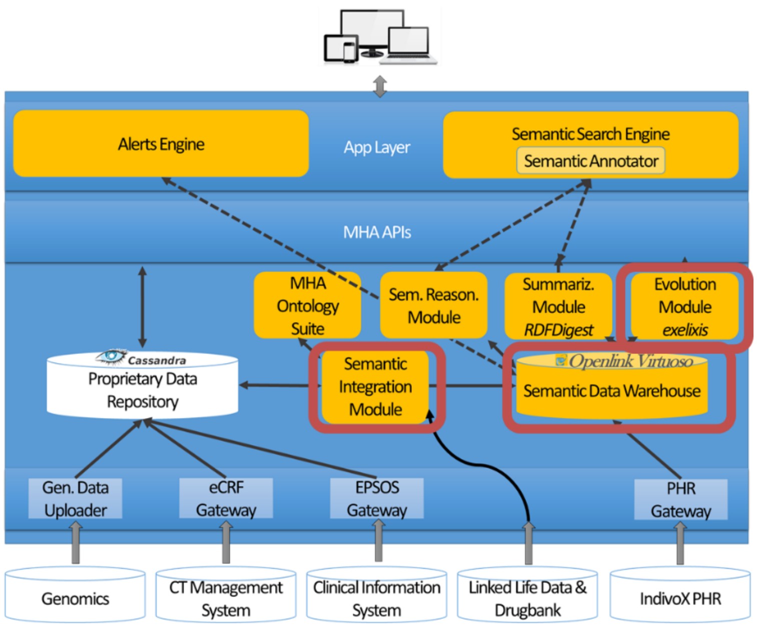
Test #: Alers1	**Type: Alerts Engine -** Integration test
Date: 20/10/2015	**Duration:** 5 minutes
Objective: To identify if the proper alerts are generated exploiting a) guideline rules specified and b) the integrated patient profile.	
Necessary hardware: Windows Server Machine	
Necessary software: Virtuoso Triple Store, Jena Reasoning Engine, Alert's Engine, MyHealthAvatar app for visualization of clinical information	
Preconditions: To have guideline rules specified. To have the integrated patient profile at the Virtuoso	

Set-up: The components of the semantic layer participating in Alerts Test used in this scenario  Alerts Engine App Layer Semantic Search Engine Semantic Annotator MHA APIs MHA Ontology Suite Sem. Reason. Module Summariz. Module RDFDigest Evolution Module exelixis Cassandra Proprietary Data Repository Semantic Integration Module Openlink Virtuoso Semantic Data Warehouse Gen. Data Uploader eCRF Gateway EPSOS Gateway PHR Gateway Genomics CT Management System Clinical Information System Linked Life Data & Drugbank IndivoX PHR
Test Steps:
Step 1: Log in into the MyHealthAvatar portal
Step 2: Go to View of Clinical Information
Step 3: Identify the proper alert issued
Expected Test Results: An alert is issued visualized using the aforementioned alert.
Test Results: Each time executed, the proper alert was generated in a timely fashion.

CONFORMITY:	☐ **OK** *(test results in accordance with expected results)*	☐ **NOK** *(test results not in accordance with expected results)* **Comments:**

Test #: Evolution1	**Type: Evolution Module -** Integration test
Date: 20/10/2015	**Duration:** 5 minutes

Objective: To identify if the evolution module works appropriately answering queries over data mapped with different ontology versions
Necessary hardware: Windows Server Machine
Necessary software: Virtuoso Triple Store, Evolution Module, Semantic Integration Module

Preconditions: To have established the mappings with two example tables from the data lake (the Cassandra data repository) using two different ontology versions
Set-up: The components of the semantic layer participating in Evolution test are :  App Layer Alerts Engine Semantic Search Engine Semantic Annotator MHA APIs MHA Ontology Suite Sem. Reason. Module Summariz. Module *RDFDigest* Evolution Module *exelixis* Cassandra Proprietary Data Repository Semantic Integration Module Openlink Virtuoso Semantic Data Warehouse Gen. Data Uploader eCRF Gateway EPSOS Gateway PHR Gateway Genomics CT Management System Clinical Information System Linked Life Data & Drugbank IndivoX PHR
Test Steps:
Step 1: Go to interface for querying the Semantic Data Warehouse (or use the corresponding web service)
Step 2: Send a query to the evolution module involving two tables mapped with different ontology version
Step 3: Identify that data from both these tables have been retrieved.
Expected Test Results: The results returned from all tables independent of the ontology version they use.
Test Results: Each time executed, the proper results were generated

CONFORMITY:	☐ **OK** *(test results in accordance with expected results)*	☐ **NOK** *(test results not in accordance with expected results)* **Comments:**

Test #: ETL	**Type: ETL -** Integration test
Date: 20/10/2015	**Duration:** 5 minutes

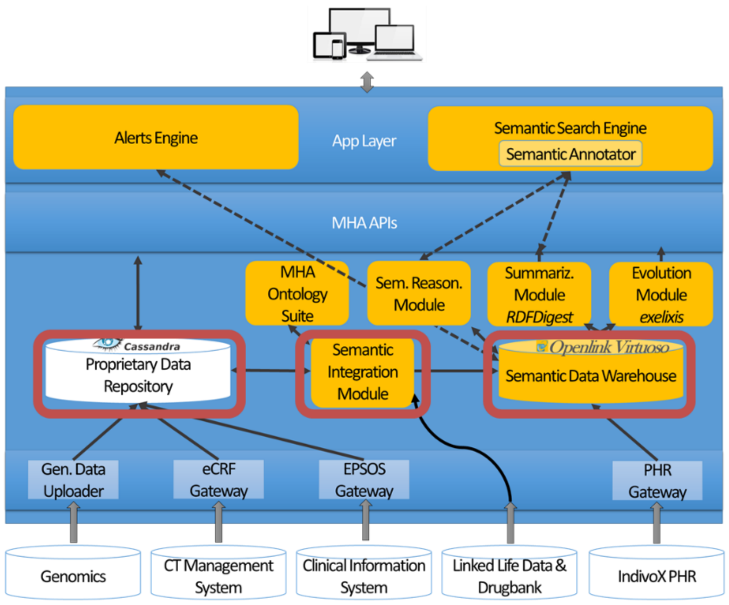
Objective: To identify whether data can be extracted from the data lake (Cassandra repository) transformed and loaded to the Triple Store

Necessary hardware: Windows Server Machine

Necessary software: Data Integration Module, Cassandra Data Repository, Semantic Data Warehouse.

Preconditions: The proper mapping should be available describing data using terms from the ontology

Set-up: The components of the semantic layer participating in ETL test are :



Test Steps:

Step 1: Go to Semantic Integration Module

Step 2: Load the generated mappings

Step 3: execute the ETL

Expected Test Results: Transformed linked data should be available at the triple store corresponding to the data available at the Cassandra repository

Test Results: Each time executed, the proper results were generated

CONFORMITY:	☐ **OK** *(test results in accordance with expected results)*	☐ **NOK** *(test results not in accordance with expected results)* **Comments:**

Test #: Clinical PS API	**Type: Epsos Gateway -** Integration Test
Date: 20/10/2015	**Duration:** 5 minutes

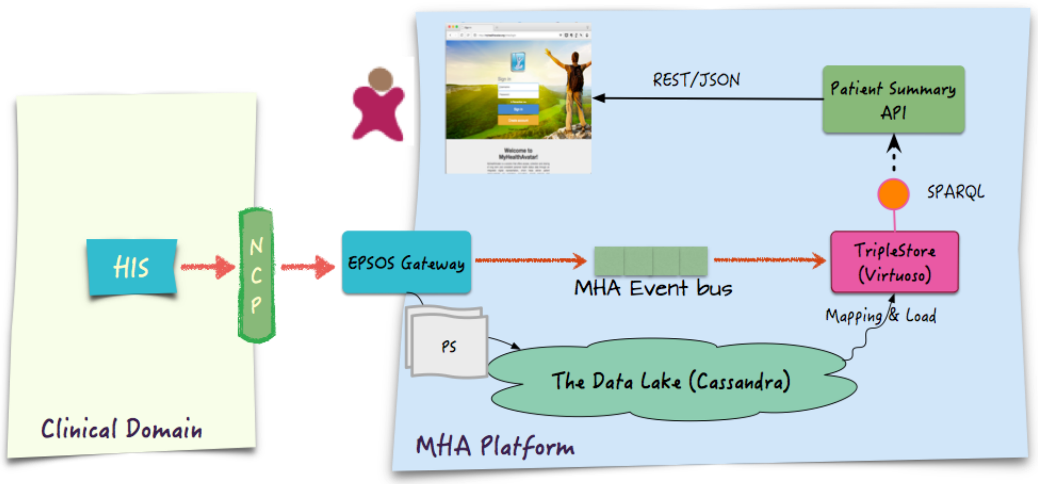
Objective: To identify Clinical PS API is able to retrieve from the virtuoso patient clinical data for MHA platform and services

Necessary hardware: Windows Server Machine

Necessary software: Clinical Information System, MHA Epsos Gateway, MHA Clinical Gateway event bus

Preconditions: The user must consent to allow retrieval of data from a hospital health record giving his SSN or other identifier

Set-up: The components for linking with clinical information systems tests are :



Test Steps:

Step 1: Go to the Clinical Information System and update patient information

Step 2: View that the even bus on MHA has received the information

Step 3: Update MHA repository with the updated patient summaries

Expected Test Results: Clinical data to retrieved, shown on MHA portal, and are accessible through MHA API for third party applications

Test Results: Each time executed, the proper results were generated

CONFORMITY:	☐ **OK** *(test results in accordance with expected results)*	☐ **NOK** *(test results not in accordance with expected results)* **Comments:**

Test #: SemanticSearchEngine | **Type: Semantic Search Engine -** Unit test

Date: 20/10/2015 | **Duration:** 5 minutes

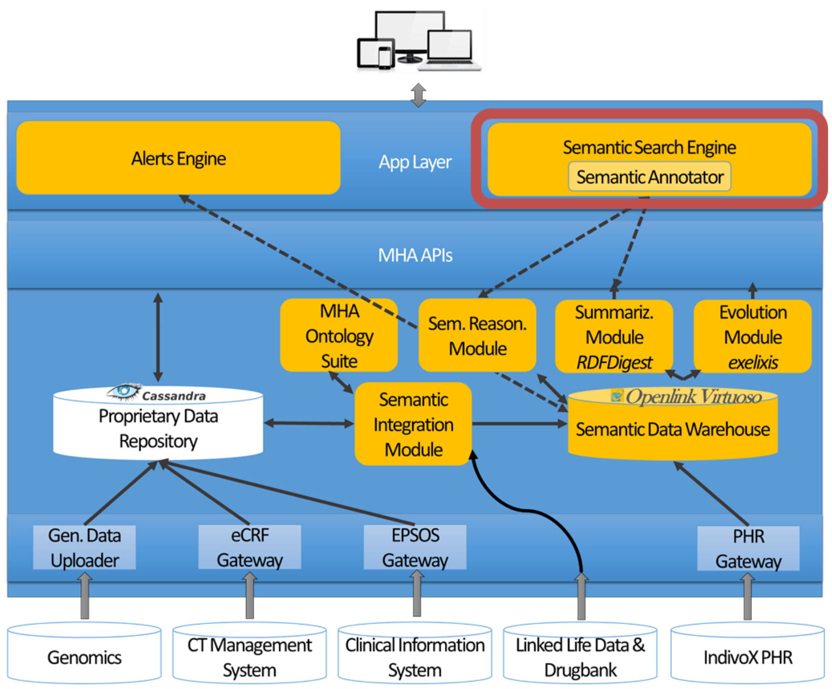
Objective: To identify whether results are returned after user queries are issued.

Necessary hardware: Windows Server Machine

Necessary software: Semantic Search Engine

Preconditions: There should be available, indexed documents at the document repository.

Set-up: The components used of the semantic layer participating in Semantic Search Test are:



Test Steps:

Step 1: Log in to the MyHealthAvatar portal

Step 2: Go to Tools → Medical Documents Semantic Search

Step 3: Enter a search keyword

Expected Test Results: results relevant to the keywords used by the user

Test Results: Each time executed, the proper results were generated

CONFORMITY:	☐ **OK** *(test results in accordance with expected results)*	☐ **NOK** *(test results not in accordance with expected results)* **Comments:**

Test #: SummarizationEngine	**Type: SummarizationModule -** Unit test
Date: 20/10/2015	**Duration:** 5 minutes

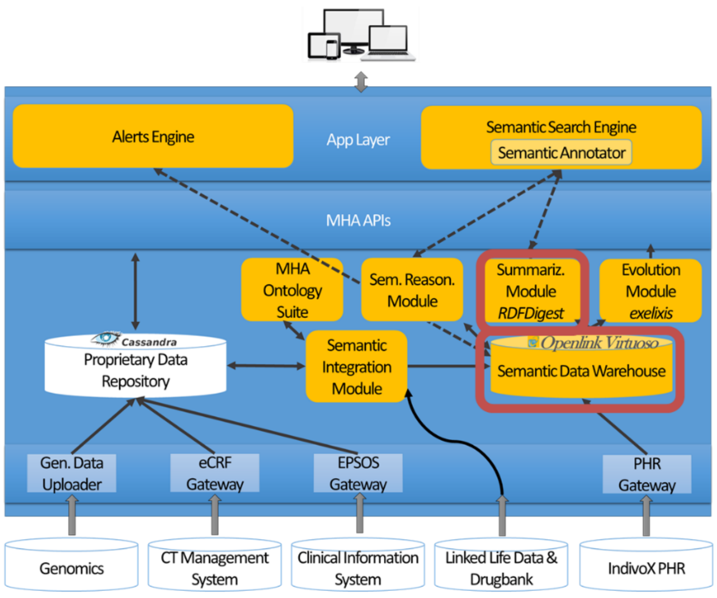
Objective: To identify whether summaries can be created for input data efficiently and effectively

Necessary hardware: Windows Server Machine

Necessary software: Summarization Module

Preconditions:

Set-up: The components of the semantic layer participating in Summarization Test are :



Test Steps:

Step 1: Go to the Summarization Module

Step 2: Load an ontology or give the url of a SPARQL endpoint

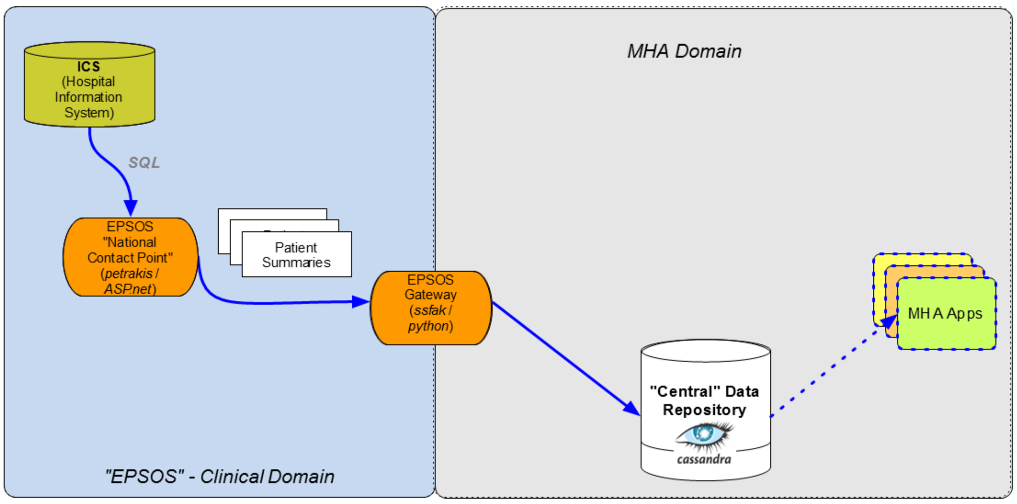
Step 3: Select the size of the summary

Step 4: Generate summary

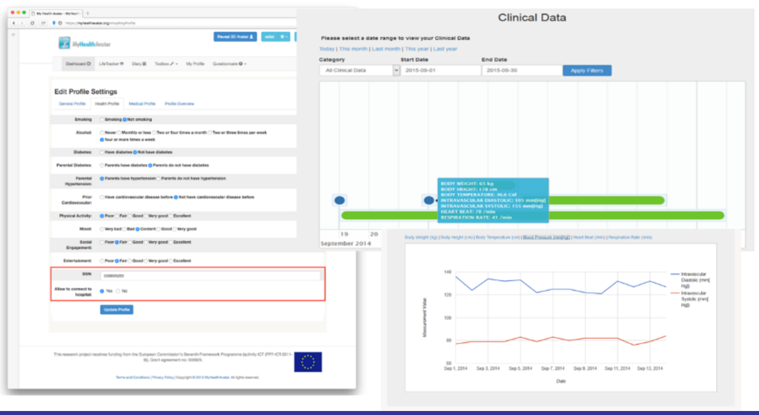
Expected Test Results: A summary should be generated in a timely fashion presenting an overview of the input data

Test Results: Each time executed, the proper results were generated

CONFORMITY:	▯ **OK** *(test results in accordance with expected results)*	▯ **NOK** *(test results not in accordance with expected results)* **Comments:**

Test #: EPSOS Gateway	**Type: Epsos Gateway -** Unit test
Date: 20/10/2015	**Duration:** 10 minutes
Objective: To identify whether patients summaries can be retrieved through a clinical information system whenever new information are available	
Necessary hardware: Windows Server Machine	
Necessary software: Clinical Information System, MHA Epsos Gateway, MHA Clinical Gateway event bus	
Preconditions: The user must consent to allow retrieval of data from a hospital health record giving his SSN or other identifier	
Set-up: The components for linking with clinical information systems tests are :  ICS (Hospital Information System) SQL EPSOS "National Contact Point" (petrakis / ASPnet) Patient Summaries EPSOS Gateway (ssfak / python) MHA Domain "Central" Data Repository cassandra MHA Apps "EPSOS" - Clinical Domain	
Test Steps:	
Step 1: Go to the Clinical Information System and update patient information	
Step 2: View that the even but on MHA has received the information	
Step 3: Update MHA repository with the updated patient summaries	
Expected Test Results: Clinical data to be updated on MHA portal	

Test Results:	Each time executed, the proper results were generated	
CONFORMITY:	☐ **OK** *(test results in accordance with expected results)*	☐ **NOK** *(test results not in accordance with expected results)* **Comments:**

Test #: Clinical Data View Test	**Type:** Clinical Data View Test – unit test
Date: 26/5/2016	**Duration:** 5 minutes
Objective: Test the view of Clinical Data in the MHA platform	
Necessary hardware: PC/laptop or mobile device	
Necessary software: HTML5 compatible web browser (Firefox, Chrome etc)	
Preconditions: User must be connected to internet and have a user account to login to MHA platform	
Set-up: Go to MHA page and go to the specific location of the clinical data to view them 	
Test Steps:	
Step 1: Login with user credentials to MHA platform	
Step 2: Click on the green “Data Input” Button on the top of the page.	
Step 3: Click on the “Clinical Data” tab. Clinical API is called and returns the available clinical data for the logged in user.	
Step 4: View clinical data as Active problems, Vital Signs, Drugs, Alerts, Medical Images	
Expected Test Results: User should see the clinical data with a search form to filter results by start date and end date.	
Test Results: User can view and filter his clinical data.	User can view and filter his clinical data.

Edit Profile Settings

Clinical Data

Please select a date range to view your Clinical Data

Today | This month | Last month | This year | Last year

Category	Start Date	End Date
All Clinical Data	2015-09-01	2015-09-30

BODY WEIGHT: 65 kg
BODY HEIGHT: 178 cm
BODY TEMPERATURE: 36.6 Cel
INTRAVASCULAR DIASTOLIC: 105 mm[Hg]
INTRAVASCULAR SYSTOLIC: 155 mm[Hg]
HEART BEAT: 78 /min
RESPIRATION RATE: 41 /min

Apply Filters
Update Profile
September 2014
Measurement Value
Date
Intravascular Diastolic (mm[Hg])
Intravascular Systolic (mm[Hg])

CONFORMITY:	☒ **OK** *(test results in accordance with expected results)*	☐ **NOK** *(test results not in accordance with expected results)* **Comments:**

Test #: Uploading DICOM	**Type:** Uploading DICOM Images Test – unit test

Date:	26/5/2016	**Duration:**	5 minutes

Objective: Test the DICOM Images Uploading in the MHA DICOM repository
Necessary hardware: PC/laptop or mobile device
Necessary software: HTML5 compatible web browser (Firefox, Chrome etc)
Preconditions: User must be connected to internet and have a user account to login to MHA platform, has a DICOM image to upload to MHA DICOM repository
Set-up: Go to MHA page and go to the specific location of the DICOM data to view them
Test Steps:
Step 1: Login with user credentials to MHA platform
Step 2: Click on the green “Data Input” Button on the top of the page.
Step 3: Click on the “Medical Images Upload” tab.
Step 4: Drag and drop medical images series with DICOM files and upload the selected medical images files. Clinical API is called and uploads the selected medical images files.
Expected Test Results: Uploading of medical images must complete.

Test Results:	User can upload his medical images examinations

CONFORMITY:	☒ **OK** *(test results in accordance with expected results)*	☐ **NOK** *(test results not in accordance with expected results)* **Comments:**

Test #: Viewing and Downloading DICOM images		**Type:** Viewing and Downloading DICOM Images Test – unit test	
Date:	26/5/2016	**Duration:**	5 minutes
Objective: Test the DICOM Images Uploading in the MHA platform			
Necessary hardware: PC/laptop or mobile device			
Necessary software: HTML5 compatible web browser (Firefox, Chrome etc)			
Preconditions: User must be connected to internet and have a user account to login to MHA platform			
Set-up: Go to MHA page and specific location to view/download an image			
Test Steps:			
Step 1: Login with user credentials to MHA platform			
Step 2: Click on the green "Data Input" Button on the top of the page.			
Step 3: Click on the "Medical Images View" tab.			
Step 4: A list with medical images appears. It has the functionality of viewing a preview thumbnail for every study as well as downloading the wanted series. The Clinical API is being called and returns the available medical images for the logged in user.			
Expected Test Results: Viewing and downloading available medical images from the list.			
Test Results:	User can view/preview and download his medical DICOM images.		
CONFORMITY:	[✓ ☒] **OK** *(test results in accordance with expected results)*	☐ **NOK** *(test results not in accordance with expected results)* **Comments:**	

3.5.6 Review of MyHealthAvatar architecture

MyHealthAvatar defined a solution for access, collection and sharing of long term and consistent personal health status data through an integrated environment, which will allow more sophisticated clinical data analysis, prediction, prevention and *in silico* treatment simulations tailored to the individual citizen. The technical architecture of the MHA, described in Deliverable D3.2 and D3.2 version 2, is able to efficient support the collection of information for the short and long term management of integrated citizen-specific data, effective access mechanisms for data sharing and data analysis using specialized toolboxes. We have adopted the provisions of the IEEE 1471 standard that defines an architecture as "*…the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution*".

MHA technical infrastructure provides mechanisms for data management in internal data repositories storing individual data for the avatars; links to external sources for health and health related data (see Deliverable D3.4); model repositories, information extraction from the web and data collection using mobile apps; semantic data harmonization to support the data/model searching and reasoning. MyHealthAvatar follows recommendations from relevant VPH activities on "Digital Patient". MyHealthAvatar architectural platform is designed as a multifunctional integrated facility. Its distinctive features include: *Data and model repositories to provide rich resources of data and models; ICT services to support data collection with minimal user input, including web information extraction, mobile apps, etc; ICT toolbox in support of clinical decision making by using multiscale models and visual analytics; Ontology and RDF repositories to support data integration, search and reasoning; an ICT architecture that allows for access to data from a range of different sources, and integration of the repositories, the toolbox and the ICT utilities; private and public cloud solutions to support the storage and computational requirements for the avatars without remote data transfer and specifications of open MyHealthAvatar APIs for external, third-party developers.*

3.6 MyHealthAvatar Stakeholders

A stakeholder is anyone who has an interest in or concerns about the system that we actually building. In MyHealthAvatar the most important stakeholders are:

- ✓ ***Patients/Citizens***. In MyHealthAvatar where personalized provision of health and patient empowerment services are to be delivered, patients represent a prominent type of stakeholders.
- ✓ ***Researchers/Medical personnel expert domain Users***. These can be further classified in bioinformaticians, clinicians, users of clinical trials management systems, etc.
- ✓ ***Software Engineers/Developers***. The people who actually build the system.
- ✓ **Administrators */Maintainers***. The people that evolve and sustain and guarantee the good operation and sustainability of the system.

During the project we focus on the domain users and the patients, and secondly on the developers and similar stakeholders. Focusing on the expert users/patients means that we elaborate on their concerns, which mostly have to do with the functionality, and some of its quality attributes such as security and usability. On the other hand, the developers' concerns relate to the development process, its phases (e.g. design, code, test), and various "satellite" issues like the choice of the programming environment, the development tools, etc. The project has carried out analysis of detailed end-user requirements, reported in D3.1, and needs by collecting an initial set of Scenarios / Use Cases (D2.2). These cases were collected by consortium members through interaction with all MyHealthAvatar's system stakeholders, including citizens/patients, clinical doctors and clinical and IT researchers. From the initial set of scenarios the consortium defined MHA high end clinical demos that were selected for further implementation and evaluation. These demonstration scenarios are close related to the prioritized and final set of Use Cases / Scenarios reported in D7.1 and D9.1. This user centered approach allowed us to us to select and describe in more details the Scenarios / Use Cases and related requirements / user need proposed for implementation and final demonstration for MyHealthAvatar. Each of these cases addresses a use scenario from a particular user perspective, either as a patient, or as a doctor, or as a clinical or IT researcher. For the evaluation we managed to attract many different volunteers that tested the platform functionality and all related reports are within deliverable D9.3.

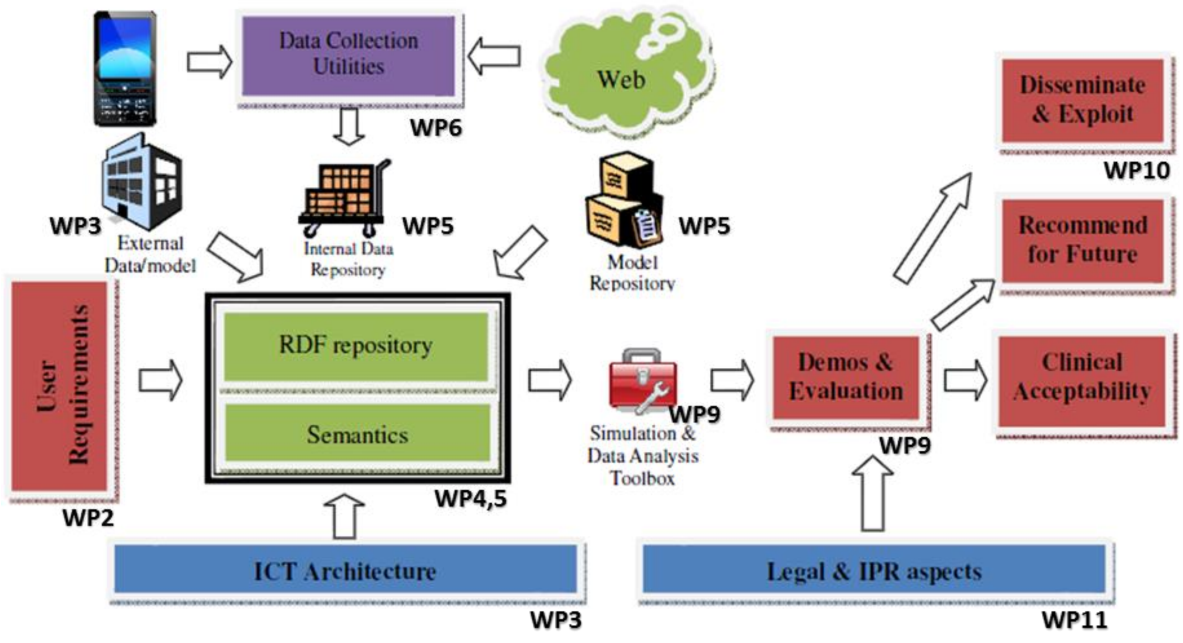


Figure 3-8: MyHealthAvatar platform as a unified system and its interactions with external entities

3.7 MyHealthAvatar Architectural design and views

A context diagram for the MHA system, when considered a single, unified system, can be seen in Figure 4-1. The main feature of this type of diagram is that it shows clearly the channels of communication with all sub-systems and external systems. It clearly presents the key components that must be built during our work plan. The key components of the related working task include: 1) user requirements 2) Internal data repositories and an internal model repository; 3) ICT architecture that support data access to internal and external resources and data management; 4) Data collection utilities; 5) Semantics and RDF repository to support data search and reasoning; 6) Simulation and data analysis toolbox 6) Demo & evaluation; 7) Investigation of the legal and IPR aspects of the avatars; 8)

Understanding of clinical acceptability; 9) Recommendations for the future work; 10) Dissemination and exploitation of the results to influence the future healthcare system

The requirement analysis of MyHealthAvatar clearly described the borders of the system architecture allowing the technical manager and the developing team to identify external entities that considered necessary for the system development : HIS, EHR and PHR systems, Drug data, Social Networks and other sources of online activity of the users, Model repositories that contains simulation models, PubMed Repository, Clinical Trials information, news articles, etc., Clinical processed data and data from external Warehouse (lab results, images etc.), Third party application (external to MyHealthAvatar) data, Diabetes and Emergency Demo, Personalized CHF Related Risk Profiles and "Real-Time Monitoring" (CHF), Osteoarthritis (OST), Nephroblastoma (Wilms Tumour) Simulation Model and Clinical Trial (UC-NEPH): In-silico Profiling of Patients and Predictions. All these entities were implemented and all related software components were integrated to MHA final integrated platform prototype. A logical view of the architecture based on the required functionality already defined in the project's description of work can be found in D3.2 version 2. In this document we describe the identified scenarios and the responsibilities of the components, their interactions, and we try to elaborate through MyHealthAvatar platform's computational nodes and heterogeneity of components.

MyHealthAvatar deployment view diagrams are shown below:

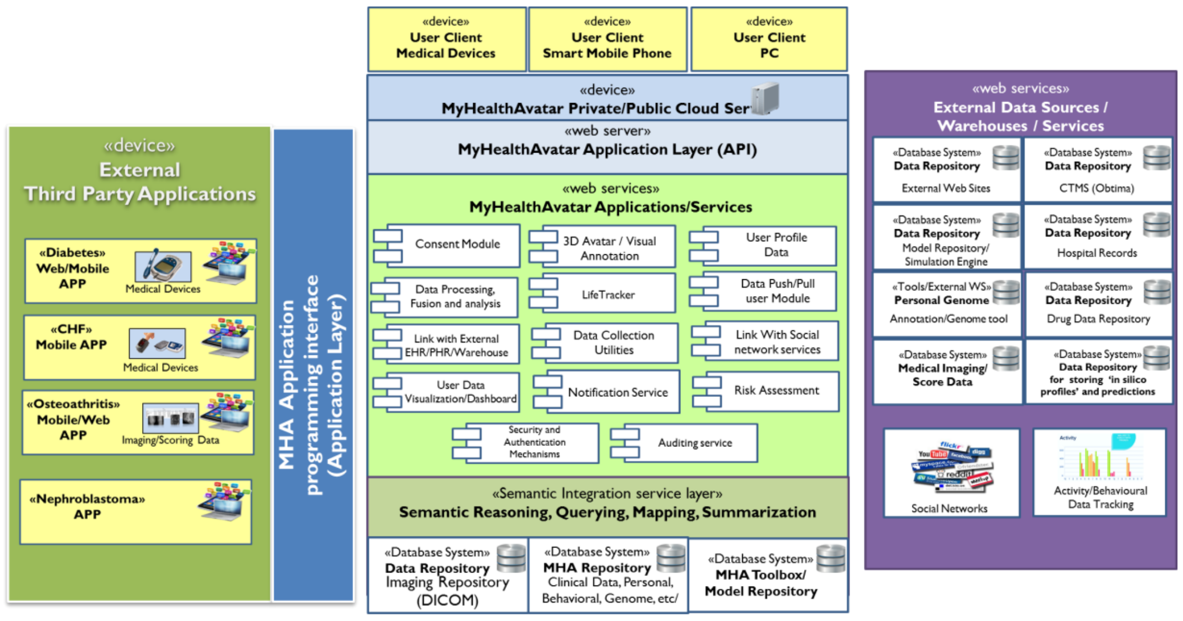

«device» User Client Medical Devices
«device» User Client Smart Mobile Phone
«device» User Client PC
«device» MyHealthAvatar Private/Public Cloud Ser
«web server» MyHealthAvatar Application Layer (API)
«web services» MyHealthAvatar Applications/Services
Consent Module
3D Avatar / Visual Annotation
User Profile Data
Data Processing, Fusion and analysis
LifeTracker
Data Push/Pull user Module
Link with External EHR/PHR/Warehouse
Data Collection Utilities
Link With Social network services
User Data Visualization/Dashboard
Notification Service
Risk Assessment
Security and Authentication Mechanisms
Auditing service
«Semantic Integration service layer» Semantic Reasoning, Querying, Mapping, Summarization
«Database System» Data Repository Imaging Repository (DICOM)
«Database System» MHA Repository Clinical Data, Personal, Behavioral, Genome, etc/
«Database System» MHA Toolbox/ Model Repository
«device» External Third Party Applications
«Diabetes» Web/Mobile APP
Medical Devices
«CHF» Mobile APP
Medical Devices
«Osteoathritis» Mobile/Web APP
Imaging/Scoring Data
«Nephroblastoma» APP
MHA Application programming interface (Application Layer)
«web services» External Data Sources / Warehouses / Services
«Database System» Data Repository
External Web Sites
«Database System» Data Repository
CTMS (Obtima)
«Database System» Data Repository
Model Repository/ Simulation Engine
«Database System» Data Repository
Hospital Records
«Tools/External WS» Personal Genome
Annotation/Genome tool
«Database System» Data Repository
Drug Data Repository
«Database System» Medical Imaging/ Score Data
«Database System» Data Repository for storing 'in silico profiles' and predictions
Social Networks
Activity/Behavioural Data Tracking

Figure 3-9: MHA final deployment diagram for the system and external third party applications for the demos

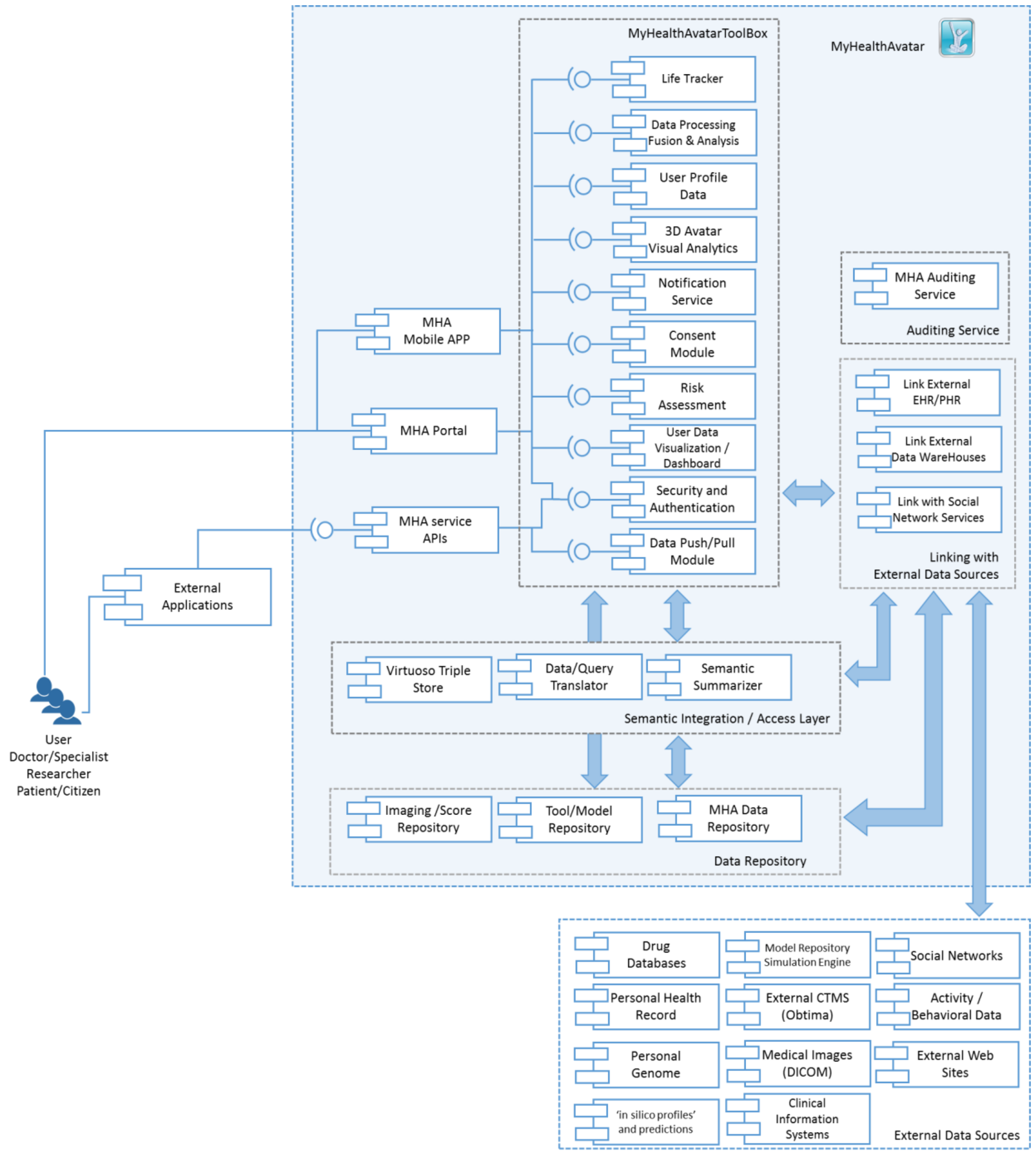


Figure 3-10: The main components of the system and their interactions

3.8 MyHealthAvatar demonstration Use cases

In this section we present the selected use cases for demonstration trying to identify the second phase of the various requirements refining the updated set of requirement that will be used to describe and design the architecture structure of MyHealthAvatar platform. A complete description of these use cases is reported in Deliverable D9.1. In MyHealthAvatar two general categories of scenarios are investigated: System use cases: these are the cases that describe the functionalities of the MyHealthAvatar system from the perspectives of both clinicians and citizens/patients and Clinical use cases: these are the cases that describe how to use the data from the MyHealthAvatar system in real clinical scenarios.

3.8.1 Diabetes and Emergency Demo

This use case is set to empower citizens by providing a supportive environment for the self-management of lifestyles for general health and wellbeing. Particular focus will be cast on risk analysis for diabetes, enabling more effective pre-diabetic care in terms of risk reduction through improving compliance with healthy lifestyle recommendation. The demonstration will allow the users to play a key role in monitoring and managing their own health. Allow multi-modal intervention of lifestyle in a shared decision manner between the doctor and citizens/patients. In the case of pre-diabetes, MyHealthAvatar will be able to demonstrate to the citizens/patients the relations between the outcomes of the self-management/treatment using prediction models. “Behaviour prescription” will be issued based on clinical guidelines and trusted sources (such as NICE), which is expected to include a set of targets in terms of daily activities, calorie intake and energy consumption, etc.

3.8.2 Personalized CHF Related Risk Profiles and "Real-Time Monitoring" (CHF)

Generally, cardiovascular disorders as chronic diseases require a continuous everyday record for patient’s status. Congestive heart failure (CHF) is a state in which the heart cannot provide sufficient cardiac output to satisfy the metabolic needs of the body. It is commonly termed congestive heart failure (CHF) since symptoms of increase venous pressure are often prominent. Its pathogenesis factors include: Age, Gender, Increased blood pressure, Smoking, Alcohol, Family and medical history, Genetic predisposition, Diabetes, Diet habits and Atherosclerosis. It’s a pathophysiologic state in which the heart, via an abnormality of cardiac function (detectable or not), fails to pump blood at a rate commensurate with the requirements of the metabolizing tissues or is able to do so only with an elevated diastolic filling pressure. Common causes include: coronary heart disease; hypertension; valvular heart disease and the general symptoms a patient/citizen can observe are: shortness of breath, leg swelling and exercise intolerance. The diagnosis is based on physical examinations and echocardiography. The management of the disease primarily involves intervention in order to improve the symptoms and the preventing disease progression mainly by altering the lifestyle (Diet, smoking, alcohol, moderate physical activity) and in many cases by pharmacological interventions.

The proposed scenario is built on the following pillars:

- **CHF Risk Assessment**: In order to tailor the proposed system to the patient’s profile and assist physicians in selecting people who are predisposed by coronary disease, hypertension, or valvular heart disease; we build a CHF related risk profile based on a risk appraisal function that is based on the diagnostic criteria [i.e. the Framingham Heart Study (486 heart failure cases during 38 years of follow-up)]. The predictors used are based on Age, Coronary heart disease and Valve disease status provided by the patient Electronic Health Record (EHR), as well as on HR, on blood pressure and on Body Mass Index (BMI) provided by the pulse oximeter, the blood pressure monitor and the weight scale, respectively. The calculated risk probability may be used to alter the default threshold values (higher risk probability adds more constraint on the physiological patterns). Furthermore, we present what else data regarding patients' health status could be embed into the platform towards the creation of a profile with necessary information for both patient and treating physicians. To this respect an approach of presenting data regarding demographic, physiology, diagnostic test results and disease management (i.e. prescribed drugs) is provided.

- **Real-time patient monitoring**: In addition to the above the dedicated clinical personnel should be contacted immediately and possibly intervene in time before an acute state is reached, by changing medication, or any other interventions, in order to ensure patient safety. There is a need to support real-time remote monitoring of patients diagnosed with congestive heart failure and MHA, enhanced with semantic technologies, may host personalized, accurate and up-to-date clinical information. To this end we built a real-time patient/ doctor alarming will be built according to rule-based alarms enabling intelligent alerting of the dedicated physician in case of an emergency. The alarming process will be based on vital signs monitoring and specifically Heart Rate (HR), Pulse Oximetry, and Blood Pressure acquisition, adapted according each specific patient's medical history and age, and even risk predictor's outcome.

3.8.3 Osteoarthritis (OST)

MHA offers a one-stop service for citizens for data collection and self-management such as monitor, record and education. Precisely, the system will support the storage of behaviours and daily activities of citizen. It will function as a supportive environment to empower normal citizens in looking after their own health, raising their self-awareness of any potential risk of developing diseases while encouraging their healthy lifestyles in terms of doing routine daily exercises, stopping smoking and controlling their diet. Therefore, naturally many existing functionalities in MHA can be directly used for the needs of osteoarthritis use case. In addition, we will incorporate genetic predisposition evaluation services for examining if an increased risk of developing osteoarthritis exists, which will be used by the citizens in order to understand their personal risk of developing osteoarthritis, and the impact of their behaviour and lifestyles towards the risk.

3.8.4 Nephroblastoma (Wilms Tumour) Simulation Model and Clinical Trial (UC-NEPH): In-silico Profiling of Patients and Predictions

The outcome of this high-end scenario is to provide a tool which produces the 'in-silico profiling' of nephroblastoma patients and performs 'in-silico' predictions of therapeutic schemes outcome.This can be used in a fourfold way:

1. To demonstrate to patients / or parents of patients how a given tumour will respond to preoperative chemotherapy. This will help in explaining diagnosis and treatment of nephroblastoma to patients and/or parents of patients. Such a demo will not use the actual data of the given patient.
2. To give physicians treating a patient with a nephroblastoma the ability to check how this specific nephroblastoma will respond to preoperative treatment with vincristine and actinomycin-D.
3. To provide clinical researchers and modelers a powerful tool to define an in silico patient profile and further exploit it in other modelling approaches and VPH projects. Moreover, it could serve as a statistical tool to categorize patients (by associating their clinical and *in silico*

profiles) and define ranges of model parameter values to guide the process of model adaptation for new patient cases.

4. To demonstrate to citizen what ‘in silico’ models/tools can do today. This can serve as a learning environment for ‘in silico’ models and will help to disseminate the importance of ‘in silico’ models in medicine to the public, to medical stakeholders, industry and funding agencies. It is pointed out that the purpose of the in silico experimentation functionality is currently limited to the education of the public so that they can be prepared for the future translation of thoroughly clinically validated models to clinical practice.

3.9 MyHealthAvatar Integrated Platform

A simplified view of the integrated platform can be seen below.

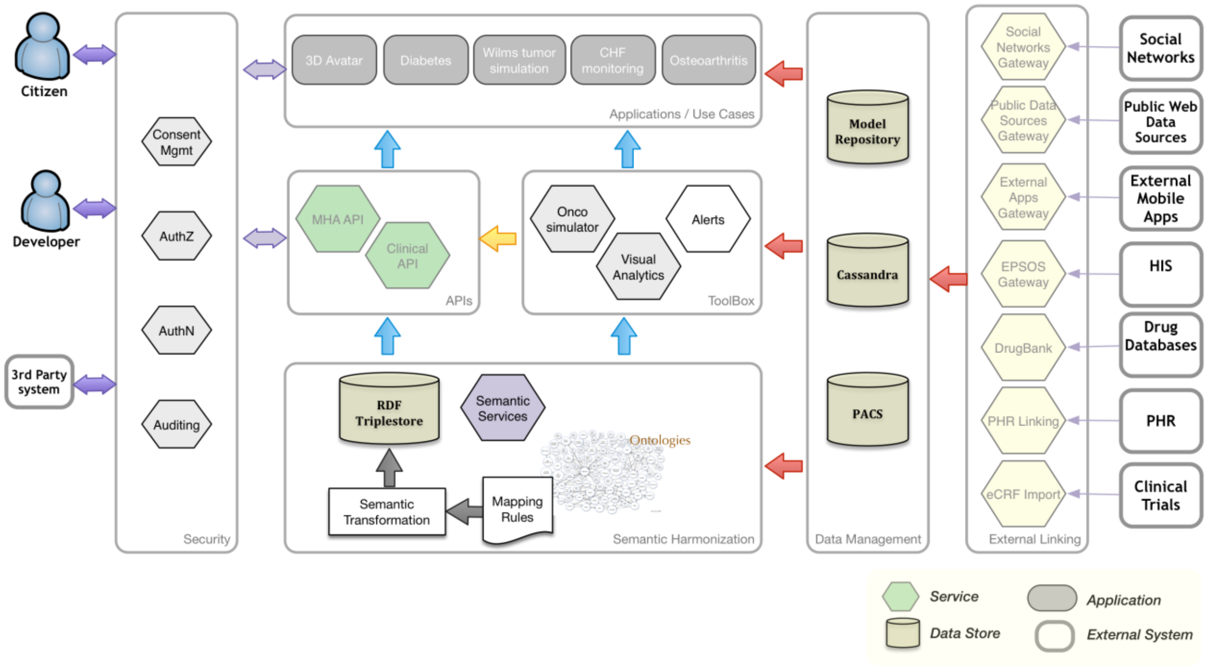


Figure 3-11: The integrated platform of MHA, as a composition of multiple domains of functionality.

The system as a whole consists of a number of services, data storage components, adapters/gateways, and applications performing specific functionalities of the system that we have grouped and categorized as follows:

- **Data Management**. This is the layer of the architecture addressing the data persistence requirements of the system. The challenges for high availability, performance, and scalability led to the adoption of the Apache Cassandra in order to ingest and store large-scale data. Cassandra is an open-source, peer-to-peer and key value based store, where data are stored in key spaces and it’s considered state of the art for real-time (big) data analysis with advanced replication functions. In MHA platform the Cassandra repository is an instantiation of a “data lake" concept[8] that stores the raw data supplied by the different information sources. Next to Cassandra, there are additional specialized data stores like the DICOM Repository (Picture Archiving and Communicating System, PACS[9]) and the Model Repository. The requirement for increased availability and scalability in this architectural layer does not really affects the design of these additional data storage components, for the following reasons: For the model repository, its contents do not change frequently and additionally the limited number of models do not necessitate extensive query and retrieve capabilities and computational power. The situation is slightly different for the DICOM repository of medical images, where depending on the use of the DICOM import functionality the space requirements for storing these images can be really big. Nevertheless, the DICOM protocol that follows a hierarchical data model, starting from the patient identifier, and the use of Cassandra as the primary data repository in this layer allow us to horizontally scale even for this case. Currently, there's no such a need but if in the future the size of the image repository and the load of image retrieval requests exceeds the capacity of a single deployment we can adopt a parallel DICOM storage architecture in the cloud[10].

- **External Linking**. The components laid in this layer are the "ports" and "adapters" for the external data sources, which include hospital information systems and PHRs, social networks, clinical trial management systems, etc. Specific gateways have been implemented for each of these types of external sources that perform an "Extract Transform Load" (ETL)[11] process in order to feed the Data Persistence layer with the incoming data. In this process, the gateways make use of available standard interfaces whenever is possible, such as the Clinical Document Architecture (CDA[12]) guidelines and set of specifications for the retrieval of clinical data from Hospital Information Systems (HIS), the Operational Data Model (ODM) of the Clinical Data Interchange Standards Consortium (CDISC[13]) for clinical trial specific patient, whereas cross-border healthcare provisioning is supported by the adoption of epSOS[14] Patient Summary interfaces. The main "sink" of the acquired data is the Cassandra repository so that is then made available to the rest of the platform.
- **Semantic harmonization**. This is the semantic integration layer which is responsible for summarizing, semantical integrating, and consolidating the information that is stored in the Data Management layer. There are two main reasons for the introduction of this layer in the architecture. First, in order to deal with the semantic and terminological heterogeneities of

[8] Martin Fowler, Data Lake, http://martinfowler.com/bliki/DataLake.html

[9] Choplin, R. H., Boehme 2nd, J. M., & Maynard, C. D. (1992). Picture archiving and communication systems: an overview. *Radiographics*, *12*(1), 127-129.

[10] Yuan, Y., Yan, L., Wang, Y., Hu, G., & Chen, M. (2015). Sharing of Larger Medical DICOM Imaging Data-Sets in Cloud Computing. *Journal of Medical Imaging and Health Informatics*, *5*(7), 1390-1394.

[11] Vassiliadis, P. (2009). A survey of Extract–transform–Load technology. International Journal of Data Warehousing and Mining (IJDWM), 5(3), 1-27.

[12] http://www.hl7.org/Special/committees/structure/index.cfm

[13] http://www.cdisc.org/

[14] European Patients Smart Open Services (epSOS) project, http://www.epsos.eu/

the incoming data from the External Linking tier and to provide a unified, common, and ontologies based schema of the available information. Secondly, to provide more domain specific and expressive way to perform queries on the managed information. Cassandra is a typical example of a NoSQL repository[15] and this type of technologies have increased rapidly in the recent years due to their ability to handle enormous data sets and the "schema-less" nature. But the limitations of NoSQL databases in the flexibility of the query mechanisms are a real barrier for any application that has not predetermined access use cases. The RDF Triplestore in the MHA platform fills these gaps by effectively providing a semantically enriched and search optimized index to the unstructured contents of the Cassandra repository. Of course in order to alleviate any scalability issues in the Triplestore when answering complex queries, its contents are actually a "digest" of the information stored in Cassandra. The "recipe" for building the semantic content in the Triplestore is provided by the mapping rules and the use of relevant ontologies in the Semantic Transformation process performed by specific Semantic Services, as described in Deliverable D4.3. In conclusion the Semantic Harmonization complements the Data Management layer and this design offers the best of both worlds: efficient persistence and availability of heterogeneous data, and semantic integration and searching of the "essence" of the ingested information.

- The so called MHA **Toolbox** consists of services and components to facilitate clinical data analysis and knowledge discovery. It contains a set of integrative models and data analysis tools to support clinical decisions by allowing the clinicians to take into account multiple aspects that are influential to prognosis, diagnosis, and treatment selection. Examples of such models and components are the Wilms Tumor Oncosimulator, the visual analytics infrastructure, and the generation of important alerts for the patient's wellbeing and health. The functional components in this layer use both the semantic infrastructure services and the "raw" data storage components. For example, the implementation of the alerting functionality requires strong support for semantic harmonization and clinical decision support rules expressed in the clinical ontologies and relevant terminologies. On the other hand, the in silico simulation services require information from the model repository and the DICOM medical images. Finally, the visual analytics and knowledge discovery components usually necessitate the employment of "Map Reduce"[16] tasks in the Cassandra repository to explore the whole data space.

- **Application Programming Interfaces (APIs)**. In order to actively engage the public and 3rd party developers and systems, programming interfaces have been defined for sharing the data, results, analytics, etc. gathered and managed in MHA. The APIs defined are made available to anyone interested, and for example the Clinical API for retrieving the clinical data of the MHA users is published as open source and its implementation can be found at https://github.com/sgsfak/mha-clinica-api MHA APIs are implemented as "web frontend/gateway" RESTful services that aggregate the information located in various places in the platform, from the Data Management and Semantic Harmonization layers to the ICT Toolbox components.

[15] Han, J., Haihong, E., Le, G., & Du, J. (2011, October). Survey on NoSQL database. In Pervasive computing and applications (ICPCA), 2011 6th international conference on (pp. 363-366). IEEE.

[16] Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, *51*(1), 107-113.

- **Security**. Protecting the privacy of the data against unauthorized access and modification is of course of paramount importance. The security layer includes the services that address these requirements and provide support for authentication and single-sign-on, authorization, auditing, identity management, etc. This layer is therefore intercepting every access to the services and interfaces of the platform. Furthermore, more domain specific requirements like the management of patient consent (for importing and using her/his clinical data, for example) are supported by specific infrastructure in this layer.
- **Applications and Use Cases**. As the upper layer and the user interfacing view of the system, the Applications tier includes the value-added, specialized, end-users applications and scenarios. Most of these applications are listed within above. These applications use mainly the programming interfaces and the services offered by the Toolbox since they effectively hide and abstract away much of the underlying data retrieval and management.

The above description of the MHA architectural elements and their interactions demonstrate the level of integration achieved. This is of course a "20,000 feet overview" of the integrated platform that shows the flow and transformations of the data and the information extracted from them by the interacting MHA components. In section 5 we take a close look into specific software elements and processes of the platform and we provide details on their integration to the whole system.

3.10 MyHealthAvatar Integration/deployment environment

This section shows the delivery models supported by MHA within the deployed cloud infrastructure. Their scope is to provide resources, application platforms, common infrastructure and software as services to consumer[17]. These service models also place different levels of security requirements upon the deployed environment. As capabilities are inherited by successive models, so too are information security issues and risks. The consideration of cloud technology was vital to ensure the long term scalability and performance of MyHealthAvatar in data management and service management architecture. The main approach is to work on a locally-deployed cloud infrastructure which can utilize local computing power as well as maintain the ability to outsource the infrastructure to commercial cloud computing facilities (e.g. Amazon EC2). This way we are able to provide and sustain stable and production ready resources to support MHA needs regarding data preservation. The swift services are highly autonomous so the whole architecture is flexible enough to allow different deployment scenarios. The four main services are: Proxy Services, Object Services, Container Services and Account Services. Proxy Service plays role of the contact point (API) with users and 3rd party services, while other three kind of services manage files, containers and accounts so are used to manage physical data and logical structure. The major difference between locally deployed cloud (also known as private cloud) and public cloud is the control and access of the resource. In private cloud, resources are controlled and accessed by the premise only. On the other hand, in public cloud, resources are controlled by the cloud provider but are accessible for public users. Therefore, a hybrid cloud infrastructure is adopted in MyHealthAvatar with both public and private cloud facilities available. In

[17] Shey, H., R. Wang, J.P. Garbini and E. Daley, 2009. The State of Enterprise Software: 2009. Forrester Research, Inc.

deliverable D3.2 2nd version we have included detailed information about the private and public cloud deployment of MHA including all technical information of the supporting cloud software and services.

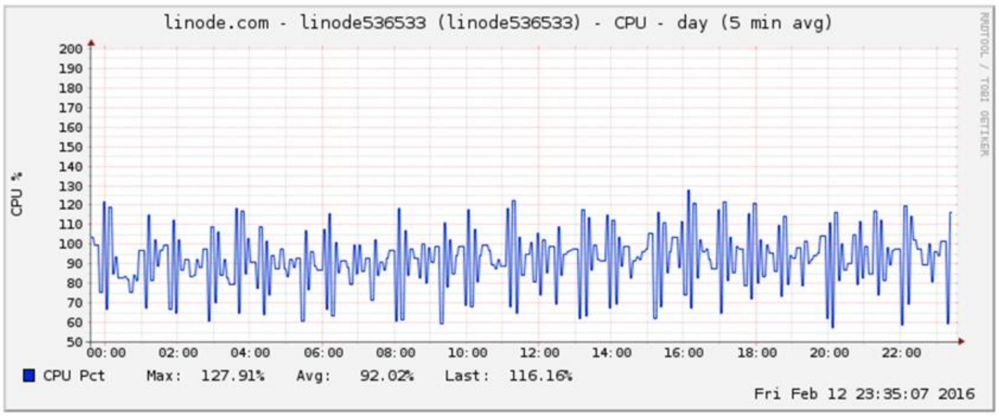
Physical Deployment

MHA platform is deployed in the premises of FORTH, in the form of a private computational and storage cloud.

Figure 3-12: Physical installation

In terms of hardware resources, the cloud infrastructure allows for maximum elasticity and flexibility by effectively adapting to the load of any given time. The current minimal specifications include: 300 GB of RAM, 9TB of storage and 16 cores Intel® Xeon® Processor E5-2690 and 4 cores Intel® Xeon® Processor E7520 (Dell PowerEdge R720 and SC 1425 Servers series). In terms of the software the OpenStack[15] open source cloud computing software has been installed on the machines using the Linux Ubuntu 12.04 operating system[16]. **Fort MHA we can have one or more VMs for 4GB RAM and 100GB for storage (or any other different setting for each VM), according to MHA deployment demands. We are able to serve MHA need as the project progress.**

MHA is also installed and deployed physically on a public cloud namely *Linode* which a cloud hosting service, the 8GB plan is chosen to deploy on Linode's London based servers. The virtual machine has 6 CPU cores, 192GB SSD storage, 8GB RAM. Linode provides a web based panel to monitor the virtual machine's resource usage.



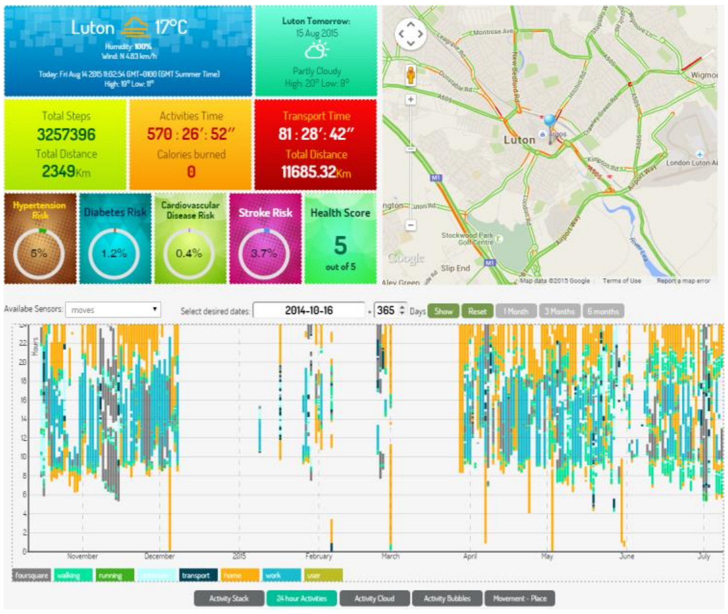
MyHealthAvatar utilize the power of both Forth cloud and Linode cloud services to form a hybrid cloud which gives the platform the benefit of both world.

4 MyHealthAvatar Technical integration and evaluation

4.1 *MyHealthAvatar Portal*

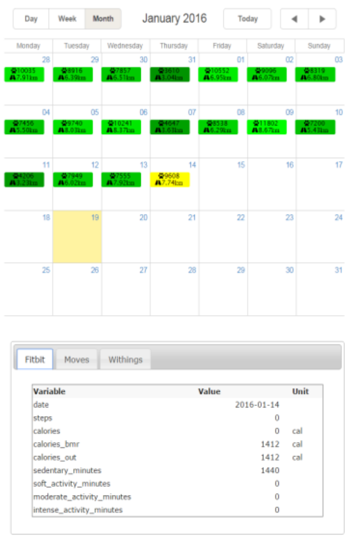
There are many components and data that can be accessed by the user from the web-based MyHealthAvatar portal. However, as there are a variety of data sources and data types, it is very difficult for a user to grasp an overview with important notifications from the scattered health status information. Various components and visualisation are utilized to present the information to users in a meaningful an intuitive manner.

4.1.1 Dashboard



To present the user a quick overview of their health status, MHA provides a dashboard on the front page. The dashboard provides a summary of the users' latest health status and present important notifications. It may include several simple visualization components to depict data for a relatively recent period. Following figure shows an example dashboard with data tiles, map and a timeline. The user can interact with the map and the timeline to obtain more detailed information.

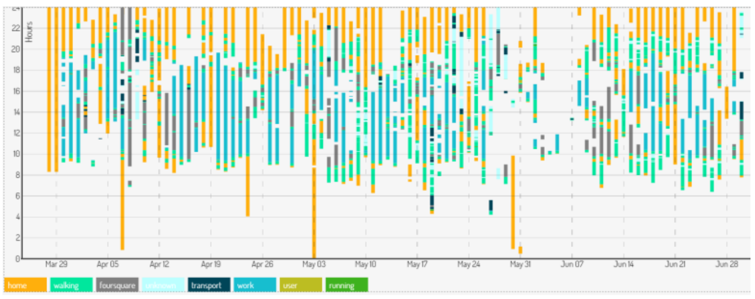
4.1.2 Diary



Variable	Value	Unit
date	2016-01-14	
steps	0	
calories	0	cal
calories_bmr	1412	cal
calories_out	1412	cal
sedentary_minutes	1440	
soft_activity_minutes	0	
moderate_activity_minutes	0	
intense_activity_minutes	0	

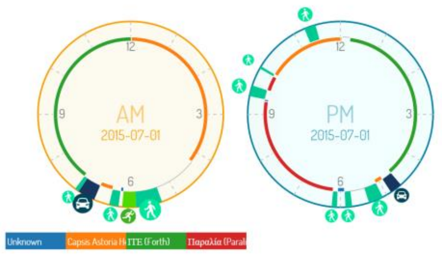
MyHealthAvatar provides health data collection, storage and access to end-users. The data could be either automatically collected or manually input by users. For lifestyle and health tracking, the data are often time-dependent, especially date-dependent. A natural form of date-based data organisation, display and editing is a calendar, which is a traditional way to visualise daily events. In MyHealthAvatar, a calendar- based diary is used for daily data display as well as daily event input, editing and planning. Following figure shows an example view of the calendar with the event editor. The calendar displays a brief summary of the fitness data such as daily steps, walking and transportation distance, as well as calories burned. With the event editor the user can add events, providing the start and end time, location, and detailed descriptions. The user can add tags for events to facilitate event categorization. The user can plan by adding events. The events and planning will be shown in the calendar.

4.1.3 Timeline



A timeline is a traditional method to visualise time-varying data and events in a linear layout. Compared to a calendar, a timeline is more suitable for visualising continuous variables, which cover a relatively long period, such as health indicators and medical measurements. Time dependent activity events can also be shown in a timeline if a longer time scale is desired to view daily activity events and activities. In the current implementation, the timeline supports interactive visualisation of Fitbit/Withings sensor data as well as Moves data. There are five different visualisation styles including activity stack, 24-hour activity, activity cloud, activity bubbles and movement-place. Activity stack shows activities directly on the timeline in a form similar to stack bar charts. A 24-hour activity organises the activities on a daily basis for easier comparison of daily activity changes. The activity cloud uses concentric disks of different radius to represent the activities; activity bubbles use bubbles of different colour and radius. Movement-place shows the movement and place in the users Moves data.

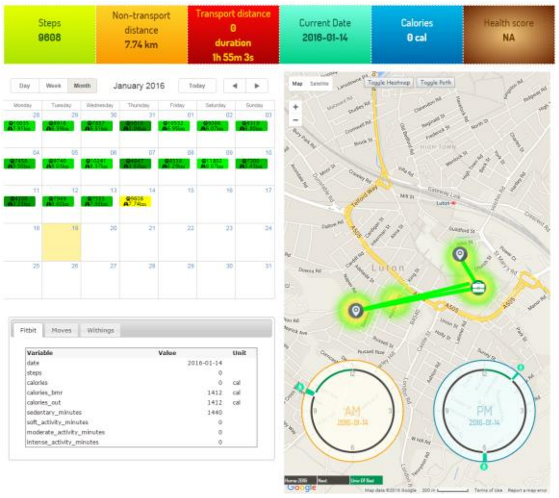
4.1.4 Clock View



For daily activities, timeline provides visualisation over a relatively long period. Interactive timelines can provide zooming to smaller scales. However, the linear layout may make it di cult for the user to understand and compare daily events. A ne-grained view of activities within one day is better visualised in a radial layout. A natural, real-life way of radial daily time representation is the clock. MyHealthAvatar uses a similar radial layout called ClockView to visualise daily events. Movements and places from Moves data are visualised in the radial layout. Activity types are marked by icons and colours. When the user hovers the mouse over the icons more detailed information will be displayed, as shown in following figure.

4.1.5 Map

While dairy, timeline and clock view are largely designed for visual analysis of temporal data, they can hardly be used to visualise spatial locations. A map is a natural choice to provide intuitive spatio-temporal visualisation and analysis of the user's locations and routes for better understanding and knowledge discovery of the lifestyle. The map implementation is based on Google Maps [6]. Currently, in MyHealthAvatar the map is used for visualisation and analysis of the Moves data only but it is capable of supporting other location-sensor-based apps. In addition, MyHealthAvatar uses an integrated view which is called Life-Tracker to visualise and analyse events and activities, including diary, map and clock view, as shown in Figure 5. The advantage of this compound view is that it provides integrated spatio-temporal visualisation and analysis. The page itself provides the user an extensive view of data collected from di erent sources and the user does not need to refer to multiple pages to view and analyse related spatio-temporal data collected and stored on the MyHealthAvatar platform.

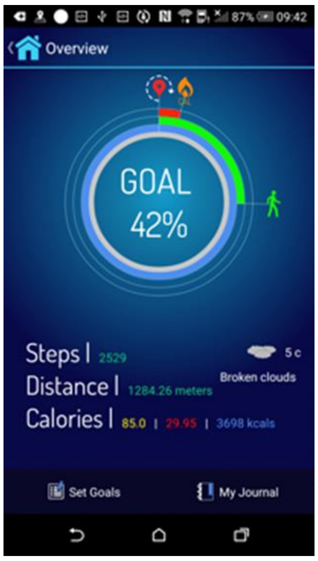


Variable	Value	Unit
date	2016-01-14	
steps	0	
calories	0	cal
calories_bmr	1412	cal
calories_out	1412	cal
sedentary_minutes	1440	
soft_activity_minutes	0	
moderate_activity_minutes	0	
intense_activity_minutes	0	

4.2 MyHealthAvatar Mobile Application

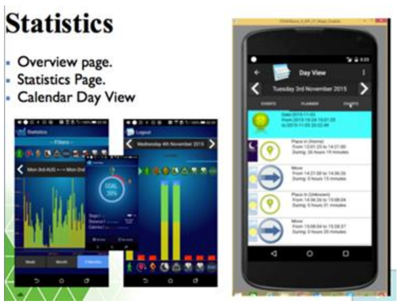
4.2.1 Overview on the mobile application

This overview utility, developed in WP6 and WP8, shows your today's activity in a simple and understandable way. Users can set their own goals (e.g. steps, walking distance, and active minutes) in consultant with their careers. (see figure on the right



4.2.2 Statistics view on the mobile application

Statistics show you a time range of activity data with goal achievement indicators.

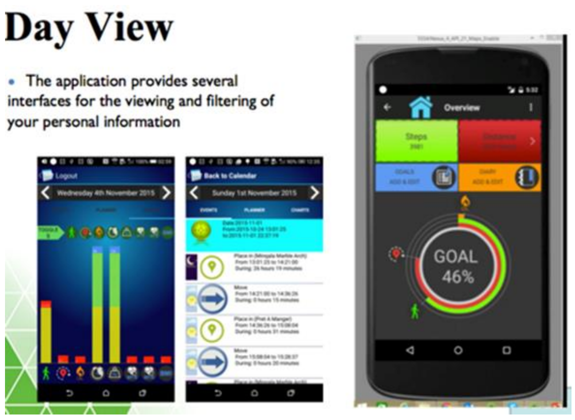


Statistics

- Overview page.
- Statistics Page.
- Calendar Day View

4.2.3 Day views on the mobile app

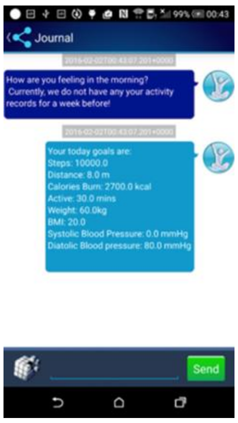
The event tab shows you a chronological break down of a specific day (WP6, WP8). The chart tab allows you to see all the day's activity data in a simplified chart.



Day View

- The application provides several interfaces for the viewing and filtering of your personal information

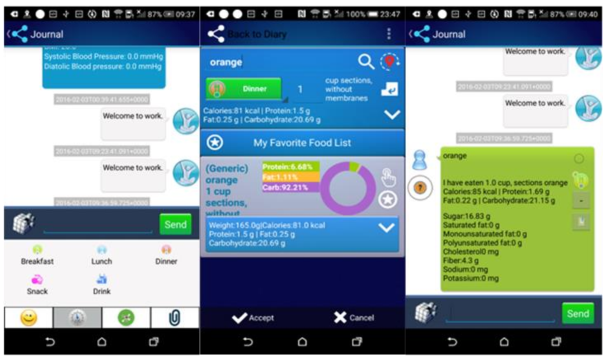
4.2.4 Chat interface


Journal

How are you feeling in the morning?
Currently, we do not have any your activity records for a week before!

Your today goals are:
Steps: 10000.0
Distance: 8.0 m
Calories Burn: 2700.0 kcal
Active: 30.0 mins
Weight: 60.0kg
BMI: 20.0
Systolic Blood Pressure: 0.0 mmHg
Diatolic Blood pressure: 80.0 mmHg

A chat interface is to log food consumption, mood and other health-related information using text, photo, icon and voice through a chat with MyHealthAvatar. Through the chat interface, the MyHealthAvatar app can act as an interactive Robot (&your buddy) to ask for your information at appropriate time and location, for example, while you are waiting for bus. This will hopefully help and encourage users to provide more of their information to the app. The personalized cancer care information may come from UK NHS websites, such as http://www.nhs.uk/pages/home.aspx - if there is any other sites that are better and more cancer specific, please advice. From these sites, the technology can extract information and deliver to the patients according to their profiles. Throughout our discussions we feel that personalization is the key here because we do not want patients to receive excessive amount of information about all types of cancer. The app also informs the patients about their targets (e.g. daily targets in the morning) and their performance (e.g. compared with goals) and the performance summary on a daily, weekly or monthly basis. It also delivers you tips and reminders for healthy behaviors, for example, breakfast recommendations, suggestions for a walk if you have been sitting for too long.


Journal

Systolic Blood Pressure: 0.0 mmHg
Diatolic Blood pressure: 0.0 mmHg

Welcome to work.

Welcome to work.

Back to Diary

orange

Dinner 1 cup sections, without membranes

Calories:81 kcal | Protein:1.5 g
Fat:0.25 g | Carbohydrate:20.69 g

My Favorite Food List

(Generic) orange 1 cup sections, without

Protein:6.68%
Fat:1.11%
Carb:92.21%

Weight:165.0g|Calories:81.0 kcal
Protein:1.5 g | Fat:0.25 g
Carbohydrate:20.69 g

Journal

Welcome to work.

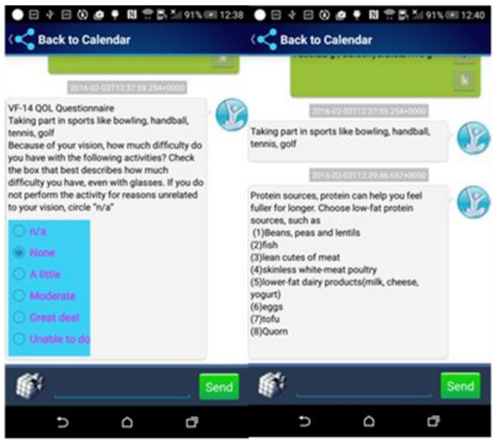
Welcome to work.

orange

I have eaten 1.0 cup, sections orange
Calories:85 kcal | Protein:1.69 g
Fat:0.22 g | Carbohydrate:21.15 g

Sugar:16.83 g
Saturated fat:0 g
Monounsaturated fat:0 g
Polyunsaturated fat:0 g
Cholesterol0 mg
Fiber:4.3 g
Sodium:0 mg
Potassium:0 mg

Morning greeting from MyHealthAvatar, who asks about your feelings and tell your daily goals. It also reminds you about what you did in the same day last week/month/year. Location based message (left), The interface to enter your diet information (center), The entered information visible on the chat interface (right).


Back to Calendar

VF-14 QOL Questionnaire
Taking part in sports like bowling, handball, tennis, golf
Because of your vision, how much difficulty do you have with the following activities? Check the box that best describes how much difficulty you have, even with glasses. If you do not perform the activity for reasons unrelated to your vision, circle "n/a"

- n/a
- None
- A little
- Moderate
- Great deal
- Unable to do

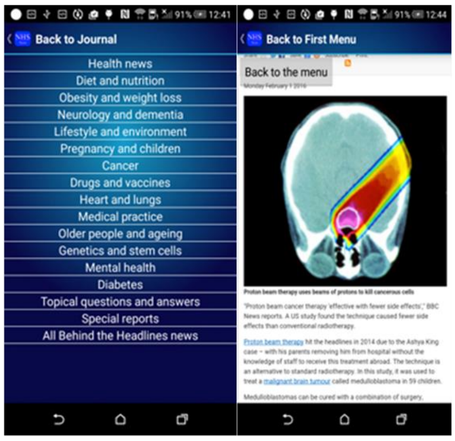
Back to Calendar

Taking part in sports like bowling, handball, tennis, golf

Protein sources, protein can help you feel fuller for longer. Choose low-fat protein sources, such as
(1)Beans, peas and lentils
(2)fish
(3)lean cutes of meat
(4)skinless white-meat poultry
(5)lower-fat dairy products(milk, cheese, yogurt)
(6)eggs
(7)tofu
(8)Quorn

You can answer questionnaires via the chat interface (left picture – right side), You can receive health information from the app (right picture – right side).

You can also receive published news from NHS (below)



Monday February 1 2016

Proton beam therapy uses beams of protons to kill cancerous cells

"Proton beam cancer therapy 'effective with fewer side effects'," BBC News reports. A US study found the technique caused fewer side effects than conventional radiotherapy.

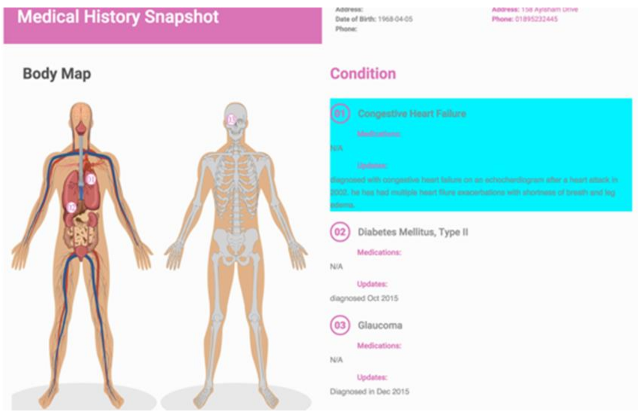
Proton beam therapy hit the headlines in 2014 due to the Ashya King case – with his parents removing him from hospital without the knowledge of staff to receive this treatment abroad. The technique is an alternative to standard radiotherapy. In this study, it was used to treat a malignant brain tumour called medulloblastoma in 59 children.

Medulloblastomas can be cured with a combination of surgery,

4.2.5 Profile summary and visualization

Once logged in, you may update and simultaneously view your general and health profile by using the My Profile menu as shown below (developed in WP6 and WP8). These include:

- Demographics
- Immunization
- Allergies
- Medical history
- Medication summary and schedule
- Lab tests results



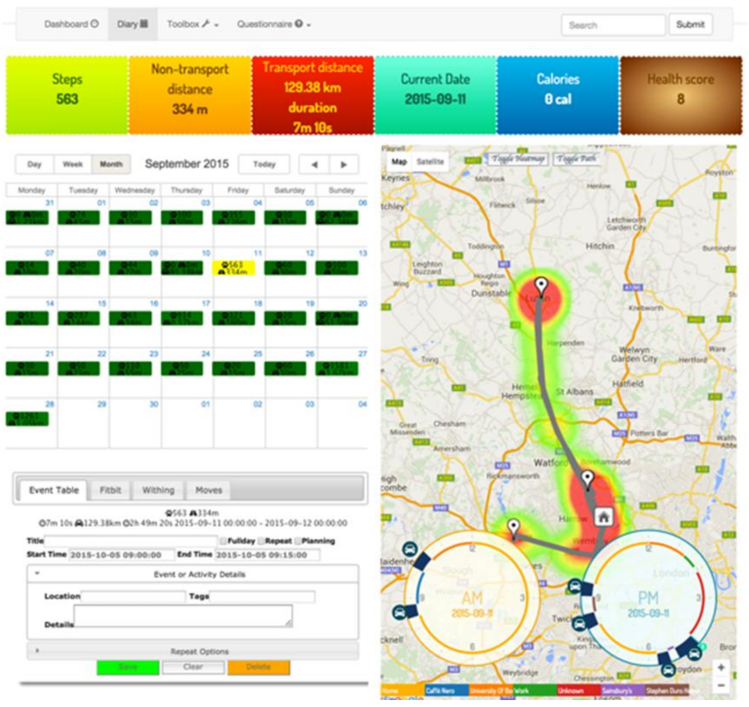
To edit your profile, click on the button with your account name on the top right of the page. Then from the drop down menu, select My Profile.

4.3 MyHealthAvatar Toolbox / Data collection utilities

4.3.1 User Profile Data

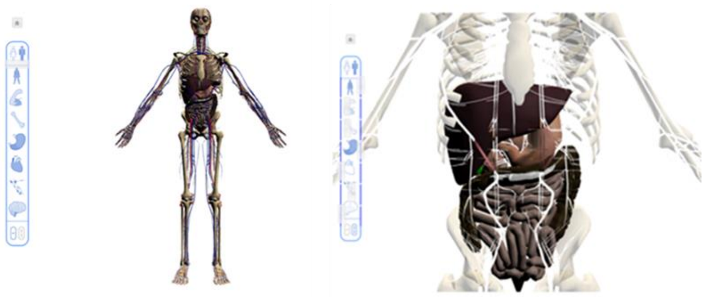
4.3.1.1 Diary

The Diary, a web application developed in WP8, shows your movement on a daily basis. The calendar also shows the information of your daily step counts, travel distance, etc. By selecting one day from the calendar (which is then highlighted in yellow), the colour tiles on the top of the page display your information on the day, including step count, non-transport travel distance, transport travel distance, and the map on the right of the page displays your movement and activity of the selected day. Heatmap and clock-views are used to illustrate your movements and activities on the map.



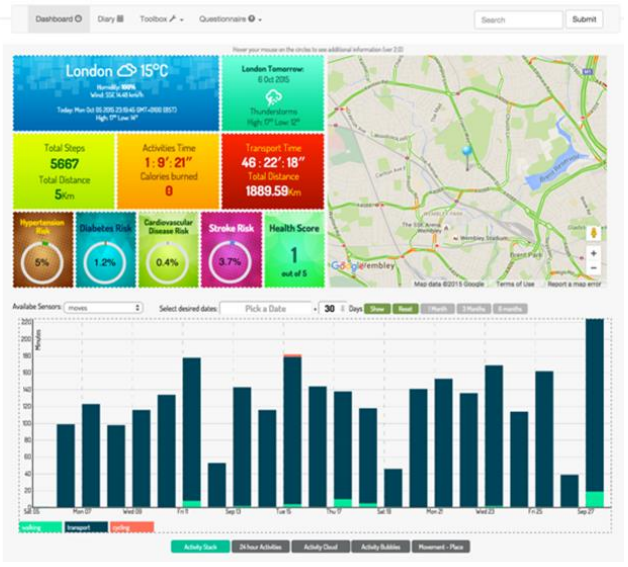
4.3.1.2 3D avatar visual Analytics

The web-based 3d avatar rendering suite shows the web-based 3d avatar (developed in WP8). This shows the structure of human anatomy and it is used for the purpose of patient education.



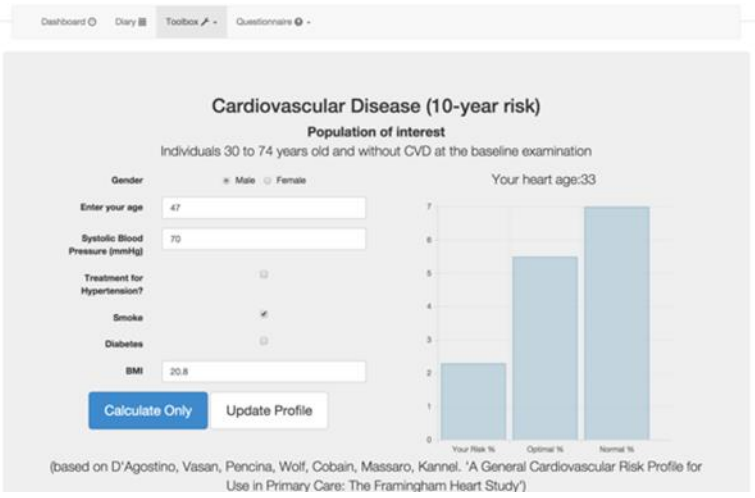
4.3.1.3 User Data Visualization and Dashboard

The Dashboard is web application, developed in WP8, showing your current location, the weather of your location and your overall activities – see below. The interactive timeline on the Dashboard allows you to view your activity data within a selected time period.



4.3.2 Risk assessment in toolboxes

The toolbox includes 4 established risk assessment models from the Framingham study to predict your risk of having cardiovascular disease, hypertension, diabetes and stroke according your profile, developed in WP6 and WP8.



Cardiovascular Disease (10-year risk)

Population of interest

Individuals 30 to 74 years old and without CVD at the baseline examination

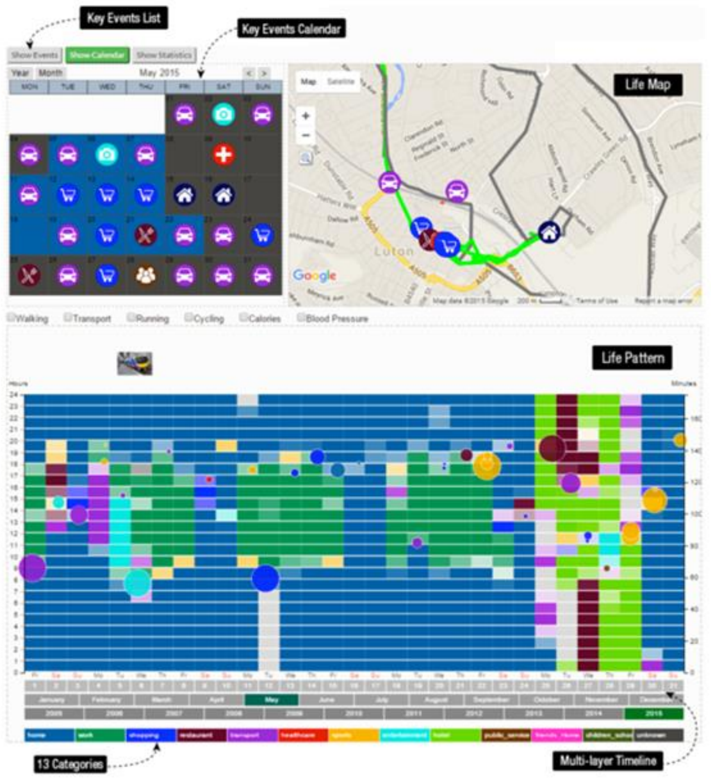
Gender	[● Male] ○ Female
Enter your age	47
Systolic Blood Pressure (mmHg)	70
Treatment for Hypertension?	☐
Smoke	✓
Diabetes	☐
BMI	20.8

Your heart age:33

(based on D'Agostino, Vasan, Pencina, Wolf, Cobain, Massaro, Kannel. 'A General Cardiovascular Risk Profile for Use in Primary Care: The Framingham Heart Study')

4.3.3 LifeTracker

LifeTracker, a web app developed in WP8, is a suite of techniques that presents lifestyle data and allows users to investigate it interactively. It employs a range of visual analytics techniques to make the outcomes of data summarisation and ranking available to the users, hence allowing them to identify the highlighted key events from the data. This supports the users to explore the data at different levels of detail and at different time scales. LifeTracker follows the principle of "overview first, zoom and filter, then details on demand" and offers an integrated environment to present information about individuals' daily activities in one place.



4.3.4 Security, Authentication and Consent module

In the exploitation stage, cConsent will not be sought by using a hardcopy written consent form, but by an online registration because MyHealthAvatar can reach more interested people by using e-consent forms. The drafted General Terms and Conditions explain the usual individual who wishes to join the MyHealthAvatar platform will be at home rather than in a clinical environment. The individual will not be able to enter the platform before he has granted consent. The consent form will explain the purpose of the platform (section 2, including the explanation of and what data the user can upload to the platform for using the different functionalities of MHA (section 4). and how the data will be used. Furthermore, it will be explained how the user's data will be protected and, following that, the rights of the user pursuant to Article 10 of the Data Protection Directive will be explained and described. To ensure that the user has understood all provided information a qualified member of the platform administration team can be contacted to ask questions. Currently, this is the project coordinator Professor Feng Dong. For the exploitation stage after the project's end, a contact point with qualified platform team members is desirable and a well-trained contact person will answer questions. Since consent has to be given explicitly according to Article 8 (2) lit. a of the Data Protection Directive, the consent form will not include a pre-ticked box. Instead, the user will have to take some positive action to signify consent. Deliverable D3.3 Security measures and guidelines reports the work done for building and maintaining the structure of the architecture platform by investigating and reporting security issues and measures for infrastructure, resource management, data access and federation, computing resource (possible links with external HPC). It deals with all the security aspects of the technological platform, ranging from user authentication, authorization, and auditing, to data integrity and privacy, to pseudo anonymization and re identification of patient data. The security tools and policies that are developed ensure and enforce the legal and regulatory compliance and encompasses the appropriate auditing mechanisms that are needed by the legislation. This deliverable includes guidelines on how to pose these security mechanism in MHA platform and examples on how to use these guidelines for the MHA use cases. The objective of a secure system is to protect sensitive information from unauthorized access, manipulation, misuse, etc. In MyHealthAvatar the information to protect are, patients citizen personal data stored in MHA repositories, medical measurements, private patient data, medical history, activity data, medical images, model repositories data etc. The protection goals which define the requirements of a secure system are defined by the following objectives[18, 19]. In Deliverable D3.3 we present in detail the security framework of MyHeathAvatar, its principles, privacy and legal framework, trust and security mechanisms employed as well as the technical details of the deployment framework for MHA cloud security. Security tools and policies that are developed ensure and enforce the legal and regulatory compliance and encompasses the appropriate auditing mechanisms that are needed by the legislation. This deliverable includes guidelines on how to pose these security mechanism in MHA platform and examples on how to use these guidelines for the MHA use cases.

4.4 MyHealthAvatar Data repositories

4.4.1 MHA central repository (Casandra)

The MHA central repository is used to store the data of the users of the MyHealthaVatar platform. This includes a range of user activities, health status related information, health profile and medical history. Cassandra is chosen for MHA data repository as described in *D6.2 Design for Data and RDF repositories*. A two-datacenter Cassandra cluster is deployed in the MHA hybrid cloud.

4.4.2 DICOM repository

For the needs of the MyHealthAvatar project, DCM4CHEE[20] was chosen to be used as a medical imaging DICOM repository. *DCM4CHEE* is a free and open - source DICOM archive and image manager, forming the server side of a PACS system. It is actively developed and updated, with modules including HL7 and WADO, and is based on JEE, JMX and the JBOSS Application Server. Administration is through a web-based interface and is compatible with a wide range of databases (PostgreSQL, MySQL, SQL Server and Oracle). The DCM4CHEE server has been already installed on a LINUX virtual machine

[18] Claudia Eckert. *IT-Sicherheit – Konzepte, Verfahren, Protokolle*. Oldenbourg Verlag, 5th edition, 2007. ISBN 978-3-486-58270-3.

[19] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography*. CRC Press series on discrete mathematics and its applications. CRC Press, 1997. ISBN 0-8493-8523-7.

[20] http://www.dcm4che.org/confluence/display/ee2/Home

(located at FORTH with IP address 139.91.210.41) and can be used directly by the partners for the needs of the project.

4.4.3 Model repository

In order to further facilitate citizen and patient self-assessment and education through the use of general models (such as diabetes risk, or educational oncosimulator implementations), the WP5 Tool/Model repository is divided into two parts. Within the MHA platform the Generic Tool/Model Repository part will reside. It will store these models and communicate with the platforms other parts such as the MHA central repository to utilize the collected data. The Nephroblastoma use case will utilize the other part, named the Onco-simulation Tool/Model repository.

4.5 Semantic Integration – Data access/model Layer

The semantic conceptual substrate of the overall information system described all components developed within WP4 and shown in the figure below (the highlighted ones in yellow). Bellow we will describe in detail each one of them.

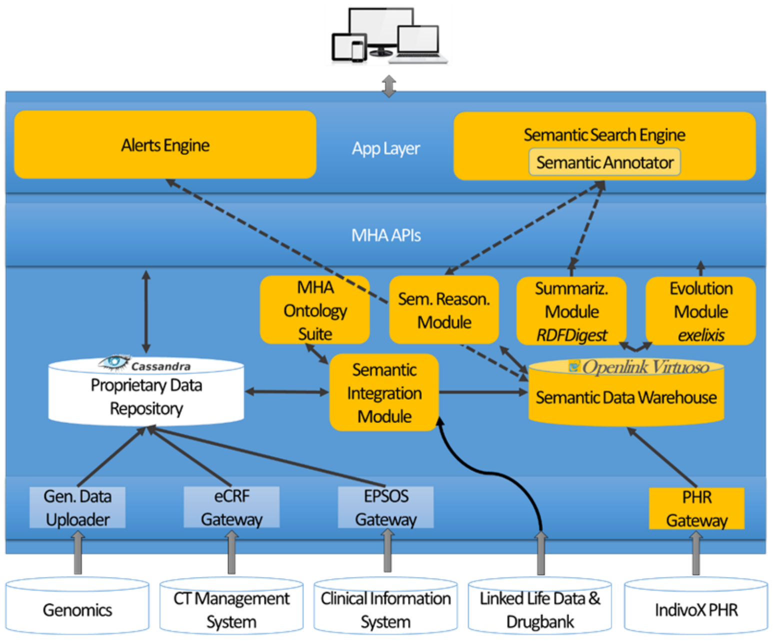


Figure 4-1: The interactions of components of WP4.

4.5.1 MyHealthAvatar Ontology Suite

This model has been developed for WP4 as a result of Task 4.1 and has been reported in D4.2[21].

[21] MyHealthAvatar Consortium, D4.1 Requirements analysis for semantic core ontology, July 2013

4.5.1.1 Functionality

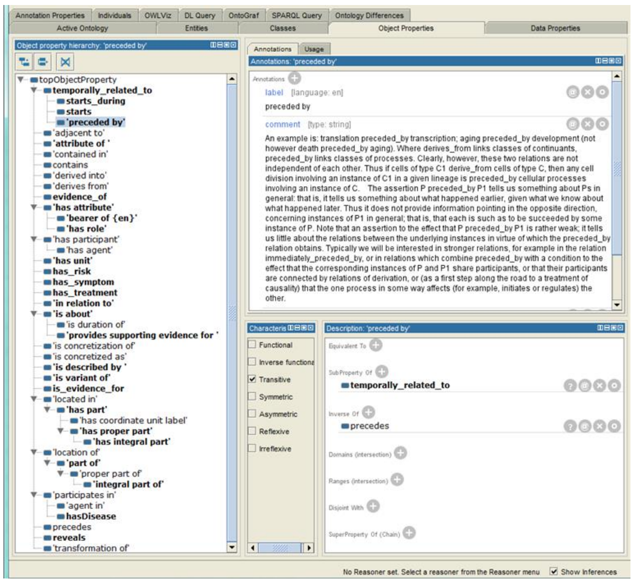
A modular ontology was generated to describe the details of the MyHealthAvatar clinical domains, the social life of the users and everyday activities. The ontology has been developed using RDF/S and it is consisted of an upper layer ontology the extended Translational Medical Ontology (eTMO) and 34 sub-ontologies with about 300 equivalence relations between them. A screenshot of the object property hierarchy of the eTMO is shown below whereas the all modules of the ontology[22] are shown in Figure 5-3[23].

4.5.1.2 Deployment Details

The ontology can be found online on the MyHealthAvatar (http://www.myhealthavatar.eu/) Ontology Portal. In addition it is used by all tools, components and services of WP4 in order to model the managed information.

4.5.1.3 Technical Evaluation

Since MHA Semantic Core Ontology is consisted of several subontologies, we have performed an extensive evaluation on each one of the used modules which presented in D4.1. The interested reader is forwarded to this deliverable for more information. In addition the ontology has shown its great value in being able to represent and manage all information managed within the MyHealthAvatar platform.



Object property hierarchy: 'preceded by'

```
topObjectProperty
    temporally_related_to
        starts_during
        starts
        'preceded by'
    'adjacent to'
    'attribute of '
    'contained in'
    contains
    'derived into'
    'derives from'
    evidence_of
    'has attribute'
        'bearer of {en}'
        'has role'
    'has participant'
        'has agent'
    'has unit'
    has_risk
    has_symptom
    has_treatment
    'in relation to'
    'is about'
        'is duration of'
        'provides supporting evidence for '
    'is concretization of'
    'is concretized as'
    'is described by '
    'is variant of'
    is_evidence_for
    'located in'
        'has part'
            'has coordinate unit label'
            'has proper part'
                'has integral part'
    'location of'
        'part of'
            'proper part of'
                'integral part of'
    'participates in'
        'agent in'
        hasDisease
    precedes
    reveals
    'transformation of'
```

Annotations: 'preceded by'

Annotations
label [language: en]
preceded by

comment [type: string]
An example is: translation preceded_by transcription; aging preceded_by development (not however death preceded_by aging). Where derives_from links classes of continuants, preceded_by links classes of processes. Clearly, however, these two relations are not independent of each other. Thus if cells of type C1 derive_from cells of type C, then any cell division involving an instance of C1 in a given lineage is preceded_by cellular processes involving an instance of C. The assertion P preceded_by P1 tells us something about Ps in general: that is, it tells us something about what happened earlier, given what we know about what happened later. Thus it does not provide information pointing in the opposite direction, concerning instances of P1 in general; that is, that each is such as to be succeeded by some instance of P. Note that an assertion to the effect that P preceded_by P1 is rather weak; it tells us little about the relations between the underlying instances in virtue of which the preceded_by relation obtains. Typically we will be interested in stronger relations, for example in the relation immediately_preceded_by, or in relations which combine preceded_by with a condition to the effect that the corresponding instances of P and P1 share participants, or that their participants are connected by relations of derivation, or (as a first step along the road to a treatment of causality) that the one process in some way affects (for example, initiates or regulates) the other.

Characteris
- ☐ Functional
- ☐ Inverse functiona
- ☑ Transitive
- ☐ Symmetric
- ☐ Asymmetric
- ☐ Reflexive
- ☐ Irreflexive

Description: 'preceded by'

Equivalent To

SubProperty Of
temporally_related_to

Inverse Of
precedes

Domains (intersection)

Ranges (intersection)

Disjoint With

SuperProperty Of (Chain)

No Reasoner set. Select a reasoner from the Reasoner menu ☑ Show Inferences

Figure 4-2: The object property hierarchy of eTMO Ontology

[22] MyHealthAvatar Consortium, D4.2 Extension of the semantic core ontology, February 2015

[23] Kondylakis, H., Spanakis, M. Sfakianakis, S., Sakkalis, V., Tsiknakis, M., Marias, K., Xia, Z., Yu, H.Q., Dong, F.: Digital Patient: Personalized and Translational Data Management through the MyHealthAvatar EU Project, IEEE International Conference of Engineering in Medicine and Biology Society (EMBC), Milan, Italy (2015)

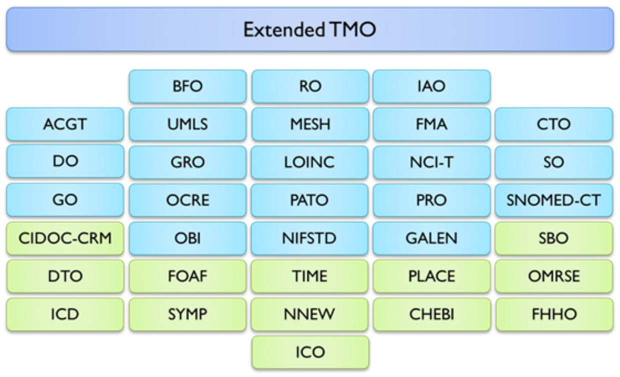


Figure 4-3: The modules of MyHealthAvatar Semantic Core Ontology[24]

4.5.2 Semantic Data Warehouse

This component has been developed for WP4 as a result of Task 4.3 and has been reported in D4.2.

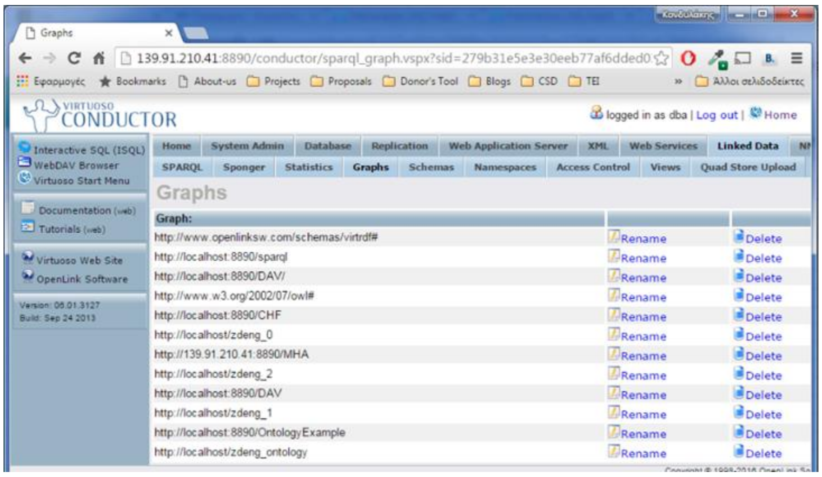
4.5.2.1 Functionality

The responsibility of this component is to store the linked, integrated information of the citizens. It is a central RDF Triple store relying on a Virtuoso instance. The main innovation of Virtuoso is that it delivers a platform-agnostic solution for data management, access and integration. It supports the management of various types of data, including relational, RDF, XML, text documents and others. This way, the users can employ the hybrid server architecture of Virtuoso to get access to all these different types of data. The component gets as input data from the PHR gateway and the Semantic Integration Module which are stored. Then those data are provided as input to the Semantic Reasoning Module, the Alerts Engine, the Semantic Search Engine and the Summarization and the Evolution Modules.

4.5.2.2 Deployment Details

Two Virtuoso Instances have been deployed on for development at FORTH's local cloud (139.91.210.41) and one stable at public cloud at the UK (178.79.142.72). Both instances offer a SPARQL endpoint for answering SPARQL queries and inserting new data. A screenshot of the Virtuoso Administration Environment is shown below.

[24] ACGT: ACGT Master Ontology, BFO: Basic Formal Ontology, CHEBI: Chemical Entities of Biological Interest, CIDOC-CRM: CIDOC Conceptual Reference Model, CTO: Clinical Trial Ontology, DO: Human Disease Ontology, DTO: Disease Treatment Ontology, FHHO: Family Health History Ontology, FMA: Foundation Model of Anatomy, FOAF: Friend of a Friend Ontology, GALEN: Galen Ontology, GO: Gene Ontology, GRO: Gene Regulation Ontology, IAO: Information Artifact Ontology, ICD: International Classification of Diseases, ICO: Informed Consent Ontology, LOINC: Logical Observation Identifier Names and Codes, MESH: Medical Subject Headings, NCI-T: NCI theraurus, NIFSTD: Neuroscience Information Framework Standardized ontology, NNEW: New Weather Ontology, OBI: Ontology for Biomedical Investigation, OCRE: Ontology for Clinical Research, OMRSE: Ontology of Medically Related Social Entities, PATO: Phenotypic Quality Ontology, PLACE:Place Ontology, PRO: Protein Ontology, RO: Relation Ontology, SBO: Systems Biology Ontology, SNOMED-CT: SNOMED linical terms, SO: Sequence Ontology, SYMP: Symptom Ontology, TIME: Time Ontology, UMLS:Unified Modeling Language System.



Graphs

Graph:		
http://www.openlinksw.com/schemas/virtrdf#	Rename	Delete
http://localhost:8890/sparql	Rename	Delete
http://localhost:8890/DAV/	Rename	Delete
http://www.w3.org/2002/07/owl#	Rename	Delete
http://localhost:8890/CHF	Rename	Delete
http://localhost/zdeng_0	Rename	Delete
http://139.91.210.41:8890/MHA	Rename	Delete
http://localhost/zdeng_2	Rename	Delete
http://localhost:8890/DAV	Rename	Delete
http://localhost/zdeng_1	Rename	Delete
http://localhost:8890/OntologyExample	Rename	Delete
http://localhost/zdeng_ontology	Rename	Delete

Figure 4-4: The Virtuoso Administration Environment

4.5.3 Exelixis - Semantic Integration Engine and Evolution Module

This component has been developed for WP4 as a result of Task 4.3 and has been reported in D4.2. In addition this component included algorithms for the integrating data under multiple ontology versions. These algorithms have been developed as a result of T4.4 and have been reported in D4.3[25].

4.5.3.1 Functionality

exelixis[26] is a data integration engine that a) achieves query answering by accepting SPARQL queries and b) extracts, transforms and loads underlying data to the semantic triple store. In the former case the input queries are then rewritten according to the source schemata and forwarded to the sources to be answered whereas in the latter cases the selected data are transformed and loaded to the Virtuoso triple store. In both cases the proper mappings should be established between the source schemata and the ontology. Although query rewriting always ensures accessing the latest information it suffers from efficiency problems and as such the ETL process is preferred. However, we have implemented a messaging notification queue allowing near real-time access to the latest information even in ETL scenarios. In such a case, as soon as a new information arrives to the underlying Cassandra databases, the corresponding service notifies the ETL process that new information should be extracted and loaded to the Virtuoso.

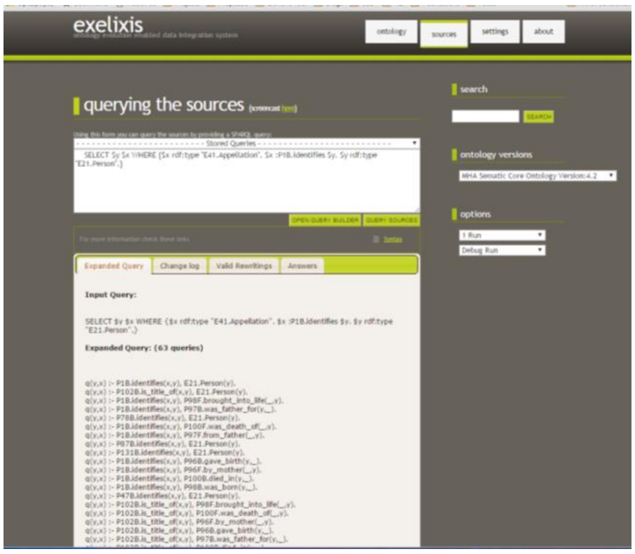
4.5.3.2 Deployment Details

Two versions of the semantic integration module have been deployed. One stable at public cloud in the UK (178.79.142.72) and one development version at (http://139.91.183.29:8080/exelixis/) for

[25] MyHealthAvatar Consortium, D4.3 Technical evaluation report of ontology including ontology evolution and summarization, November 2015

[26] Kondylakis, H., Plexousakis, D.: Ontology Evolution without Tears, Journal of Web Semantics (2013), 19, pp. 42-58, Elsevier

demonstration and development purposes. A screenshot of an exemplary interface showing query rewriting under different ontology versions is shown in below.



Using this form you can query the sources by providing a SPARQL query:

```
SELECT $y $x WHERE {$x rdf:type "E41.Appellation". $x :P1B.identifies $y. $y rdf:type
"E21.Person".}
```

For more information check these links

Input Query:

```
SELECT $y $x WHERE {$x rdf:type "E41.Appellation". $x :P1B.identifies $y. $y rdf:type
"E21.Person".}
```

Expanded Query: (63 queries)

```
q(y,x) :- P1B.identifies(x,y), E21.Person(y).
q(y,x) :- P102B.is_title_of(x,y), E21.Person(y).
q(y,x) :- P1B.identifies(x,y), P98F.brought_into_life(_,y).
q(y,x) :- P1B.identifies(x,y), P97B.was_father_for(y,_).
q(y,x) :- P78B.identifies(x,y), E21.Person(y).
q(y,x) :- P1B.identifies(x,y), P100F.was_death_of(_,y).
q(y,x) :- P1B.identifies(x,y), P97F.from_father(_,y).
q(y,x) :- P87B.identifies(x,y), E21.Person(y).
q(y,x) :- P131B.identifies(x,y), E21.Person(y).
q(y,x) :- P1B.identifies(x,y), P96B.gave_birth(y,_).
q(y,x) :- P1B.identifies(x,y), P96F.by_mother(_,y).
q(y,x) :- P1B.identifies(x,y), P100B.died_in(y,_).
q(y,x) :- P1B.identifies(x,y), P98B.was_born(y,_).
q(y,x) :- P47B.identifies(x,y), E21.Person(y).
q(y,x) :- P102B.is_title_of(x,y), P98F.brought_into_life(_,y).
q(y,x) :- P102B.is_title_of(x,y), P100F.was_death_of(_,y).
q(y,x) :- P102B.is_title_of(x,y), P96F.by_mother(_,y).
q(y,x) :- P102B.is_title_of(x,y), P96B.gave_birth(y,_).
q(y,x) :- P102B.is_title_of(x,y), P97B.was_father_for(y,_).
```

Figure 4-5: The exelixis system

4.5.3.3 Technical Evaluation

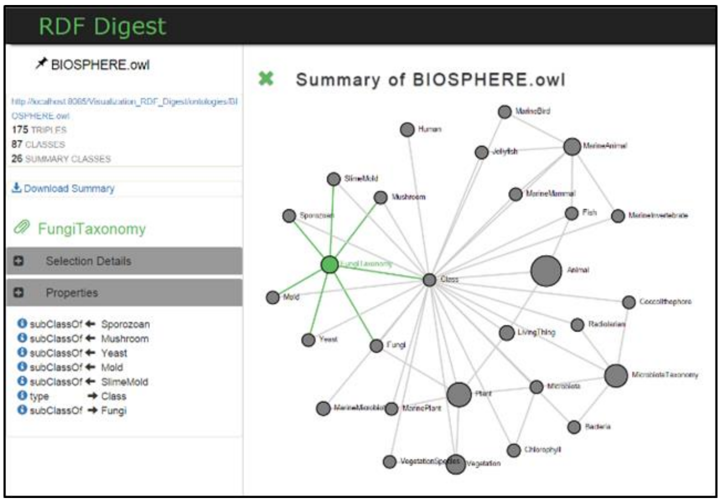
The exelixis system was evaluated using two sub-ontologies of the MHA Ontology Suite, namely the CIDOC CRM and the Gene Ontology. The potential impact of our approach is witnessed by being able to successfully provide rewritings on the worst case for the 88% of the CIDOC-CRM queries (after 711 change operations) and for the 97% of the GO queries (after 3482 change operations) among ontology versions. On the other hand if our system was not used, only a small percentage of the initial queries would be successful. For most of the queries, query answering is achieved within 5sec using a simple workstation, which also shows the usability and the scalability of our approach. The great benefit of our approach is the simplicity, modularity and the short deployment time it requires. It is only a matter of providing a new ontology version to our system to be able to use it to formulate queries that will be answered by data integration systems independent of the ontology version used. For the detailed technical evaluation of exelixis the interested reader is forwarded to the D4.4 and the corresponding publications.

4.5.4 RDF Digest - Semantic Summarization Module

This component has been developed for WP4 as a result of Task 4.4 and has been reported in D4.3.

4.5.4.1 Functionality

RDF Digest constructs high quality summaries as valid RDF/S graphs that include the most representative concepts of the schema, adapted to the corresponding instances. To construct this graph our algorithm exploits the semantics and the structure of the schema and the distribution of the corresponding data/instances. A screenshot of the interface of the system is shown below.



BIOSPHERE.owl

http://localhost:8085/Visualization_RDF_Digest/ontologies/BIOSPHERE.owl
175 TRIPLES
87 CLASSES
26 SUMMARY CLASSES

subClassOf ← Sporozoan
subClassOf ← Mushroom
subClassOf ← Yeast
subClassOf ← Mold
subClassOf ← SlimeMold
type → Class
subClassOf → Fungi

Figure 4-6: The RDF Digest system

4.5.4.2 Deployment Details

RDF Digest has been deployed as a web app in an Apache Tomcat Server at FORTH's local cloud ip (139.91.210.38). It is visible using the following url: www.ics.forth.gr/isl/rdf-digest/. In the same server a web service is running accepting as input the remote url of an ontology and returning the corresponding summary

4.5.4.3 Technical Evaluation

To create high quality summaries our algorithm exploits the semantics and structure of the schema and the distribution of the data by combining all these information using the relevance and the coverage properties. The performed evaluation verifies the feasibility of our solution and demonstrates the advantages gained by efficiently producing good summaries. Compared to other similar systems, our approach produces better results, further improved by exploiting knowledge about the instance distribution. Moreover, although most of the systems just select nodes or paths as the result summary, our result is a valid RDFS graph/document out of the initial RDF schema graph and can be used for query answering as well. For the detailed technical evaluation of RDF Digest the interested reader is forwarded to the D4.4 and the corresponding publications[27,28].

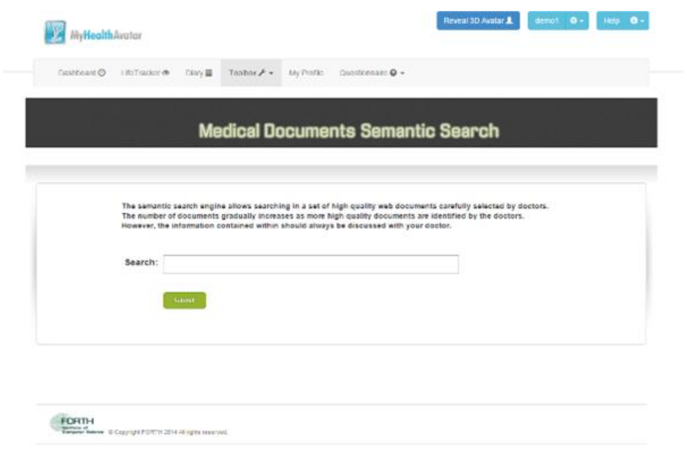
4.5.5 Semantic Search Engine (SSE)

This component has been developed for WP4 as a result of Task 4.5 and has been reported in D4.4[29].

[27] Troullinou, G., Kondylakis, H., Daskalaki, E., Plexousakis, D.: RDF Digest: Efficient Summarization of RDF/S KBs, Extended Semantic Web Conference (ESWC), pp 119-134, Portoroz, Slovenia (2015) (top 3 rated paper)
[28] Troullinou, G., Kondylakis, H., Daskalaki, E., Plexousakis, D.: RDF Digest: Ontology Exploration Using Summaries, International Semantic Web Conference (ISWC), Bethlehem, Pennsylvania (2015)
[29] MyHealthAvatar Consortium, D4.4 Semantic reasoning utilities for decision support, December 2015

4.5.5.1 Functionality

The semantic search engine (SSE) allows end-users to search in useful, selected, high-quality information. Medical experts can select useful web documents targeting patients that they can register to the engine using the semantic annotator app. Then an NLP annotator parses those documents using the MHA Ontology Suite and the annotations are stored in the internal database of the SSE. When end-users send a query to the SSE the query is annotated again using the same ontology, the annotated terms are expanded using the semantic reasoning module and the matched results are returned to the user.



The semantic search engine allows searching in a set of high quality web documents carefully selected by doctors.
The number of documents gradually increases as more high quality documents are identified by the doctors.
However, the information contained within should always be discussed with your doctor.

Search:

© Copyright FORTH 2014 All rights reserved.

Figure 4-7: The interface if the Semantic Search Engine

4.5.5.2 Deployment Details

The aforementioned app has been deployed as a web application at the stable version of the MyHealthAvatar portal at https://myhealthavatar.org/mha/.

4.5.5.3 Technical Evaluation

(Evaluation Pending)

4.5.6 Alerts Engine

This component has been developed for WP4 as a result of Task 4.5 and has been reported in D4.4.

4.5.6.1 Functionality

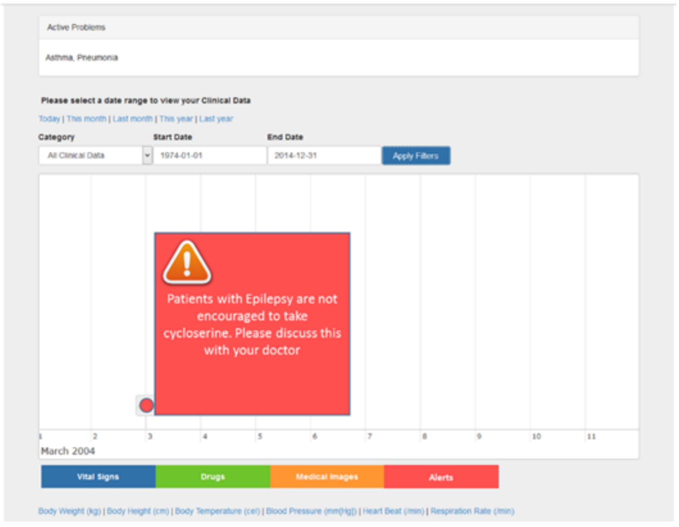
This engine is actually a service that gets as input a) the patient integrated profile within Virtuoso and b) rules described by clinical experts and identifies possible alerts and recommendations that should be presented to end-users such as possible drug interaction, side-effects etc. The semantic reasoning module that is used using the defined rules can produce new knowledge through inference using the integrated patient profile.

4.5.6.2 Deployment Details

The aforementioned app has been deployed as a web application at the stable version of the MyHealthAvatar portal at https://myhealthavatar.org/mha/.

4.5.6.3 Technical Evaluation

(Evaluation Pending)



Active Problems

Asthma, Pneumonia

Please select a date range to view your Clinical Data

Category: All Clinical Data

Start Date: 1974-01-01

End Date: 2014-12-31

Patients with Epilepsy are not encouraged to take cycloserine. Please discuss this with your doctor

Today | This month | Last month | This year | Last year
Apply Filters
1
2
3
4
5
6
7
8
9
10
11
March 2004
Vital Signs
Drugs
Medical Images
Alerts
Body Weight (kg) | Body Height (cm) | Body Temperature (cel) | Blood Pressure (mm[Hg]) | Heart Beat (/min) | Respiration Rate (/min)

Figure 4-8: Visualizing an alert

4.6 Auditing Service

A successful auditing system requires high quality information on events with potential hazardous security risks. The provision of this information is primarily the responsibility of the service provider, but well-coordinated cooperation between the developers of service providers, and audit engineers is necessary to enrich the audit messages with enough data to provide clear and sufficient audit trails in a production environment. The preferred audit data model is based on openXDAS2[30] for MHA. In deliverable D3.3 we provide a detailed description of the main elements of an audit elements and auditing concept architecture.

4.7 Linking with External Data Sources

4.7.1 Electronic Health Records

The “EPSOS Gateway” component enables MHA linking with a hospital information system to allow the exportation of the health related data of the patients into MHA platform. Although the abundance of standardized technologies accommodate the support of a large number of uses cases, due to the time constraints and the primary objectives of the project we focus on the use of epSOS Patient Summary for the retrieval of clinical data. The predominant issues relate to the security and transformation of the data followed by the proper annotation in order to be compliant with the syntactic and semantic principles of the system. This component described the technical implementation in respect to the legal work presented in WP11. The integration with the Hospital

[30] http://openxdas.sourceforge.net/architecture.html

information systems and other clinical data sources is described in Deliverable 3.4. The implementation of the services was done in WP3. The implementation is based on the EPSOS architecture and specifications and it consists of the following components:

- EPSOS-MHA "Gateway": the component of the External Linking layer of the MHA architecture that provides the bridge with the EPSOS infrastructure. It basically contacts the EPSOS "National Contact Points" (NCPs) in order to retrieve the "patient summaries" of the users of MHA.
- MHA "Clinical API" service: it provides a RESTful programmatic interface for delivering user related clinical information to other MHA components, such as the Portal, or third party applications that have been linked with the MHA platform.
- MHA "Event Bus": a message queue supporting the one-to-many communication in the platform based on the "publish-subscribe" message exchange pattern. This functionality is based on the MQTT[31] (formerly MQ Telemetry Transport), which is an ISO standard (ISO/IEC PRF 20922), and the actual broker used is mosquito[32]
- MHA PACS Server: the Data Management component that is responsible for the storage of clinical images in the DICOM format. We are using the open source DCM4CHEE PACS Server[33] for the storage and retrieval of the DICOM files.

The following picture depicts the integration of these components.

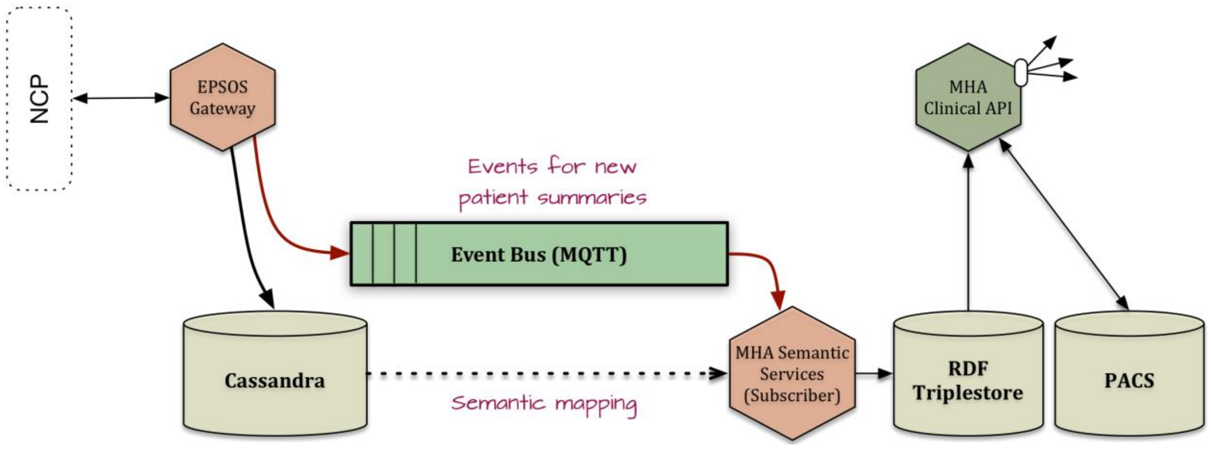


Figure 4-9: The main components that are responsible for the import, management, and the access of clinical data in the MHA platform

[31] Banks, Andrew, and Rahul Gupta. "MQTT Version 3.1.1" *OASIS Standard* (2014).
[32] http://mosquitto.org (accessed on February 25, 2016)
[33] http://www.dcm4che.org (accessed on February 25, 2016)

The incorporation of a messaging infrastructure (event bus) allows the efficient transfer of the data to the semantic layer where the semantic transformation, summarization, and harmonization take place.

4.7.2 Data repositories (CHIC repository)

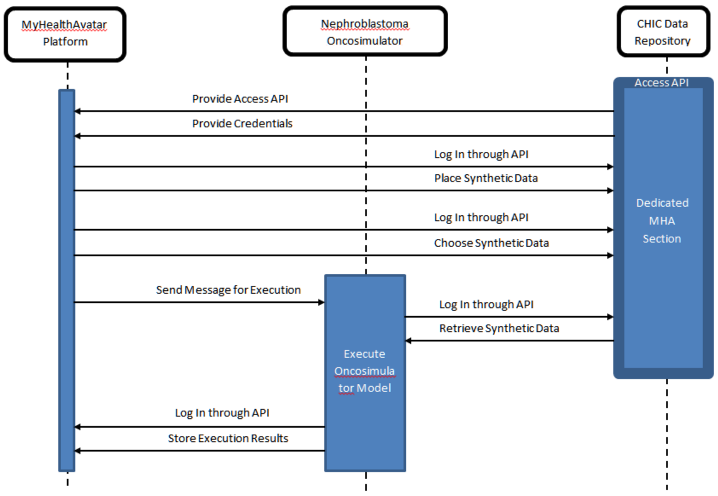
MyHealthAvatar has created a link to the CHIC project, by utilizing its Clinical Data Repository as an external data source.[34] The CHIC repository hosts all medical data produced or collected by the CHIC project and has interfaces to import or export its contents. The collected data are intended to be reusable by other projects. For each patient all the relevant medical data, including imaging data, clinical data, histological data and genetic data are stored.

Under the signed CHIC-MyHealthAvatar agreement, the CHIC repository will be involved in the following workflow pertaining to the Nephroblastoma Use Case:i) MyhealthAvatar partners will create a set of synthetic data for use as input for the Nephroblastoma Oncosimulator.

i) The CHIC project will create a section in the CHIC repository where these synthetic data will be stored. In addition it will provide credentials for access and application programming interfaces (access API) for data storage and retrieval, which will be used from MyHealthAvatar to log in and store their synthetic data

iii) For a Nephroblastoma Oncosimulator Execution, MyHealthAvatar will log to the CHIC repository via its API and chose the necessary data. Then the Nephroblastoma Oncosimulator will log to the CHIC repository, retrieve the data and proceed with its execution.

iv) The execution results will be stored back to the MyHealthAvatar Platform by the Nephroblastoma Oncosimulator, by using the platform's API.[35]



The figure on the right demonstrates the connections taking place in the aforementioned workflow. For the implementation of the described workflow, the Nephroblastoma Oncosimulator Application, which encases the simulation model, is build using python's Django. Therefore, libraries such as urllib2 and/or pycurl (where necessary), are used to make requests to the MHA platform and the CHIC repository API's, using the necessary credentials as headers. Through these requests, files are exchanged via streams

[34] Please see D11.4, section 5.6, pp. 66 ff. for the legal aspects.

[35] Please see D11.4, pp. 67 f. for more legal details.

(content_type='application/octet-stream) and the descriptive data are exchanged via JSON. The API documentations for the MHA platform and the CHIC repository are given through the corresponding links: https://myhealthavatar.org/api/doc/v2/index.html and https://cdr-chic.ics.forth.gr/api/help (production version). Since the CHIC project is still ongoing, there is also a development version of the repository's API: https://cdr-dev-chic.ics.forth.gr/api/help.

4.7.3 Drug data repositories

Personal health systems try to deal with issues of well-being, prevention and management of diseases and thus enhance patient empowerment and self-care management. To this respect a major challenge for patients –especially for chronic diseases where comorbidities may co-exist –is the optimum prescription of administered drugs in order to avoid any adverse drug reactions (ADRs), drug –drug interactions (DDIs) as well as patient's compliance with physician's instructions for lifestyle (i.e. alcohol consumption and smoking) and optimum drug administration. DDIs describe the modulation of action of one or more concurrently administered medications due to prescriptions errors or due to self – medication[36]. The problem is becoming complex taking into account patient's lifestyle and the use of alternative therapies (i.e. herbal medicines), alcohol consumption and smoking[37, 38]. The interaction mechanism can be related either with synergistic or antagonistic effects in the site of action (pharmacodynamic, PD) or with alteration of absorption, distribution, metabolism and elimination pathways (pharmacokinetic, PK).

In this section, a DDIs tool that was generated in the context of MyHealthAvatar (MHA) is described. The DDI tool in MHA platform was developed using the following programming languages and frameworks: RESTWeb services, Spring MVC 3 framework, Twitter Bootstrap, HTML5, Java server pages, JQuery, CSS, Apache Tomcat server. The information source in the case of MHA was through the Drugbank[39] bioinformatics and cheminformatics database which was provided via an XML file online. , it is parsed and stored in query optimized relational schema (in Postgresql). The complete functionality of the application is provided by two autonomous web services**.** The first service gives feedback regarding information for a specific drug when a full text search" in order to support non-exact string searching is typed. The feedback includes the name and the synonyms for the compound, a brief description of the pharmacological action along with the therapeutic category, route of administration and the DDIs that are known for this compound. Through this approach MHA manage

[36] P. D. Hansten, and J. R. Horn, Drug Interactions: Analysis and Management: Wolters Kluwer Health, 2006.
[37] I. Vizirianakis, M. Spanakis, A. Termentzi, I. Niopas, and E. Kokkalou, "Clinical and pharmacogenomic assessment of herb-drug interactions to improve drug delivery and pharmacovigilance," Plants in Traditional and Modern Medicine: Chemistry and Activity, pp. 978-81, 2010.
[38] . R. G. Smith, "An appraisal of potential drug interactions in cigarette smokers and alcohol drinkers," J Am Podiatr Med Assoc, vol. 99, no. 1, pp. 81-8, Jan-Feb, 2009.
[39] http://www.drugbank.ca/

to provide information for a patient-user regarding drugs that he/she may be receives. Using the MHA Semantic Core ontology[40], a modular ontology developed within MHA, we can describe prescribed drug information, and how these drugs can, harmfully or not, interact among each other. This integration is achieved by reusing terms from the sub-ontologies of the MHA Semantic Core Ontology. There are 34 sub-ontologies within MHA Sematic core ontology linked between to the eTMO ontology and via relations of equivalence (using owl:equivalentClass) and subsumption (rdfs:subClassof). One of these sub-ontologies is the DrugBank Ontology[41] with an equivalent link between the eTMO:Drug with the DrugBank:Drug. As such the entire ontology and information available in the DrugBank can directly be exploited and used. Some specific classes and used to represent patient medication information, DDIs and DFIs are shown in (figure 5-11). The second service that is provided through MHA is a DDIs checker. Through this service information for potential interactions between two drugs can be retrieved and provided to the user. The table below (Table 1) summarizes some characteristic paradigms of drug interactions that were retrieved from the implementation of the tool.

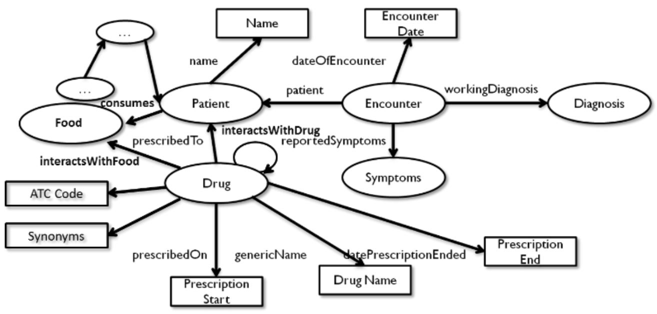


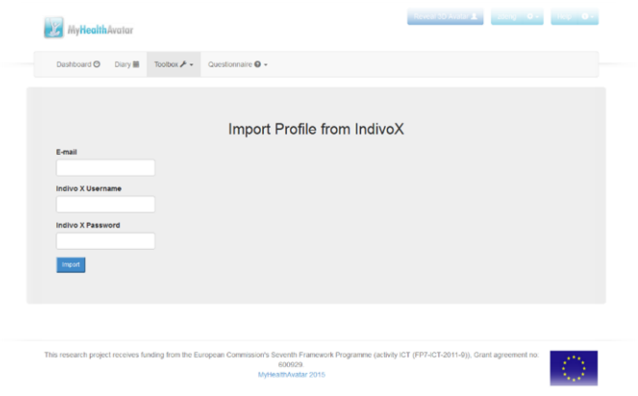
Figure 4-10: Classes from the MHA Semantic Core Ontology related medication DFIs and DDIs.

4.7.4 Personal Health Records

This component has been developed for WP4 as a result of Task 4.3 in order to demonstrate linkage to external data sources such as the IndivoX PHR system.

4.7.4.1 Functionality

The PHR gateway is actually an app of the MyHealthAvatar platform. As soon as the MyHealthAvatar user enters his IndivoX and presses the export button, his entire profile from the PHR is exported from IndivoX, transformed to



Import Profile from IndivoX

E-mail

Indivo X Username

Indivo X Password

This research project receives funding from the European Commission's Seventh Framework Programme (activity ICT (FP7-ICT-2011-9)). Grant agreement no: 600929.
MyHealthAvatar 2015

Figure 4-11 The interface

[40] D4.2 Extension of the semantic core ontology, D4.3 Technical evaluation report of ontology including ontology evolution and summarization
[41] https://datahub.io/dataset/bio2rdf-drugbank

RDF/S and loaded to his integrated profile at the Virtuoso Triple Store. A screenshot of the aforementioned interface is shown in 5-12.

4.7.4.2 Deployment Details

The aforementioned app has been deployed as a web application at the stable version of the MyHealthAvatar portal at https://myhealthavatar.org/mha/.

4.7.5 Link with Social Network Services

MHA infrastructure supports linking to social networks. MHA provides social web mechanisms and encourage the patients/citizens to adopt those in order to define their digital avatar. This requires integration with the social web accounts that the patients maintain already and the extraction of the social graph and other information. MHA is able to collect data from online patient diary using the utilities provided by T6.1. Volunteers will be organized to participate the research in this task. The task will continue towards the end of the project, leading to a considerable collection of information from the participants. The popularity of social media allows users to link their account profiles from social networks like Twitter, Facebook or PatientsLikeMe with MHA platform that become a communication hub for collecting people's personal stories and life experiences. MHA will then be able to contain a large volume of potential personal health information. The use of data mining techniques in the exploration of personal health information from these social networking services is the goal of this specific research task. The key problem we identified and focus on is on how to extract meaningful information from a large volume of data from popular social media services like Facebook, Twitter, etc. Details on the storage of these information have already been reported in D6.1.

We have explored and implemented the connection to the main social networks including Facebook, Twitter and Google +, we support connect the social network by their API (through OAuth 1 and 2). The detail of how it is connected and architecture graphic is described in previous deliverables. We have taken into consideration of Identity Federation with social networking, namely Login with Facebook/Twitter/Google, which would ease the user in terms of manage their credentials (this is work in progress). With users' explicit authorisation, MHA is able to read user's post, friend list, etc. and also able to post information back to the social network. E.g. Their daily activities from MHA. The post back to social network will increase the MHA exposure to general public, and potentially attracts new users to MHA. MHA also consider to allow users to invite friends from his social network, contact lists to MHA.

4.8 MyHealthAvatar API

4.8.1 Overview

MyHealthAvatar APIs can be accessed by implicit grant and authorization code grant. Client id and secret may be required for access.

4.8.2 MHA API Endpoints

The complete description of the MHA API can be found in *D3.6 Report on the Review of Open Source APIs for MHA*. This section lists the available endpoints for each API.

Activities

1) [GET] http://myhealthavatar.org/mha/api/v2/user/activities
2) [GET] http://myhealthavatar.org/mha/api/v2/user/activities?from={yyyy-mm-dd}
3) [GET] http://myhealthavatar.org/mha/api/v2/user/activities?to={yyyy-mm-dd}
4) [GET] http://myhealthavatar.org/mha/api/v2/user/activities?from={yyyy-mm-dd}&to={yyyy-mm-dd}
5) [GET] http://myhealthavatar.org/mha/api/v2/user/activities?date={yyyy-mm-dd}
6) [POST] http://myhealthavatar.org/mha/api/v2/user/activities?source=[fitbit, moves, withings]

Activity Daily Summary

1) [GET] http://myhealthavatar.org/mha/api/v2/user/activities/summary
2) [GET] http://myhealthavatar.org/mha/api/v2/user/activities/summary?from={yyyy-mm-dd}
3) [GET] http://myhealthavatar.org/mha/api/v2/user/activities/summary?to={yyyy-mm-dd}
4) [GET] http://myhealthavatar.org/mha/api/v2/user/activities/summary?from={yyyy-mm-dd}&to={yyyy-mm-dd}
5) [GET] http://myhealthavatar.org/mha/api/v2/user/activities/summary?date={yyyy-mm-dd}

Activity Segments

1) [GET] http://myhealthavatar.org/mha/api/v2/user/activities/segments
2) [GET] http://myhealthavatar.org/mha/api/v2/user/activities/segments?from={yyyy-mm-dd}
3) [GET] http://myhealthavatar.org/mha/api/v2/user/activities/segments?to={yyyy-mm-dd}
4) [GET] http://myhealthavatar.org/mha/api/v2/user/activities/segments?from={yyyy-mm-dd}&to={yyyy-mm-dd}
5) [GET] http://myhealthavatar.org/mha/api/v2/user/activities/segments?date={yyyy-mm-dd}
6) [GET] http://myhealthavatar.org/mha/api/v2/user/activities/segments?start_time={yyyy-mm-dd hh:mm:ss}

[GET]
http://myhealthavatar.org/mha/api/v2/user/activities/segments?start_time={yyyy-mm-dd hh:mm:ss}&end_time={yyyy-mm-dd hh:mm:ss}

Diary

1) [POST] http://myhealthavatar.org/mha/api/v2/user/diary
2) [GET] http://myhealthavatar.org/mha/api/v2/user/diary
3) [GET] http://myhealthavatar.org/mha/api/v2/user/diary?time=[yyyy-MM-dd'T'HH:mmssZ]
4) [GET] http://myhealthavatar.org/mha/api/v2/user/diary?from=[yyyy-MM-dd'T'HH:mmssZ]
5) [GET] http://myhealthavatar.org/mha/api/v2/user/diary?from=[yyyy-MM-dd'T'HH:mmssZ]&to=[yyyy-MM-dd'T'HH:mmssZ]
6) [POST] http://myhealthavatar.org/mha/api/v2/user/diary/delete?id=[local_id]

Measurements

1) [GET] http://myhealthavatar.org/mha/api/v2/user/measurements
2) [GET] http://myhealthavatar.org/mha/api/v2/user/measurements?from={yyyy-mm-dd}
3) [GET] http://myhealthavatar.org/mha/api/v2/user/measurements?to={yyyy-mm-dd}
4) [GET] http://myhealthavatar.org/mha/api/v2/user/measurements?from={yyyy-mm-dd}&to={yyyy-mm-dd}
5) [GET] http://myhealthavatar.org/mha/api/v2/user/measurements?date={yyyy-mm-dd}
6) [POST] http://myhealthavatar.org/mha/api/v2/user/measurements?source=[*]

General Health

1) [POST] http://myhealthavatar.org/mha/api/v2/user/general
2) [GET] http://myhealthavatar.org/mha/api/v2/user/general
3) [GET] http://myhealthavatar.org/mha/api/v2/user/general?from={yyyy-mm-dd}
4) [GET] http://myhealthavatar.org/mha/api/v2/user/general?to={yyyy-mm-dd}
5) [GET] http://myhealthavatar.org/mha/api/v2/user/general?from={yyyy-mm-dd}&to={yyyy-mm-dd}
6) [GET] http://myhealthavatar.org/mha/api/v2/user/general?date={yyyy-mm-dd}

User Profile

1) [POST] http://myhealthavatar.org/mha/api/v2/user/full_profile
2) [GET] http://myhealthavatar.org/mha/api/v2/user/full_profile
3) [GET] http://myhealthavatar.org/mha/api/v2/user/personal_information
4) [GET] http://myhealthavatar.org/mha/api/v2/user/insurance
5) [GET] http://myhealthavatar.org/mha/api/v2/user/allergy

6) `[GET] http://myhealthavatar.org/mha/api/v2/user/diagnosis`
7) `[GET] http://myhealthavatar.org/mha/api/v2/user/medication`
8) `[GET] http://myhealthavatar.org/mha/api/v2/user/vital_sign`

General Insurance Contact

1) `[GET] http://myhealthavatar.org/mha/api/v2/insurance_contact`
2) `[GET] http://myhealthavatar.org/mha/api/v2/insurance_contact?id=[uuid]`

Vital Sign Codes

`[GET] http://myhealthavatar.org/mha/api/v2/vital_sign_codes`

Sharing

1) `[POST] http://myhealthavatar.org/mha/api/v2/user/followers/request?user_name=**`
2) `[POST] http://myhealthavatar.org/mha/api/v2/user/followees/request?user_name=**`
3) `[POST] http://myhealthavatar.org/mha/api/v2/user/followers/accept?user_name=**`
4) `[POST] http://myhealthavatar.org/mha/api/v2/user/followees/accept?user_name=**`
5) `[POST] http://myhealthavatar.org/mha/api/v2/user/followers/update?user_name=**`
6) `[POST] http://myhealthavatar.org/mha/api/v2/user/followees/update?user_name=**`
7) `[POST] http://myhealthavatar.org/mha/api/v2/user/followers/delete?user_name=**`
8) `[POST] http://myhealthavatar.org/mha/api/v2/user/followees/delete?user_name=**`
9) `[GET] http://myhealthavatar.org/mha/api/v2/user/followers`
10) `[GET] http://myhealthavatar.org/mha/api/v2/user/followees`
11) `[GET] http://myhealthavatar.org/mha/api/v2/user/followers/request`
12) `[GET] http://myhealthavatar.org/mha/api/v2/user/followees/request`

4.8.3 MHA Clinical API

The Clinical API allows the retrieval of patient related clinical data using a RESTful (HTTP based) interface. A graphical depiction of its collaborators and interactions can be seen in Figure 5-12. The main functionalities and HTTP URIs supported are:

- Retrieve information about a DICOM series using GET at the `/series` endpoint with the series instance UID supplied in the `series_id` query parameter. The server contacts the backend DICOM server (using C-FIND) and returns information about the patient id, the study instance UID, and the list of images ("SOP instance UIDs") in JSON format.

- Retrieve the whole DICOM series as a ZIP file using GET at the `/dcm` endpoint with the series instance UID supplied in the `series_uid` query parameter. The images comprising the series are retrieved from the backend DICOM server using the C-FIND and C-GET commands.

- Retrieve a preview (JPEG image) of an instance (image) of a DICOM series using GET at the `/wado` endpoint given an "instance UID" in the instance_uid query parameter. The endpoint actually "proxies" the backend DICOM server using the WADO protocol[42].

- Finally, retrieve a patient's clinical summary in JSON format using GET at the `/patsum` endpoint. The data are retrieved from the backend TripleStore using the SPARQL query language. Additional query string parameters exist for specifying a date range: `from` is for the start of the date period, and `to` for the end for the period. The format of these dates should be compliant with the ISO format, e.g. "2008-04-21".

Additionally, the API supports the upload of a zip of DICOM images. The files are then sent to the backend DICOM server using the C-STORE command. The endpoint for this is the `/upload` using the POST HTTP method.

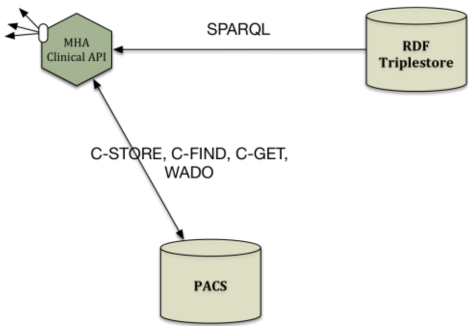


Figure 4-12: The integration of the Clinical API service with the data storage and harmonization components

4.9 Use case integration and evaluation

4.9.1 Diabetes

Diabetes is a lifelong condition problem which causes the level of a person's blood sugar to become too high. There are two main types of diabetes: Type 1 diabetes and Type 2 diabetes. Type 1 means that the pancreas doesn't produce any insulin and Type 2 means that the pancreas doesn't produce enough insulin or the body's cells don't react to insulin.

Diabetes is one of the popular disease in the world [WHO]. As the research of WHO (World Health Organization), there are about 60 million people with diabetes in the European Region in 2015. It includes about 10.3% of men and 9.6% of women aged 25 years and over. Moreover, the prevalence

[42] http://www.research.ibm.com/haifa/projects/software/wado/

of diabetes is increasing among all ages in the European Region. The mostly reason is that the population increases in overweight and obesity, unhealthy diet and physical inactivity.

Diabetes is a higher risk disease to cause death. As the research [WHO], in Worldwide, high blood glucose kills about 3.4 million people annually. Especially, almost 80% of these deaths occur in low- and middle-income countries, and almost half are people aged under 70 years. WHO projects diabetes deaths will double between 2005 and 2030. By suffering the diabetes, the people may be in danger of some series disease [Diabetes, Disease], such as:

1. Heart Disease – People with diabetes have a higher risk for heart attack and stroke.
2. Eye Complications – People with diabetes have a higher risk of blindness and other vision problems.
3. Kidney Disease – Diabetes can damage the kidneys and may lead to kidney failure.

Therefore, it is important for the people who has been diagnosed with diabetes to change their lifestyle with a confidence way. This section provides information about how diabetes can be fit around you and your life by using our Diabetes Program.

Diabetes Program is a native function that build in mobile application (MHA-pro) for Android operating system using android SDK versions 19-23. The application was developed in Android Studio. The MHA-pro is the extend version of current MHA application that provides program as shown in the following figure.

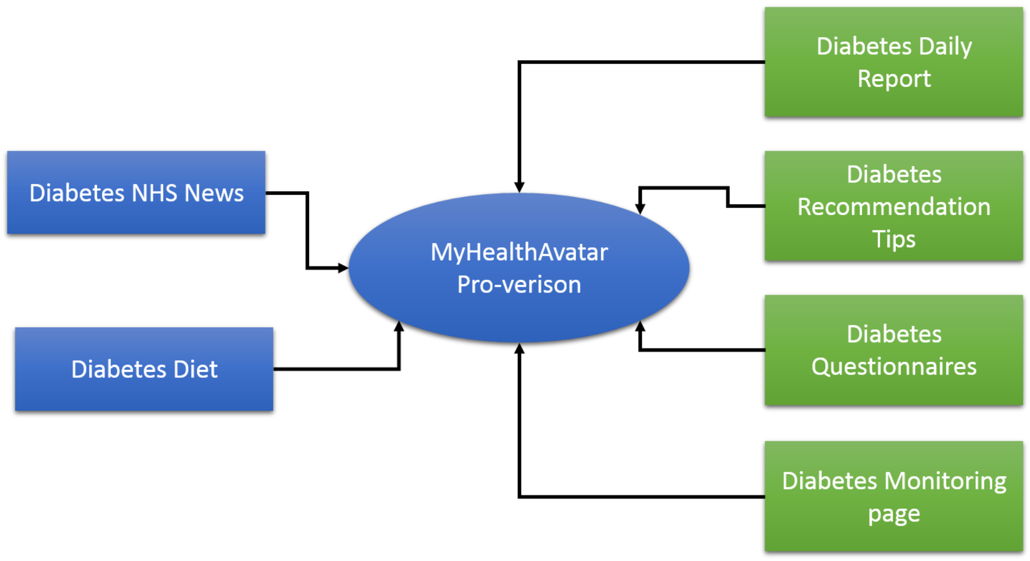


Figure 4-13: Structure of Diabetes Program in MyHeathAvatar Application Pro-version

In figure 5-13, the left side components are the default functions which already exist in MHA regular application. The HNS News can provide different topic health news and the diet control can offer users useful food information, e.g. calories, protein, sugar etc.; the right side components are the basic functions for the Diabetes Program. Diabetes daily report records all the monitoring data for every day. Diabetes recommendation tips record all the Helpful information for diabetes. Diabetes questionnaire and monitoring page inquiries the user's details to fill up the diabetes report. Therefore, the diabetes daily report is the main goals of the given program. The other components are assist the users to fill up the report.

In terms of the diabetes daily report, there are three different data type in the report, such as short-term, mid-term and long-term data as shown in Figure 5-14.

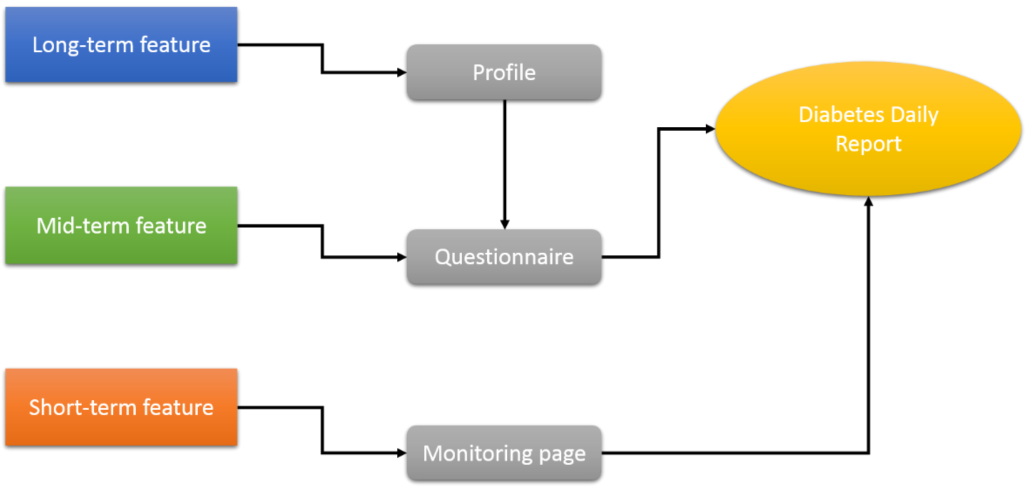


Figure 4-14: Data Collection Structure for Diabetes Daily Report

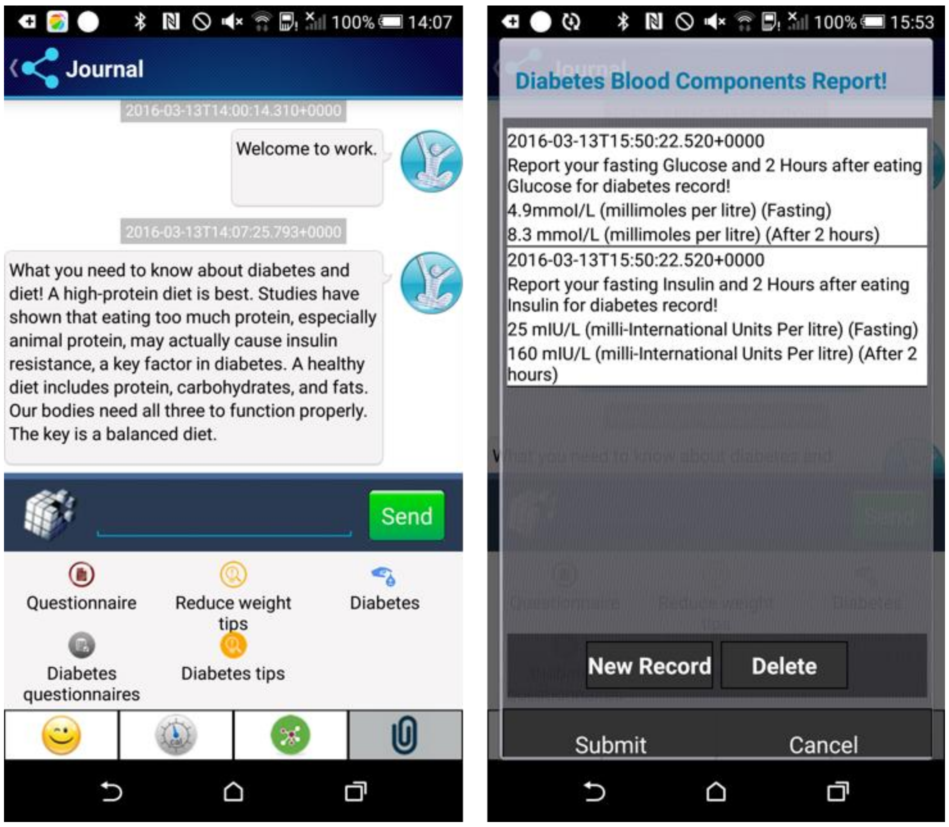
- Long-term feature means that the data is valid in long time period, e.g. gender, height, weight, family diabetes history etc. As shown in Figure 5.9.2, the Diabetes Program will obtain the data from the user profile first. If the profile does not include the required features, the program then inquiries the information through questionnaire method.
- Mid-term features means that the data will be updated every day, but only updated once a day, e.g. high sensitivity C-Reacting Protein (hsCRP), blood pressure and cholesterol etc. The mid-term features are obtained through questionnaire method which will be described in Section 5.9.1.2.
- Short-term features means that the data will be real-time updated in a day, e.g. glucose and insulin etc. the short-term feature are monitored through monitoring page which will be descript in Section 5.9.1.1.

4.9.1.1 Insulin and Glucose Monitoring and Visualisation

Monitoring your diabetes conditions is crucial to preventing some of the possible complications associated with diabetes. According to the above description, the insulin and glucose level are the important features to cope with diabetes.

Insulin is the hormone produced by the pancreas that allows glucose to enter the body's cells, where it is used as fuel for energy so we can work, play and generally live our lives. It is vital for life. Glucose comes from digesting carbohydrate and is also produced by the liver.

The monitoring function aims to monitor the users' fasting and 2 hours later blood glucose and insulin. Since eating is the important feature which affect users' blood glucose and insulin, it is very important to monitoring these values before and after eating. This function is built in Journal page as shown in Figure 5-15.


Journal

2016-03-13T14:00:14.310+0000

Welcome to work.

2016-03-13T14:07:25.793+0000

What you need to know about diabetes and diet! A high-protein diet is best. Studies have shown that eating too much protein, especially animal protein, may actually cause insulin resistance, a key factor in diabetes. A healthy diet includes protein, carbohydrates, and fats. Our bodies need all three to function properly. The key is a balanced diet.

Diabetes Blood Components Report!

2016-03-13T15:50:22.520+0000
Report your fasting Glucose and 2 Hours after eating Glucose for diabetes record!
4.9mmol/L (millimoles per litre) (Fasting)
8.3 mmol/L (millimoles per litre) (After 2 hours)

2016-03-13T15:50:22.520+0000
Report your fasting Insulin and 2 Hours after eating Insulin for diabetes record!
25 mIU/L (milli-International Units Per litre) (Fasting)
160 mIU/L (milli-International Units Per litre) (After 2 hours)

Figure 4-15: Glucose and Insulin Real-time Monitoring

In Figure 5-15, the first picture shows the control menu in Journal page. Since the current mobile application cannot detect the users's activity like eating or drinking, the users need to manually measure their glucose and insulin (In the future work, we can predict user start to eat according to their lifestyle and location). The users can press the diabetes button in control menu and trigger the glucose and insulin monitoring page (Diabetes Blood Components Report) as shown in second picture. The Users can add a new record, edit or delete a exist record. Once, the users submit the record, an 2 hours later alarm will be setup as well. This insure to remind users to measure the 2 hours later glucose and insulin.

Moreover, all these records will be saved in the daily report for future analysis purpose. The Users also can check their glucose and insulin records in the statistics in a time period as shown in Figure 5-16.

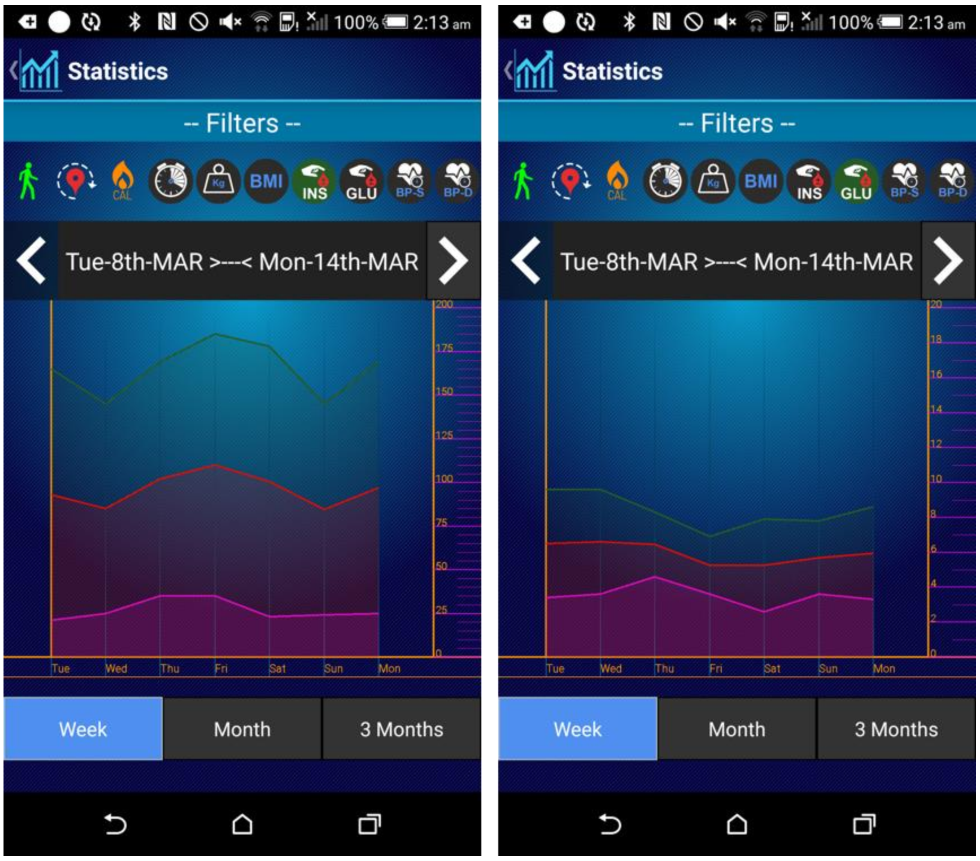


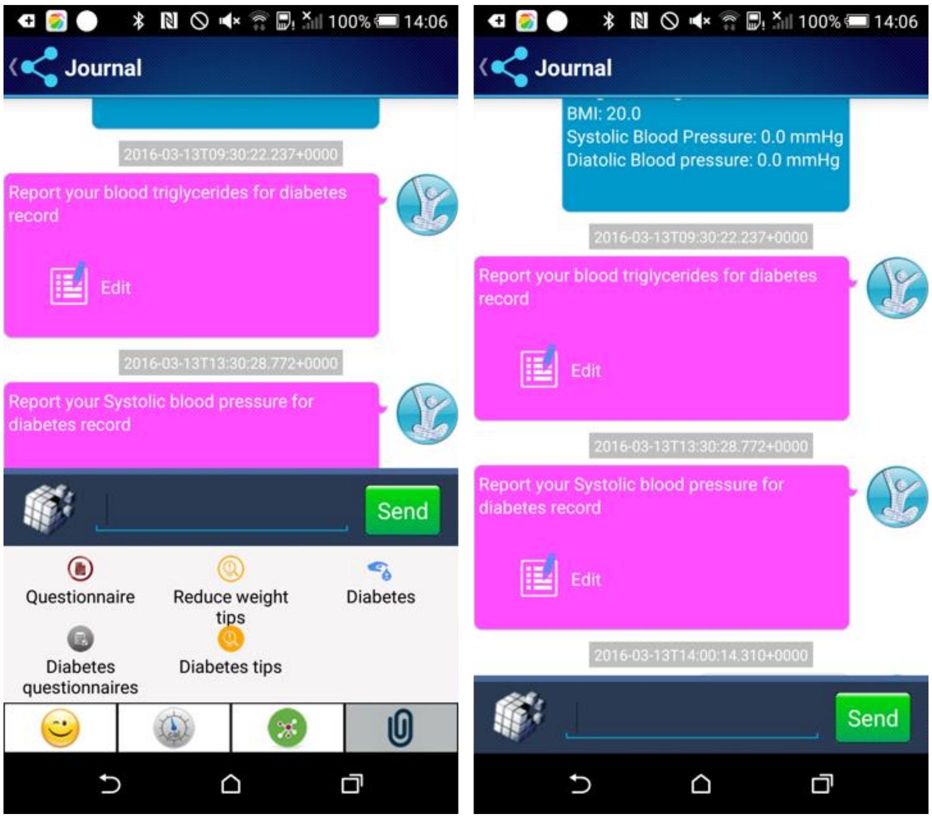
Figure 4-16: Insulin and Glucose Visualisation

In Figure 5-16, the purple curve represents the fasting insulin and glucose, the green curve represents the 2 hours later values and the red curve represents the average values. The figure shows that during a week period and the changes of your glucose and insulin. It is very easy to check how you improve your lifestyle quality through the changes of read curve, in order to overbear diabetes.

4.9.1.2 Diabetes Questionnaire

Diabetes Program aims not only to monitor the user's blood component but also to build a diabetes daily report for the future analysis. As mentioned before, in diabetes report, there are several conditions (features), such as weight, age, family diabetes history and blood cholesterol etc. Since most of the features are long-term and mid-term conditions, these conditions will be inquired by using questionnaires in the Journal page as shown in Figure 5-17.

In Figure 5-17, the first picture shows that user can manually to trigger the diabetes questionnaire. The second picture shows that user can update the conditions in Journal page. Moreover, MHA also support the time-based method to trigger the questionnaire as shown in Figure 5.9.6.



2016-03-13T09:30:22.237+0000

Report your blood triglycerides for diabetes record

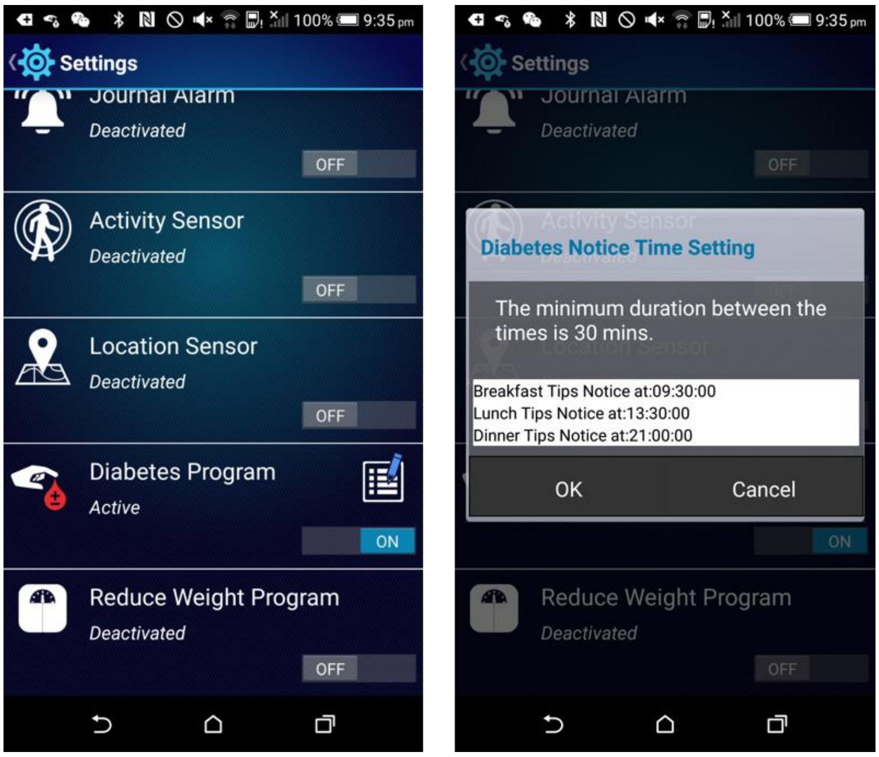
2016-03-13T13:30:28.772+0000

Report your Systolic blood pressure for diabetes record

BMI: 20.0
Systolic Blood Pressure: 0.0 mmHg
Diatolic Blood pressure: 0.0 mmHg

2016-03-13T14:00:14.310+0000

Figure 4-17: Diabetes Questionnaire



Journal Alarm
Deactivated OFF

Activity Sensor
Deactivated OFF

Location Sensor
Deactivated OFF

Diabetes Program
Active ON

Reduce Weight Program
Deactivated OFF

Diabetes Notice Time Setting

The minimum duration between the times is 30 mins.

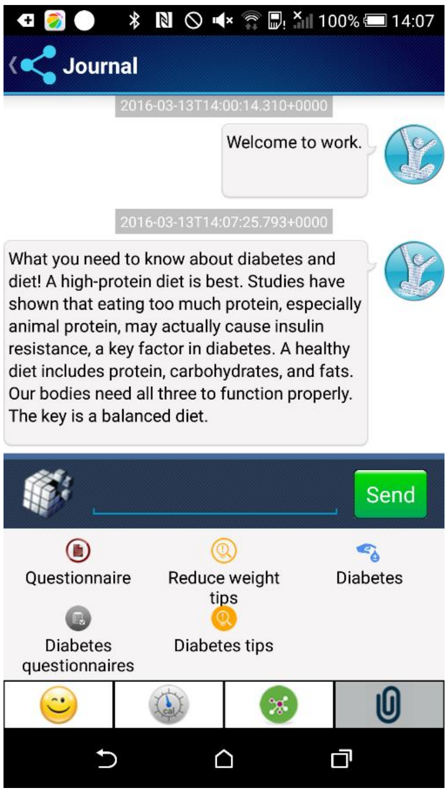
Breakfast Tips Notice at:09:30:00
Lunch Tips Notice at:13:30:00
Dinner Tips Notice at:21:00:00

Figure 4-18: Diabetes Program and Time-based Questionnaire in Settings Page

Figure 5-18 shows that the users can turn on the Diabetes Program in the settings page and then set the time-based questionnaire depends on their lifestyle.

4.9.1.3 Diabetes Tips and Diet

The diabetes tips can be triggered as same as the questionnaire, the users can easily let tips display in the Journal page through the control menu. If user has filled up all the conditions for diabetes report, the tips will be triggered instead of the questionnaire as shown in Figure 5-19.


Journal

2016-03-13T14:00:14.310+0000

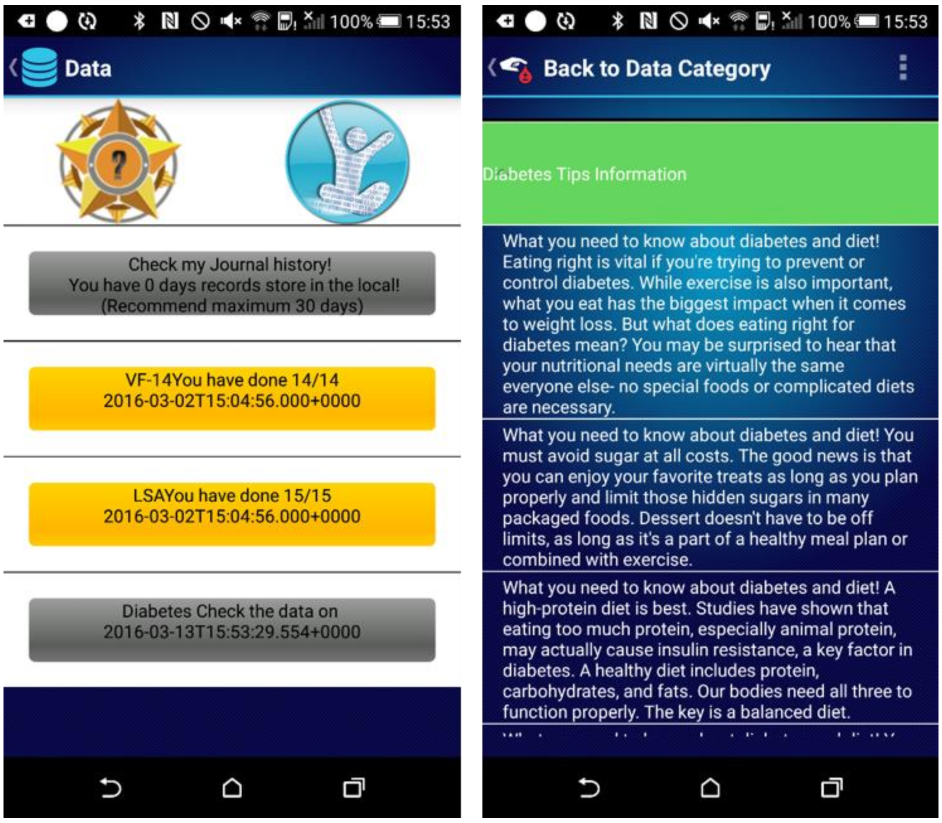
Welcome to work.

2016-03-13T14:07:25.793+0000

What you need to know about diabetes and diet! A high-protein diet is best. Studies have shown that eating too much protein, especially animal protein, may actually cause insulin resistance, a key factor in diabetes. A healthy diet includes protein, carbohydrates, and fats. Our bodies need all three to function properly. The key is a balanced diet.

Figure 4-19: Diabetes Tips in Journal Page

Users also can check all the tips in Data page as shown in Figure 5-20.



Check my Journal history!
You have 0 days records store in the local!
(Recommend maximum 30 days)

VF-14You have done 14/14
2016-03-02T15:04:56.000+0000

LSAYou have done 15/15
2016-03-02T15:04:56.000+0000

Diabetes Check the data on
2016-03-13T15:53:29.554+0000

Diabetes Tips Information

What you need to know about diabetes and diet! Eating right is vital if you're trying to prevent or control diabetes. While exercise is also important, what you eat has the biggest impact when it comes to weight loss. But what does eating right for diabetes mean? You may be surprised to hear that your nutritional needs are virtually the same everyone else- no special foods or complicated diets are necessary.

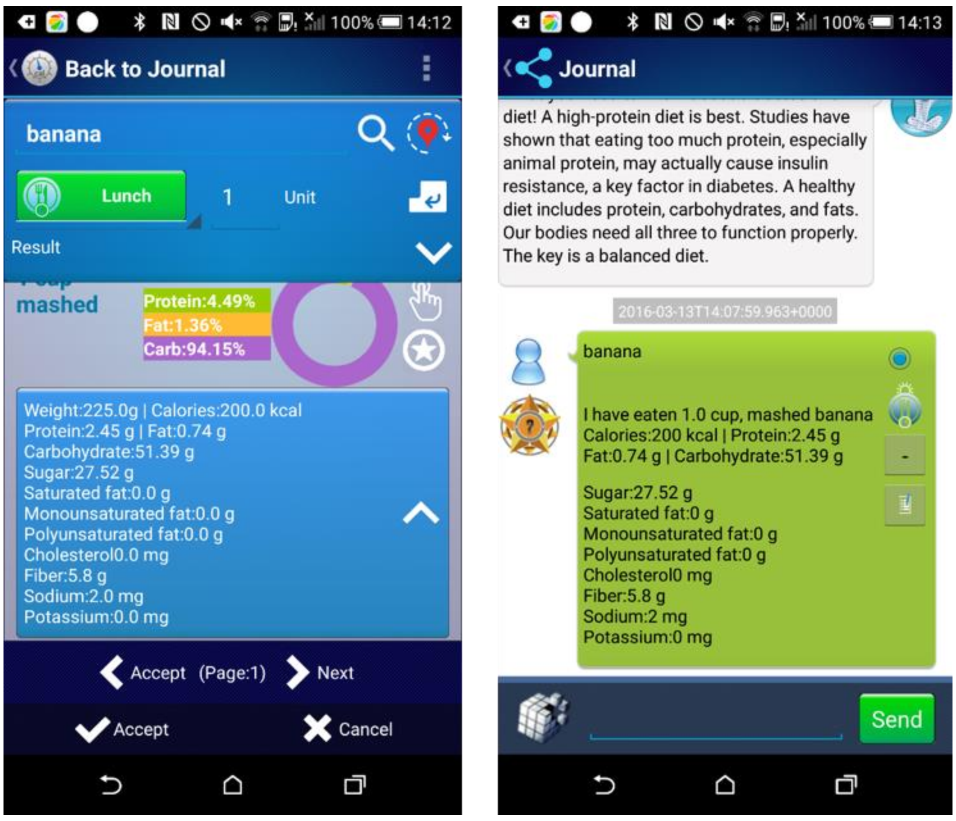
What you need to know about diabetes and diet! You must avoid sugar at all costs. The good news is that you can enjoy your favorite treats as long as you plan properly and limit those hidden sugars in many packaged foods. Dessert doesn't have to be off limits, as long as it's a part of a healthy meal plan or combined with exercise.

What you need to know about diabetes and diet! A high-protein diet is best. Studies have shown that eating too much protein, especially animal protein, may actually cause insulin resistance, a key factor in diabetes. A healthy diet includes protein, carbohydrates, and fats. Our bodies need all three to function properly. The key is a balanced diet.

Figure 4-20: Diabetes Tips in Data Page

A diabetes diet is simply a healthy eating plan that is high in nutrients, low in fat and added sugar, and moderate in calories [Diabetes, Diet]. The Diet control function can help the users choose the proper food (most notably the carbohydrates) and avoid the danger.

To use the diet control function, user can enter a food name through Journal page, then choose the proper food item and enter the counter for eating as shown in Figure 5-21.



Weight:225.0g | Calories:200.0 kcal
Protein:2.45 g | Fat:0.74 g
Carbohydrate:51.39 g
Sugar:27.52 g
Saturated fat:0.0 g
Monounsaturated fat:0.0 g
Polyunsaturated fat:0.0 g
Cholesterol0.0 mg
Fiber:5.8 g
Sodium:2.0 mg
Potassium:0.0 mg

diet! A high-protein diet is best. Studies have shown that eating too much protein, especially animal protein, may actually cause insulin resistance, a key factor in diabetes. A healthy diet includes protein, carbohydrates, and fats. Our bodies need all three to function properly. The key is a balanced diet.

2016-03-13T14:07:59.963+0000

banana

I have eaten 1.0 cup, mashed banana
Calories:200 kcal | Protein:2.45 g
Fat:0.74 g | Carbohydrate:51.39 g

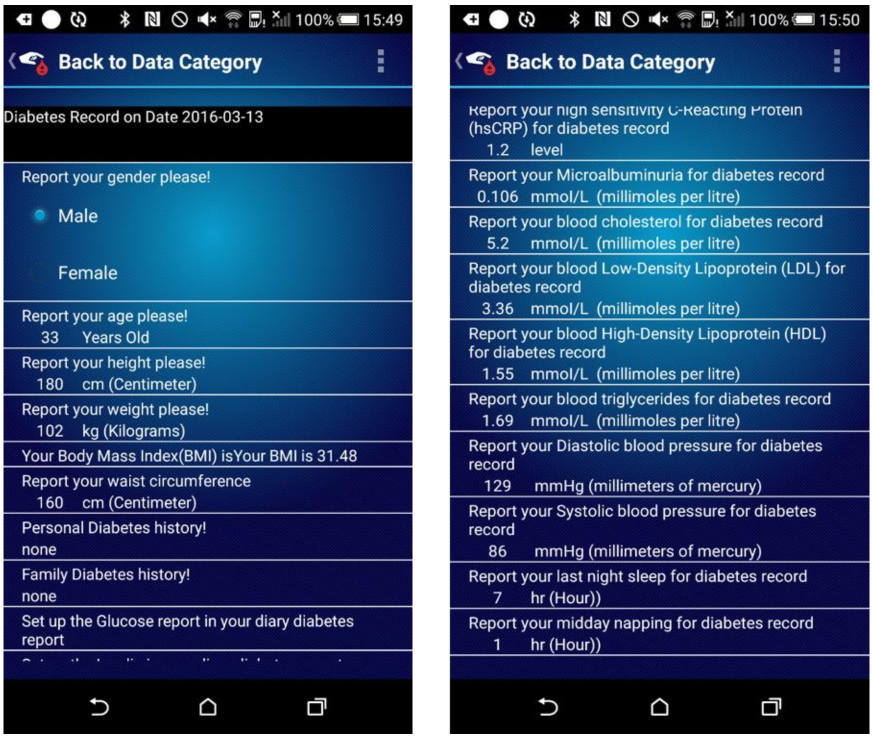
Sugar:27.52 g
Saturated fat:0 g
Monounsaturated fat:0 g
Polyunsaturated fat:0 g
Cholesterol0 mg
Fiber:5.8 g
Sodium:2 mg
Potassium:0 mg

Figure 4-21: Diet Control in Journal Page for Diabetes

In Figure 5-21, the user can easily to check the composition of the given food. This is very easy for person who suffering with diabetes to choose the proper food and control the glucose level.

4.9.1.4 Diabetes Daily Report

Diabetes daily report is crucial to preventing the conditions of your current diabetes situation. As mentioned before, this report includes many different type of conditions, user should give accurate information due to it is relative with your health. The more detail of the report, the more confident you will become and the easier you manage your diabetes. The detail of diabetes report is shown in Figure 5-22.


Back to Data Category

Diabetes Record on Date 2016-03-13

Report your gender please!
Male
Female

Report your age please!
33 Years Old

Report your height please!
180 cm (Centimeter)

Report your weight please!
102 kg (Kilograms)

Your Body Mass Index(BMI) isYour BMI is 31.48

Report your waist circumference
160 cm (Centimeter)

Personal Diabetes history!
none

Family Diabetes history!
none

Set up the Glucose report in your diary diabetes report

Back to Data Category

Report your high sensitivity C-Reacting Protein (hsCRP) for diabetes record
1.2 level

Report your Microalbuminuria for diabetes record
0.106 mmol/L (millimoles per litre)

Report your blood cholesterol for diabetes record
5.2 mmol/L (millimoles per litre)

Report your blood Low-Density Lipoprotein (LDL) for diabetes record
3.36 mmol/L (millimoles per litre)

Report your blood High-Density Lipoprotein (HDL) for diabetes record
1.55 mmol/L (millimoles per litre)

Report your blood triglycerides for diabetes record
1.69 mmol/L (millimoles per litre)

Report your Diastolic blood pressure for diabetes record
129 mmHg (millimeters of mercury)

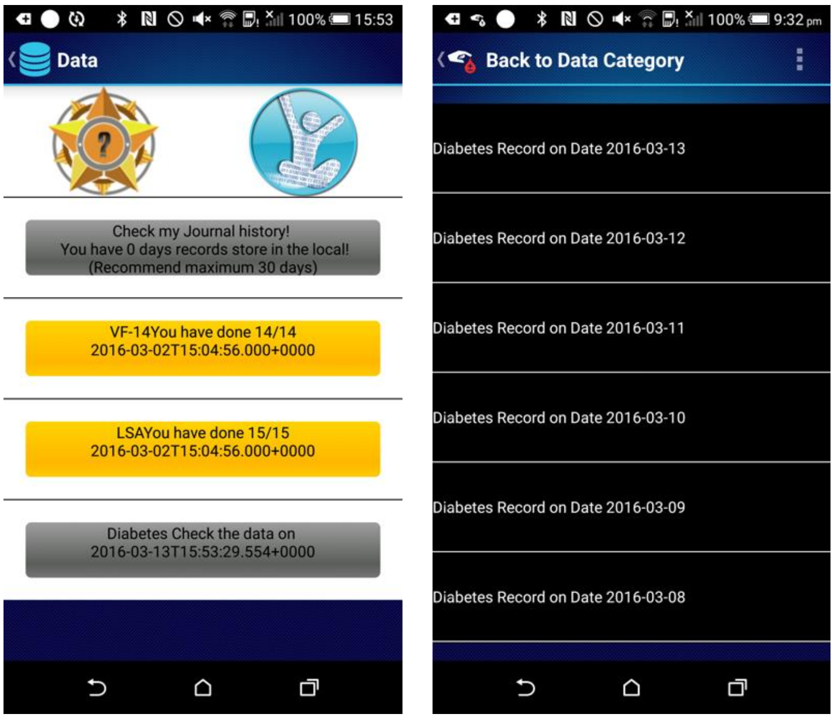
Report your Systolic blood pressure for diabetes record
86 mmHg (millimeters of mercury)

Report your last night sleep for diabetes record
7 hr (Hour))

Report your midday napping for diabetes record
1 hr (Hour))

Figure 4-22: Diabetes Daily Report

The diabetes report is a daily report, the user can check the all the report through the Data page in application as shown in Figure 5-23.


Data

Check my Journal history!
You have 0 days records store in the local!
(Recommend maximum 30 days)

VF-14You have done 14/14
2016-03-02T15:04:56.000+0000

LSAYou have done 15/15
2016-03-02T15:04:56.000+0000

Diabetes Check the data on
2016-03-13T15:53:29.554+0000

Back to Data Category

Diabetes Record on Date 2016-03-13

Diabetes Record on Date 2016-03-12

Diabetes Record on Date 2016-03-11

Diabetes Record on Date 2016-03-10

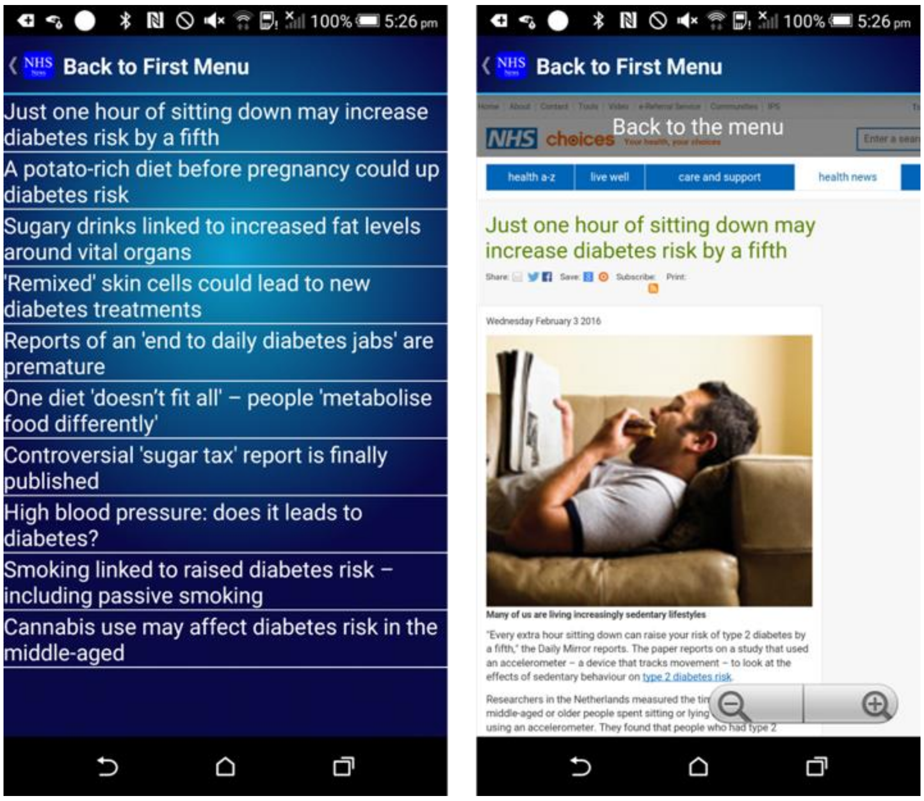
Diabetes Record on Date 2016-03-09

Diabetes Record on Date 2016-03-08

Figure 4-23: Diabetes Daily Records in Data Page

4.9.1.5 Diabetes NHS News

The diabetes NHS news is quite similar to the diabetes tips. It uses to give the suggestion to the user who suffering with diabetes. Once the user has turn on the Diabetes Program in settings page as shown in Figure 5.9.6, the application will set the NHS news topics to diabetes. Therefore, the system will automatically obtain the data from NHS website and the user can easily to check the diabetes news from the Journal page. The NHS news is shown in Figure 5-24.



Just one hour of sitting down may increase diabetes risk by a fifth

A potato-rich diet before pregnancy could up diabetes risk

Sugary drinks linked to increased fat levels around vital organs

'Remixed' skin cells could lead to new diabetes treatments

Reports of an 'end to daily diabetes jabs' are premature

One diet 'doesn't fit all' – people 'metabolise food differently'

Controversial 'sugar tax' report is finally published

High blood pressure: does it leads to diabetes?

Smoking linked to raised diabetes risk – including passive smoking

Cannabis use may affect diabetes risk in the middle-aged

Just one hour of sitting down may increase diabetes risk by a fifth

Wednesday February 3 2016

Many of us are living increasingly sedentary lifestyles

"Every extra hour sitting down can raise your risk of type 2 diabetes by a fifth," the Daily Mirror reports. The paper reports on a study that used an accelerometer – a device that tracks movement – to look at the effects of sedentary behaviour on type 2 diabetes risk.

Researchers in the Netherlands measured the tim middle-aged or older people spent sitting or lying using an accelerometer. They found that people who had type 2

Figure 4-24: NHS News for Diabetes

4.9.1.6 Testing and Evaluation

Feature	Expected Functionality	Actual
Glucose and insulin Real-time monitoring	Users can easily report the Glucose and Insulin values	Users can easily report their fasting and 2 hours later glucose and insulin values.
Glucose and insulin Real-time monitoring	2 hours reminder alarm works properly	The reminder works fine
Glucose and insulin Real-time monitoring	Data has completely and accurately display in the statistic page	Data correctly displays in the statistic page

Glucose and insulin Real-time monitoring	Data has completely and accurately saved in the diabetes report	Data correctly saves in the diabetes report
Diabetes Questionnaire	Questions completely and accurately display in the Journal page	Questions correctly display in the Journal page
Diabetes Questionnaire	Data has completely and accurately saved in the diabetes report	Data correctly saves in the diabetes report
Diabetes Questionnaire	Questions has completely and accurately display in the Journal page according to the time-based triggering method	Questions correctly display in the Journal page
Diabetes Tips	Tips has completely and accurately display in the Journal page	Tips correctly displays in the Journal page ISSUE: on devices with a small screen, it is difficult to see the entire information of a recommendation tips, due to the large detail of the tips
Diabetes Tips	Tips has completely and accurately display in the Journal page according to the time-based triggering method	Tips correctly displays in the Journal page
Diabetes Diet Control	Food information has completely and accurately display in the food search page	Data correctly displays in the food search page
Diabetes Diet Control	Food information has completely and accurately display in Journal page	Data correctly displays in the Journal page ISSUE: the user has to search the several food, it may cost a bit long time. Solution: the application support the users to save their favourite food, in order to improve the efficiency of using this function
Diabetes Daily Report	Data has completely and accurately display and stored in the diabetes report	Data correctly displays and stores in the diabetes report
Diabetes Daily Report	Data has correctly recorded in every day	Diabetes report record function works fine
Diabetes Daily Report	Data can be updated depends on the Profile	Data can be correctly updated depends on the Profile
Diabetes NHS News	NHS news page has completely and accurately display the diabetes information	NHS news works fine

4.9.2 Personalized CHF Related Risk Profiles and "Real-Time Monitoring" (CHF)

Congestive heart failure (CHF) is a state in which the heart cannot provide sufficient cardiac output to satisfy the metabolic needs of the body. It is commonly termed congestive heart failure (CHF) since symptoms of increase venous pressure are often prominent. Its pathogenesis factors include: *Age, Gender, Increased blood pressure, Smoking, Alcohol, Family and medical histor, Genetic predisposition, Diabetes, Diet habits and Atherosclerosis*. It's a pathophysiologic state in which the heart, via an abnormality of cardiac function (detectable or not), fails to pump blood at a rate commensurate with the requirements of the metabolizing tissues or is able to do so only with an elevated diastolic filling pressure. Common causes of the disease include coronary heart disease, hypertension and valvular heart disease. Diagnosis can be achieved through physical examination (i.e. blood pressure, body mass index, blood tests) and echocardiography. A major challenge related to caring for patients with chronic conditions is the early detection of exacerbations of the disease that may be of great significance.

In this use case and demonstration scenario we focus on methodologies that would facilitate the early detection and monitoring of CHF exacerbation, enabling prevention on a daily basis.

Especially, early detection is of upmost importance; hence remote health monitoring systems are in the research focus so as to provide to a doctor the ability to monitor the progress of a patient on a daily basis and issue alerts in case of potential health risks. The objective thus is to create a service able to empower citizens, patients and doctors by providing a supportive environment for the self-management of patients/ citizens with cardiovascular disease risks. To do so we incorporate a pool of verified risk assessment models for cardiovascular diseases into the MyHealthAvatar platform and an external to MyHealthAvatar mobile application for real time monitoring and intelligent rule-based alerting in case of an eminent CHF episode.

We define the "*CHF Real-time patient monitoring"* and the "*CHF Risk Assessment"* service in order to: assist individualized out self-monitoring of their own health-status, provide risk analysis for personal risk monitoring for developing a cardiovascular related episode in the future, provide comorbidities and drug interaction information in both the treating physicians, but also the patient him/ herself regarding negative drug interactions (optional).

CHF real-time patient monitoring is realized by a mobile application that monitors and records the vital signs (such as heart rate, oxygen saturation, systolic blood pressure and diastolic blood pressure) of patients with CHF or prone to develop CHF. It exploits sensor technologies to obtain data via Bluetooth, and wireless communications to share data with back-end databases. Vital signs are recorded every second locally in SQLite, and in case of an abnormal measurement detection, an alert record is created accompanied by a notification (

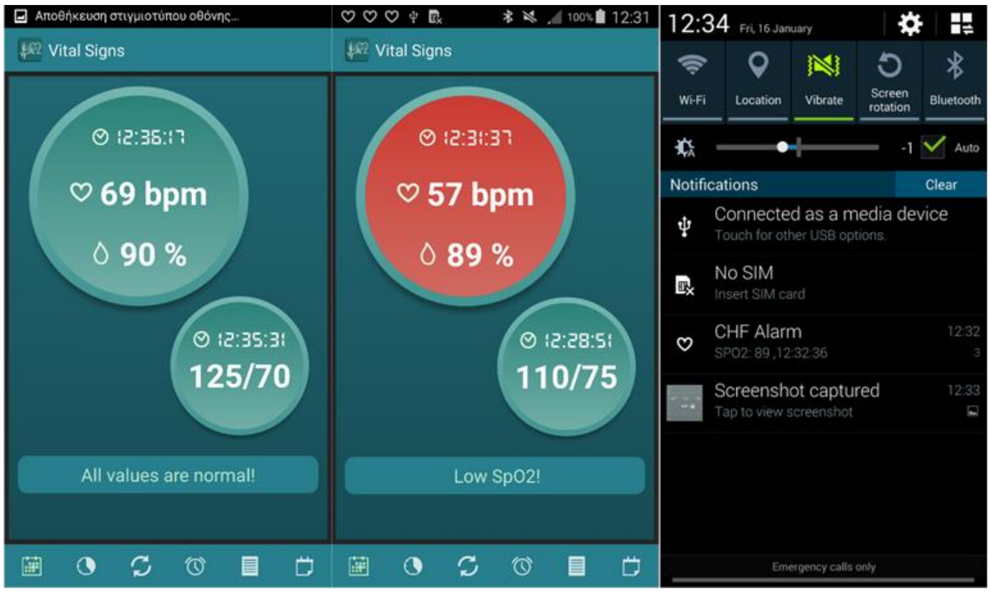


Figure 4-25). Moreover, the application provides useful charts for visual display of the vital signs' measurements (Figure 4-26). Data are sent to the MyHealthAvatar platform at the request of the user/ patient, and namely the list of alerts including the type of the alert, the value and the time it occurred (Figure 4-27).

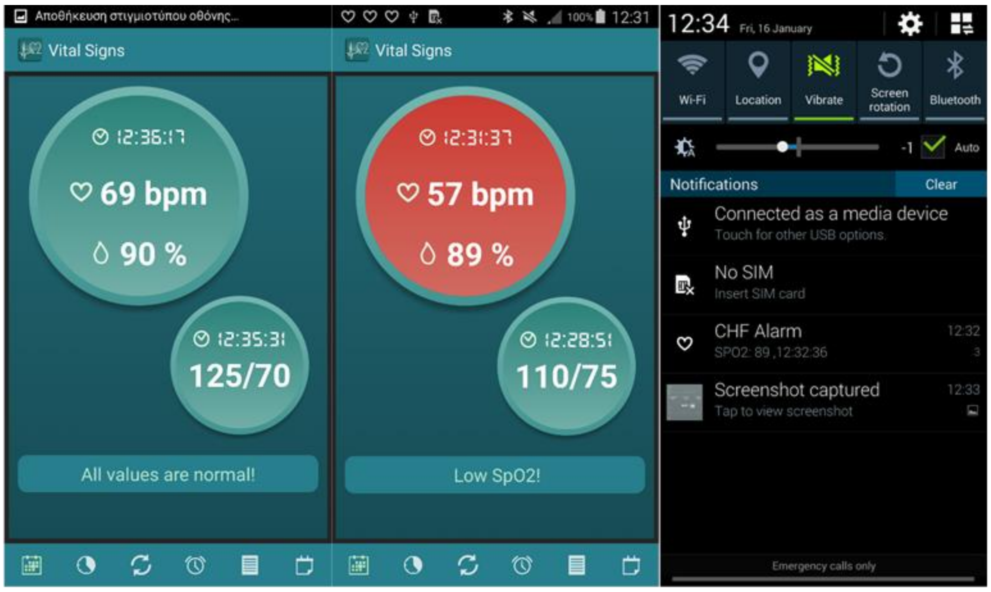
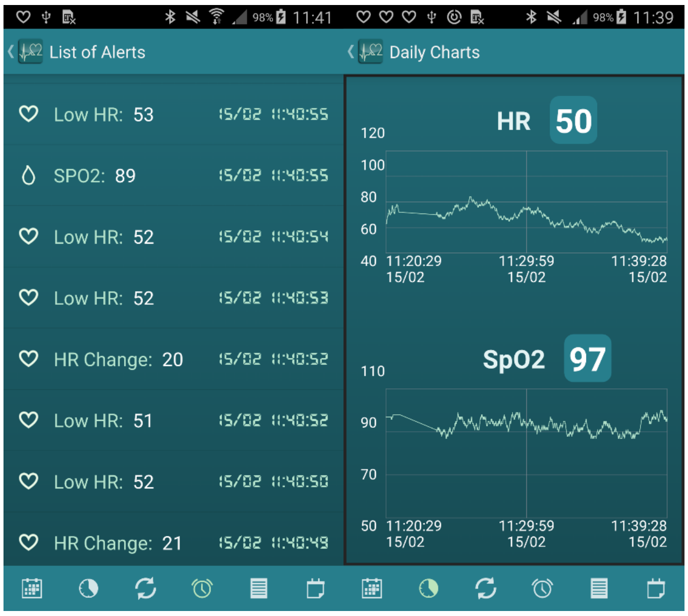
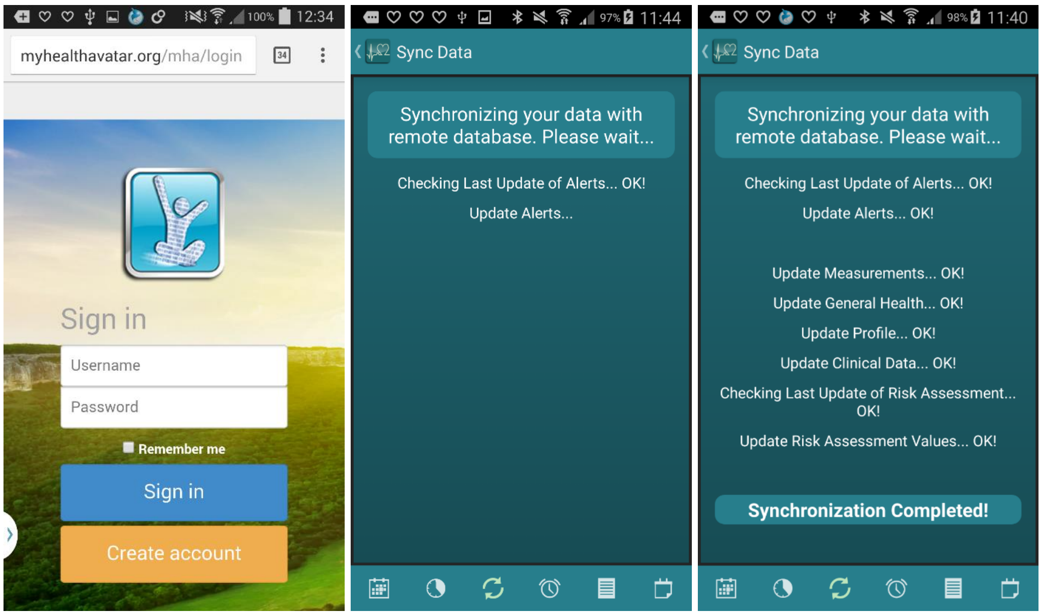


Figure 4-25: Vital signs' monitoring


List of Alerts

Low HR: 53	15/02 11:40:55
SPO2: 89	15/02 11:40:55
Low HR: 52	15/02 11:40:54
Low HR: 52	15/02 11:40:53
HR Change: 20	15/02 11:40:52
Low HR: 51	15/02 11:40:52
Low HR: 52	15/02 11:40:50
HR Change: 21	15/02 11:40:49

Figure 4-26: List of Alerts, Notifications and Vital Signs' Charts



Synchronizing your data with remote database. Please wait...

Checking Last Update of Alerts... OK!

Update Alerts...

Synchronizing your data with remote database. Please wait...

Checking Last Update of Alerts... OK!

Update Alerts... OK!

Update Measurements... OK!

Update General Health... OK!

Update Profile... OK!

Update Clinical Data... OK!

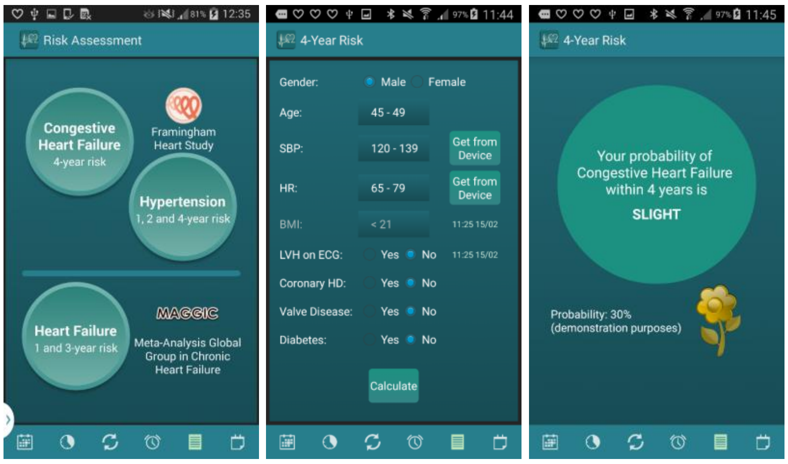
Checking Last Update of Risk Assessment... OK!

Update Risk Assessment Values... OK!

Synchronization Completed!

Figure 4-27: Synchronize data with MyHealthAvatar platform

In addition to the real-time monitoring, the *CHF Risk Assessment application* provides to the patient self-health status assessment services based on validated risk evaluation algorithms (Figure 4-28). In particular, three risk assessment models have been implemented providing longer (3-, 4-year) and shorter (1-, 2- year) heart failure risk assessment. Figure 4-2828 to Figure 4-30 display the algorithms' questionnaires with the respective results' screens, whereas Figure 4-31 shows the list of all the calculated risk scores. The bellow color conventions are followed: *green* for "low risk", *yellow* for "slight risk" and *orange* for "considerable risk". Data can be retrieved from the MyHealthAvatar platform upon the request of the user/ patient so as to be used in the algorithms, such as measurements, general health, profile and clinical data. Moreover, the list of risk scores is sent to MyHealthAvatar.


Risk Assessment
Congestive Heart Failure 4-year risk
Framingham Heart Study
Hypertension 1, 2 and 4-year risk
Heart Failure 1 and 3-year risk
MAGGIC
Meta-Analysis Global Group in Chronic Heart Failure

4-Year Risk

Gender:	[✓ Male] Female	
Age:	45 - 49	
SBP:	120 - 139	Get from Device
HR:	65 - 79	Get from Device
BMI:	< 21	11:25 15/02
LVH on ECG:	Yes [✓ No]	11:25 15/02
Coronary HD:	Yes [✓ No]	
Valve Disease:	Yes [✓ No]	
Diabetes:	Yes [✓ No]	

Calculate

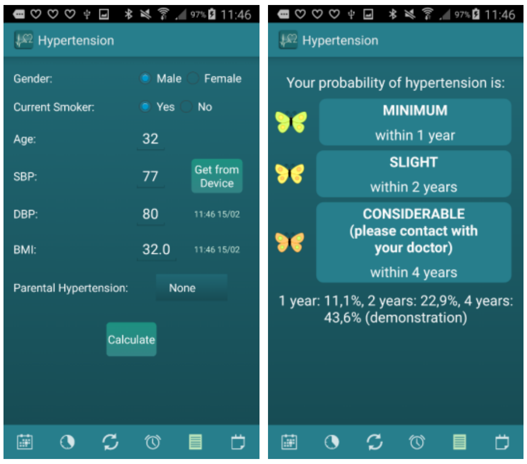
4-Year Risk

Your probability of Congestive Heart Failure within 4 years is

SLIGHT

Probability: 30%
(demonstration purposes)

Figure 4-28: Risk Assessment Algorithms, Framingham CHF risk assesment (picture on the right)



Hypertension

Gender:	[✓ Male] Female	
Current Smoker:	[✓ Yes] No	
Age:	32	
SBP:	77	Get from Device
DBP:	80	11:46 15/02
BMI:	32.0	11:46 15/02
Parental Hypertension:	None	

Calculate

Hypertension

Your probability of hypertension is:

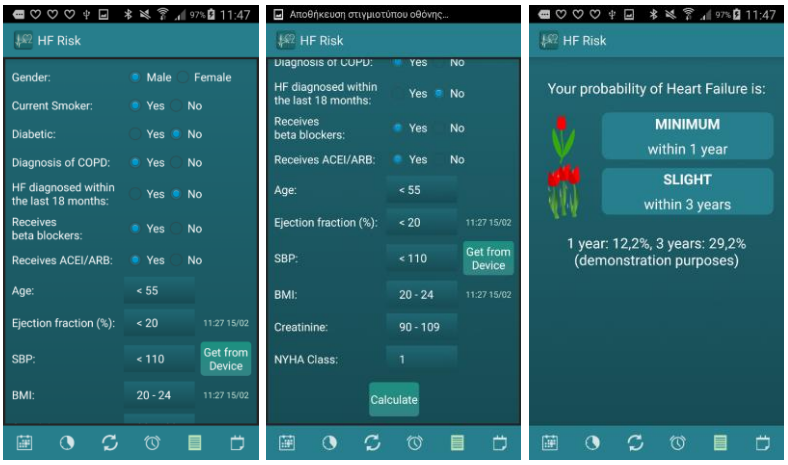
MINIMUM within 1 year

SLIGHT within 2 years

CONSIDERABLE (please contact with your doctor) within 4 years

1 year: 11,1%, 2 years: 22,9%, 4 years: 43,6% (demonstration)

Figure 4-29: Framingham Hypertension



HF Risk

Gender:	[✓ Male] Female	
Current Smoker:	[✓ Yes] No	
Diabetic:	Yes [✓ No]	
Diagnosis of COPD:	[✓ Yes] No	
HF diagnosed within the last 18 months:	Yes [✓ No]	
Receives beta blockers:	[✓ Yes] No	
Receives ACEI/ARB:	[✓ Yes] No	
Age:	< 55	
Ejection fraction (%):	< 20	11:27 15/02
SBP:	< 110	Get from Device
BMI:	20 - 24	11:27 15/02

Αποθήκευση στιγμιοτύπου οθόνης...

HF Risk

Diagnosis of COPD:	[✓ Yes] No	
HF diagnosed within the last 18 months:	Yes [✓ No]	
Receives beta blockers:	[✓ Yes] No	
Receives ACEI/ARB:	[✓ Yes] No	
Age:	< 55	
Ejection fraction (%):	< 20	11:27 15/02
SBP:	< 110	Get from Device
BMI:	20 - 24	11:27 15/02
Creatinine:	90 - 109	
NYHA Class:	1	

Calculate

HF Risk

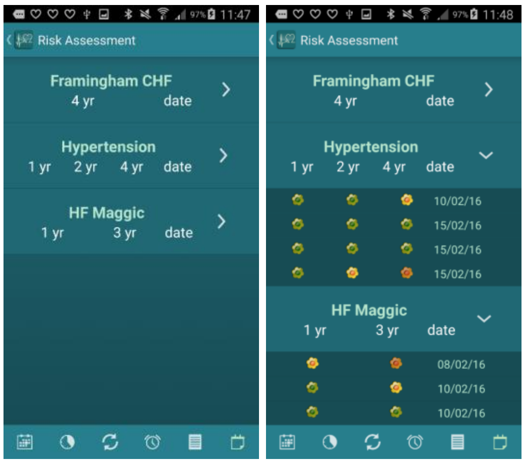
Your probability of Heart Failure is:

MINIMUM within 1 year

SLIGHT within 3 years

1 year: 12,2%, 3 years: 29,2%
(demonstration purposes)

Figure 4-30: MAGGIC HF


Risk Assessment

Framingham CHF
4 yr date

Hypertension
1 yr 2 yr 4 yr date

HF Maggic
1 yr 3 yr date

Risk Assessment

Framingham CHF
4 yr date

Hypertension
1 yr 2 yr 4 yr date
10/02/16
15/02/16
15/02/16
15/02/16

HF Maggic
1 yr 3 yr date
08/02/16
10/02/16
10/02/16

Figure 4-31: List of calculated risk scores

4.9.2.1 Technical Details

CHF Alarm is a native mobile application for Android operating system using android sdk versions 14-23. The application was developed in Eclipse for Android. It stores data locally in an SQLite database. Regarding features that the application needs access to, these include *internet*, *network state* and *bluetooth*. In order to obtain authorization to MyHealthAvatar so as to synchronize data, CHF Alarm uses OAuth 2.0 protocol.

4.9.2.2 Platform integration (through the MHA API)

CHF mobile application communicates with MHA through specific published API calls

“*CHF Real-time patient monitoring”* service sending data

- Send a list of alerts
  - https://myhealthavatar.org/mha/api/v2/user/general_alert

“*CHF Risk Assessment”* service receiving data

- Get measurements, such as bmi and left ventricular hypertrophy on ECG
  - https://myhealthavatar.org/mha/api/v3/measurements/latest
- Get general health data, such as diabetic, current smoker and parental hypertension
  - https://myhealthavatar.org/mha/api/v3/user/general/latest
- Get profile data, such as age and gender
  - https://myhealthavatar.org/mha/api/v3/profile/full
- Get clinical data, such as valve disease, diagnosis of chronic obstructive pulmonary disease, heart failure diagnosed within the last 18 months, receives beta blockers and receives ACEI/ARB

- https://myhealthavatar.org/mha/api/chf

“*CHF Risk Assessment”* service sending data

- Send a list of risk scores
    - https://myhealthavatar.org/mha/api/v2/user/general_risk

4.9.2.3 Evaluation Results

This use case scenario was evaluated from 22 volunteers. The evaluation procedure is described in D9.3 deliverable. The following table present the results from the evaluation reports summarizing the mean score values for each individual question. In general we see that the users believe that the platform has very good functionality and can efficiently respond to all tasks utilizing all necessary resources without harming them. According to the users evaluation, the system is able to reliably and efficiently share information with MHA (process of syncing is transparent and secure) and has a friendly and usable interface since users found easy to use it with not much effort. They found easy the way there were able to navigate from one screen to the other and monitor alerts. Security was of great concern and portability as expected wasn’t an issue since this is a web based application. Lastly most of the users reported that the software can deliver the intended goals and be used without harming people. They feel this tool allows them to be able to use validated models for risk assessment both in real time (short term) and in long term. One comment (stated by many) was the possibility of porting (and creating) similar applications to other types of mobile operating systems (like Apple in iPhone iOS).

Question	Mean Value
Functionality	
Can software perform the tasks required?	4,7
Can software perform "Real-time monitoring and recording of patient's vital signs" (i.e. heart rate and oxygen saturation)?	4,5
Can software perform the detection of abnormal measurements?	4,5
Can software perform the provision of risk assessment models for self-health status assessment?	4,5
Is the result as expected?	4,4
Does the application records, detects and displays alert events?	4,6
Does the application notify the user every time an alert event occurs?	4,7
Does the application calculates and displays risk probabilities based on the models?	4,7
Can the system interact with MyHealthAvatar platform, sending and receiving data?	4,6
Does the application uses OAuth 2.0 protocol to access MyHealthAvatar APIs?	4,7
Efficiency	
How quickly does the system respond?	4,7
Does the system utilize resources efficiently?	4,7

Compatibility	
Can the system share resources without loss of its functionality?	4,6
Can the system share information/data with other MyHealthAvatar components?	4,5
Usability	
Does the user comprehend how to use the system easily?	4,7
Is the navigation through pages straightforward and smooth?	4,6
Is it straightforward to install the app on a supported system?	4,7
Can the user learn to use the system easily?	4,6
Can the user use the system without much effort?	4,7
Does the interface look good?	4,7
Reliability	
Have most of the faults in the software been eliminated over time?	4,3
Is the software capable of handling errors?	4,5
Can the software resume working & restore lost data after failure?	4,3
Security	
Does the system provide identification access wherever is needed?	4,5
Are data accessible only to authorized users?	4,7
Can the system trace actions uniquely?	4,5
Does the system prevent unauthorized access?	4,4
Maintainability	
Can faults be easily diagnosed?	4,6
Can the software be easily modified?	4,4
Can the software continue functioning if changes are made?	4,4
Can the software be tested easily?	4,5
Portability	
Can the software be moved to other environments?	4,3
Can the software be installed easily?	4,4
Does the software comply with portability standards?	4,4
Can the software easily replace other software?	4,4
Quality in Use	
How accurate and complete is the software for the intended use?	4,6
Does the software improve the time or reduce resources for the intended goal?	4,5
Does the software satisfy the perceived achievements of pragmatic goals?	4,6
Can the software harm people in the intended contexts of use?	1,5

The figures below shows the opinion of the users in terms of the quality metrics presented in section 3.1.4 of this deliverable. As it is shown the application was able to deliver its goals according to the user's assessments. The users were concerned with the interaction of MHA but they found that the system can perform the described tasks in good quality of use and the general acceptance, on all of the different metrics, was high.

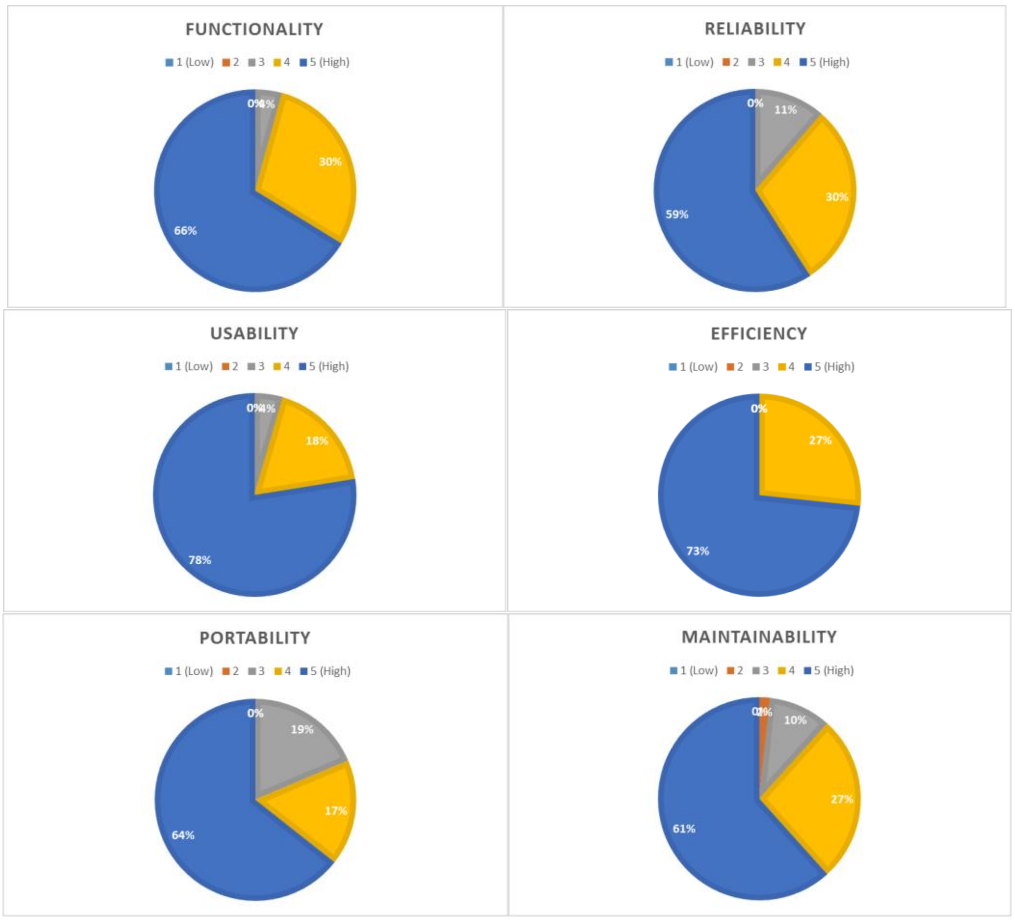


Figure 4-32: Evaluation scores CHF use case

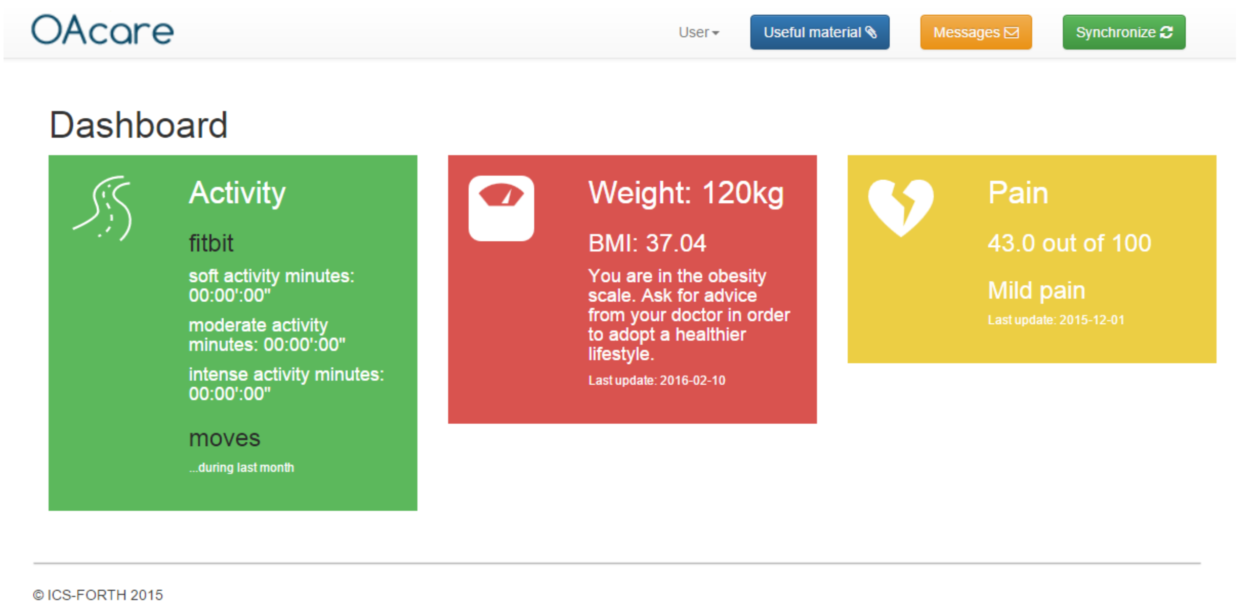
4.9.3 Osteoathritis (oaCARE, oaCARE+)

Osteoarthritis is the most common form of arthritis, affecting millions of people worldwide. It is a degenerative condition of joints and is characterized by loss of the articular cartilage that acts as a protective cushion between bones within a joint and by growth of a new bone in affected joints, causing stiffness and pain. Osteoarthritis affects mainly the knee, hip, hand, spine and less often, the feet. Its symptoms often develop slowly and worsen over time. Osteoarthritis usually affects more women than men, and tends to turn up as people get older but is also common amongst people of working age. Other common factors that may increase the risk of developing osteoarthritis are obesity, previous joint injuries, certain occupations, genetics, bone deformities and other diseases.

A patient visits Primary Care or GP complaining for knee problems and knee pain symptoms. The clinician proceeds with the diagnosis of the osteoarthritis condition by physical exam in conjunction with imaging test (radiographs and MRIs) and lab tests (blood tests and joint fluid analysis). As there is no known cure for osteoarthritis, clinician advices the patient for lifestyle interventions i.e., mild daily exercise, reduction of body weight and proper medication for reducing the pain and improving the patient’s overall condition. If these conservative treatments don't help, other procedures may be applied (e.g. surgical, lubrication injections etc.).

However, these healthy behaviors are difficult to be achieved in practice despite the fact that their value is understood by patients. Moreover, medical professionals cannot usually ascertain if the patients follow their guidelines for a healthier lifestyle, which would be helpful for better follow-up. The OAcare app was designed for empowering both clinicians and patients for the long-term management of the knee osteoarthritis condition utilizing the functionalities of the MyHealthAvatar platform. The design of the app is responsive i.e., it can be seamlessly displayed in different screen resolutions from smartphones to tablets and personal computers. The OAcare app delivers two different versions satisfying the patients and clinicians requirements and needs.

4.9.3.1 oaCARE for patients



Dashboard

Activity
fitbit
soft activity minutes: 00:00':00"
moderate activity minutes: 00:00':00"
intense activity minutes: 00:00':00"
moves
...during last month

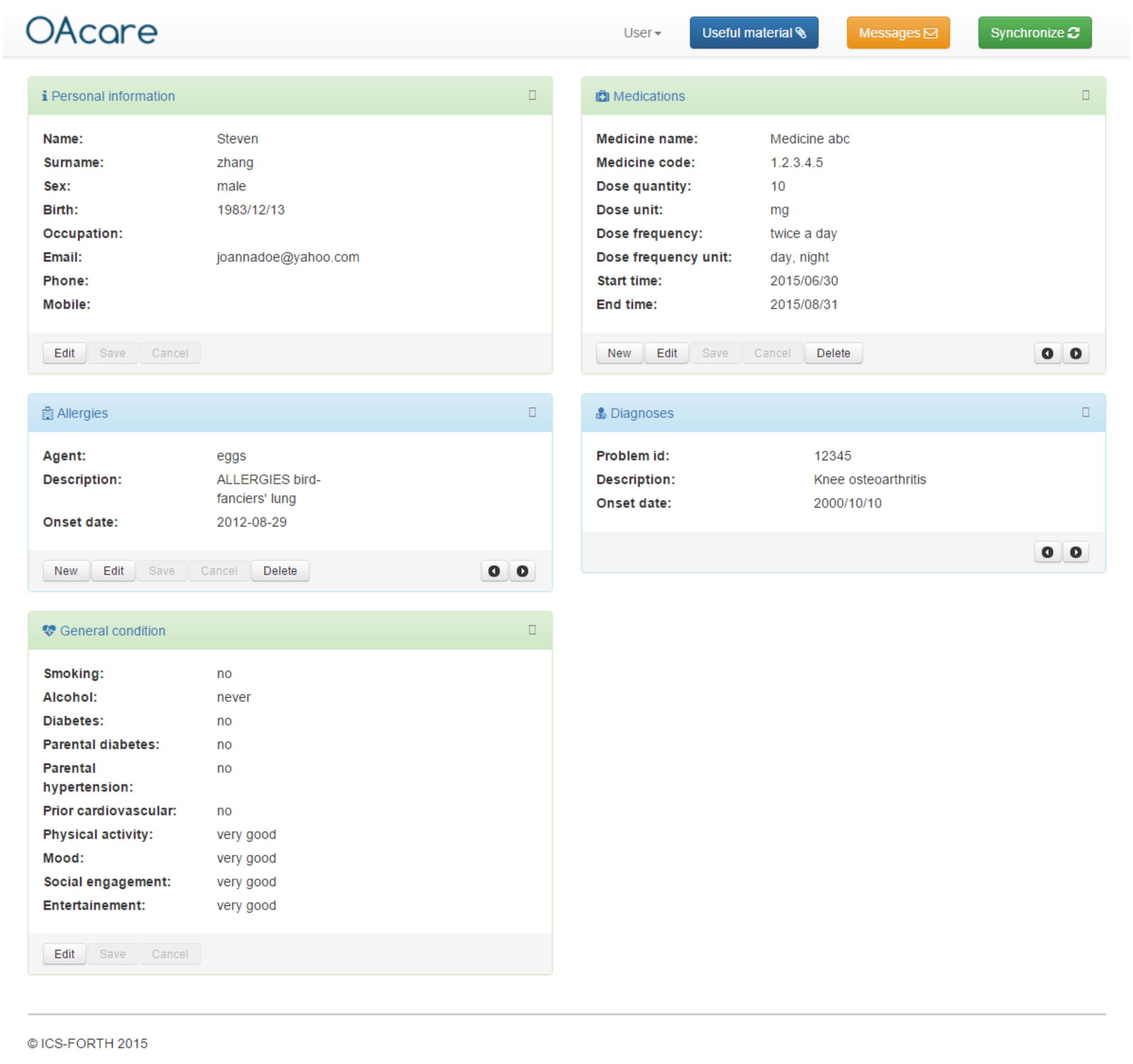
Weight: 120kg
BMI: 37.04
You are in the obesity scale. Ask for advice from your doctor in order to adopt a healthier lifestyle.
Last update: 2016-02-10

Pain
43.0 out of 100
Mild pain
Last update: 2015-12-01

© ICS-FORTH 2015

Figure 4-33: Through the dashboard of the OAcare app, the patient has an overview of his activity, weight and pain data.

The patients’ version offers a supportive environment for empowering patients in looking after their own health, raising their self-awareness of the osteoarthritis risk factors while encouraging for a healthier lifestyle. The app presents (Figure 5-33) and utilizes activity, weight and pain data and is able to advise the patient properly if he does not manage to meet the special medical guidelines issued by the care plan that was previously set up by his physician. Through the app, the patient can also update his profile data, such as allergies, medications, weight (Figure 5-34) and fill out questionnaires for extracting pain and quality of life (QoL) information (Figure 5-35).



Personal information

Name:	Steven
Surname:	zhang
Sex:	male
Birth:	1983/12/13
Occupation:	
Email:	joannadoe@yahoo.com
Phone:	
Mobile:	

Medications

Medicine name:	Medicine abc
Medicine code:	1.2.3.4.5
Dose quantity:	10
Dose unit:	mg
Dose frequency:	twice a day
Dose frequency unit:	day, night
Start time:	2015/06/30
End time:	2015/08/31

Allergies

Agent:	eggs
Description:	ALLERGIES bird-fanciers' lung
Onset date:	2012-08-29

Diagnoses

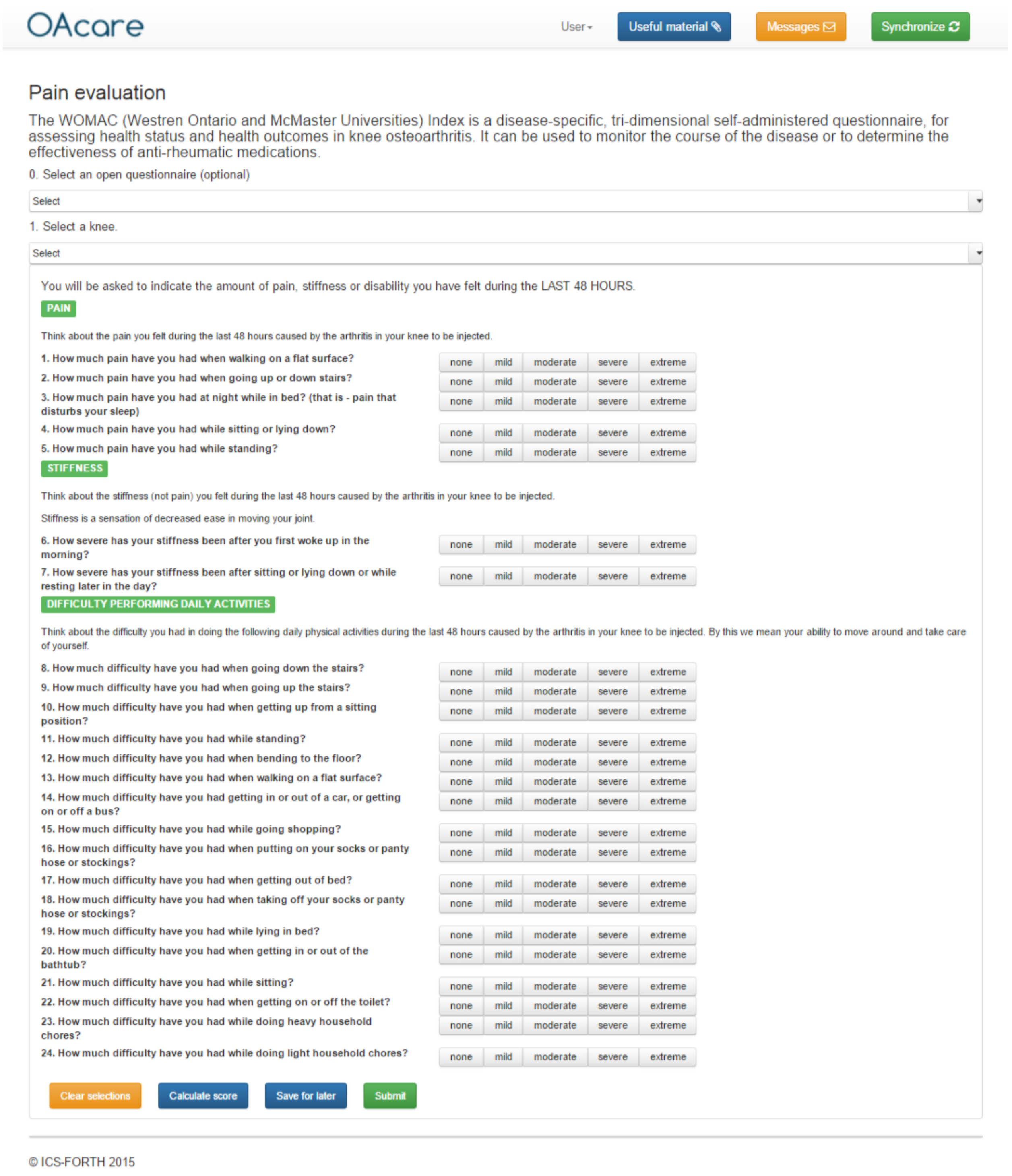
Problem id:	12345
Description:	Knee osteoarthritis
Onset date:	2000/10/10

General condition

Smoking:	no
Alcohol:	never
Diabetes:	no
Parental diabetes:	no
Parental hypertension:	no
Prior cardiovascular:	no
Physical activity:	very good
Mood:	very good
Social engagement:	very good
Entertainement:	very good

© ICS-FORTH 2015

Figure 4-34: Patient's profile data



OAcare

User ▾ | Useful material | Messages | Synchronize

Pain evaluation

The WOMAC (Westren Ontario and McMaster Universities) Index is a disease-specific, tri-dimensional self-administered questionnaire, for assessing health status and health outcomes in knee osteoarthritis. It can be used to monitor the course of the disease or to determine the effectiveness of anti-rheumatic medications.

0. Select an open questionnaire (optional)

Select

1. Select a knee.

Select

You will be asked to indicate the amount of pain, stiffness or disability you have felt during the LAST 48 HOURS.

PAIN

Think about the pain you felt during the last 48 hours caused by the arthritis in your knee to be injected.

1. How much pain have you had when walking on a flat surface?	none	mild	moderate	severe	extreme
2. How much pain have you had when going up or down stairs?	none	mild	moderate	severe	extreme
3. How much pain have you had at night while in bed? (that is - pain that disturbs your sleep)	none	mild	moderate	severe	extreme
4. How much pain have you had while sitting or lying down?	none	mild	moderate	severe	extreme
5. How much pain have you had while standing?	none	mild	moderate	severe	extreme

STIFFNESS

Think about the stiffness (not pain) you felt during the last 48 hours caused by the arthritis in your knee to be injected.

Stiffness is a sensation of decreased ease in moving your joint.

6. How severe has your stiffness been after you first woke up in the morning?	none	mild	moderate	severe	extreme
7. How severe has your stiffness been after sitting or lying down or while resting later in the day?	none	mild	moderate	severe	extreme

DIFFICULTY PERFORMING DAILY ACTIVITIES

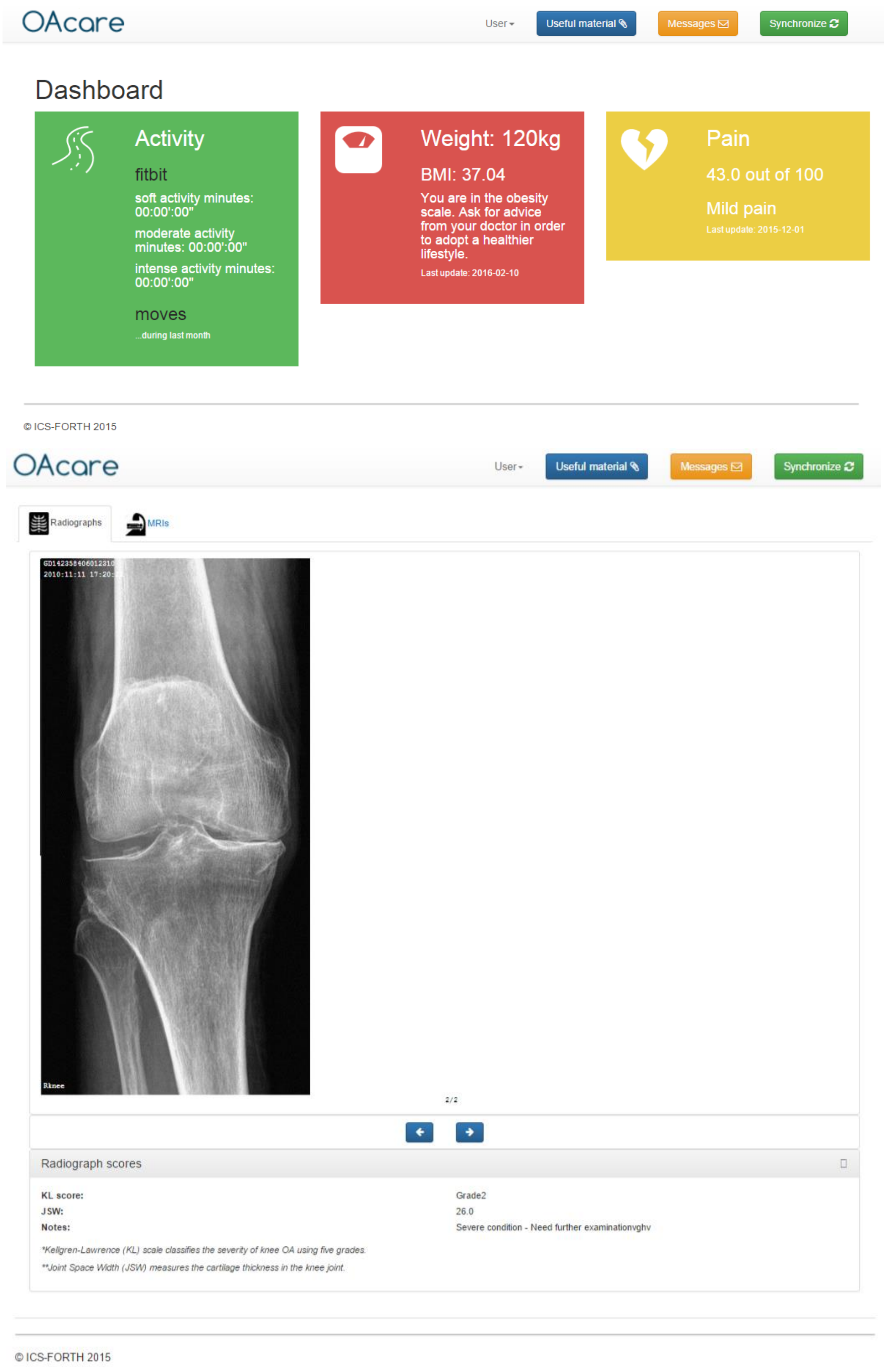
Think about the difficulty you had in doing the following daily physical activities during the last 48 hours caused by the arthritis in your knee to be injected. By this we mean your ability to move around and take care of yourself.

8. How much difficulty have you had when going down the stairs?	none	mild	moderate	severe	extreme
9. How much difficulty have you had when going up the stairs?	none	mild	moderate	severe	extreme
10. How much difficulty have you had when getting up from a sitting position?	none	mild	moderate	severe	extreme
11. How much difficulty have you had while standing?	none	mild	moderate	severe	extreme
12. How much difficulty have you had when bending to the floor?	none	mild	moderate	severe	extreme
13. How much difficulty have you had when walking on a flat surface?	none	mild	moderate	severe	extreme
14. How much difficulty have you had getting in or out of a car, or getting on or off a bus?	none	mild	moderate	severe	extreme
15. How much difficulty have you had while going shopping?	none	mild	moderate	severe	extreme
16. How much difficulty have you had when putting on your socks or panty hose or stockings?	none	mild	moderate	severe	extreme
17. How much difficulty have you had when getting out of bed?	none	mild	moderate	severe	extreme
18. How much difficulty have you had when taking off your socks or panty hose or stockings?	none	mild	moderate	severe	extreme
19. How much difficulty have you had while lying in bed?	none	mild	moderate	severe	extreme
20. How much difficulty have you had when getting in or out of the bathtub?	none	mild	moderate	severe	extreme
21. How much difficulty have you had while sitting?	none	mild	moderate	severe	extreme
22. How much difficulty have you had when getting on or off the toilet?	none	mild	moderate	severe	extreme
23. How much difficulty have you had while doing heavy household chores?	none	mild	moderate	severe	extreme
24. How much difficulty have you had while doing light household chores?	none	mild	moderate	severe	extreme

Clear selections | Calculate score | Save for later | Submit

© ICS-FORTH 2015

Figure 4-35: Pain evaluation questionnaire



OAcare

User ▾ | Useful material | Messages | Synchronize

Dashboard

Activity

fitbit

soft activity minutes: 00:00':00"

moderate activity minutes: 00:00':00"

intense activity minutes: 00:00':00"

moves

...during last month

Weight: 120kg

BMI: 37.04

You are in the obesity scale. Ask for advice from your doctor in order to adopt a healthier lifestyle.

Last update: 2016-02-10

Pain

43.0 out of 100

Mild pain

Last update: 2015-12-01

© ICS-FORTH 2015

OAcare

User ▾ | Useful material | Messages | Synchronize

Radiographs | MRIs

2 / 2

Radiograph scores

KL score:	Grade2
JSW:	26.0
Notes:	Severe condition - Need further examinationvghv

**Kellgren-Lawrence (KL) scale classifies the severity of knee OA using five grades.*

***Joint Space Width (JSW) measures the cartilage thickness in the knee joint.*

© ICS-FORTH 2015

Figure 4-36: Patient's imaging data and radiographic scores

Furthermore, the patient can view his imaging data (ragiographs and MRIs), the radiographic scores entered by the clinician after examining the radiographic data (Figure 5-36), and some useful charts regarding the variances of activity, weight and pain data over the time (Figure 5-37). In addition, the patient can view his current care plan that is set up by the physician (Figure 5-38), as well as view and update his weight and height (Figure 5-39).

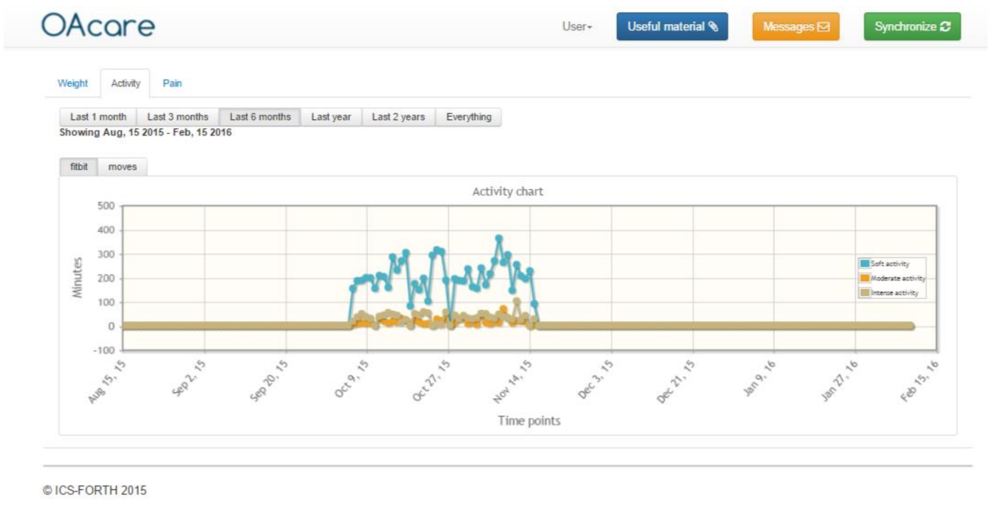
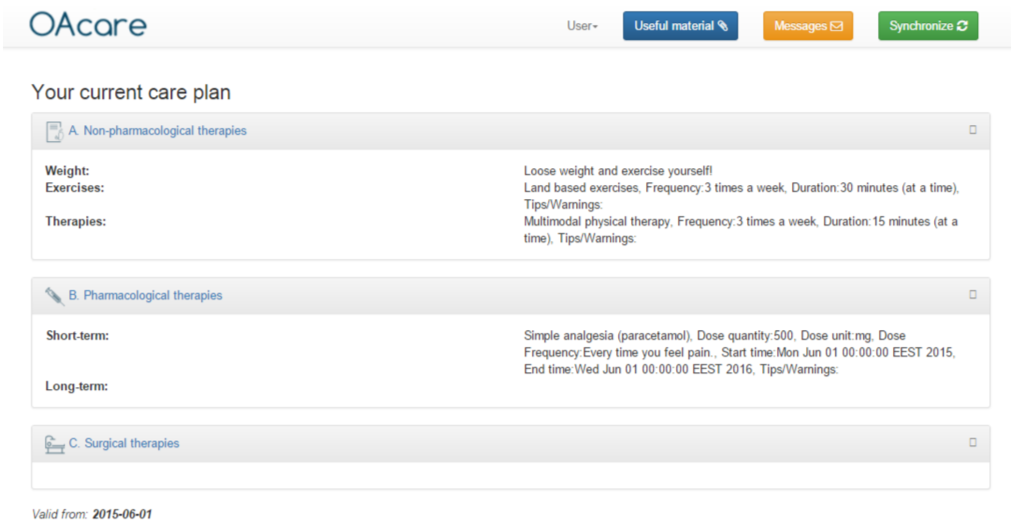


Figure 4-37: Activity, weight and pain data charts



Your current care plan

A. Non-pharmacological therapies

Weight:	Loose weight and exercise yourself!
Exercises:	Land based exercises, Frequency:3 times a week, Duration:30 minutes (at a time), Tips/Warnings:
Therapies:	Multimodal physical therapy, Frequency:3 times a week, Duration:15 minutes (at a time), Tips/Warnings:

B. Pharmacological therapies

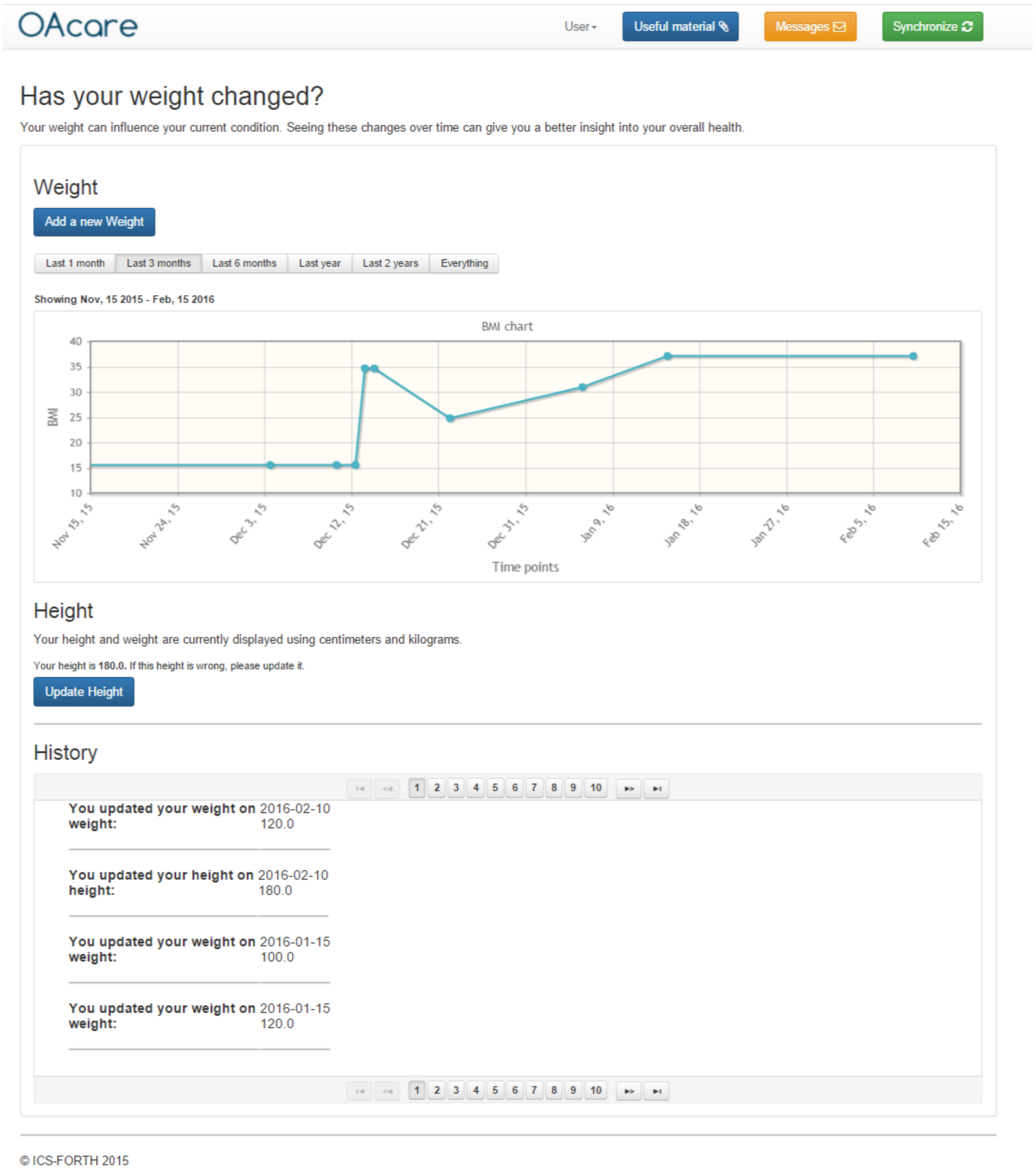
Short-term:	Simple analgesia (paracetamol), Dose quantity:500, Dose unit:mg, Dose Frequency:Every time you feel pain., Start time:Mon Jun 01 00:00:00 EEST 2015, End time:Wed Jun 01 00:00:00 EEST 2016, Tips/Warnings:
Long-term:	

C. Surgical therapies

Valid from: ***2015-06-01***

Figure 4-38: Patient's care plan

The patient can also use the app in order to directly communicate with his physician (Figure 15). Moreover, the patient can search over the educational material for the osteoarthritis condition and may be informed about the nature of the condition, the symptoms, the causes, the risk factors, the treatments and drugs, the lifestyle remedies etc (Figure 16). It is expected that a good knowledge of the condition will lead to enhanced patient behavior, allowing him to play a key role in managing his own health. Finally, OAcare app provides synchronization with MyHealthAvatar platform retrieving data from the patient's account at his request (Figure 5-41).



Has your weight changed?

Your weight can influence your current condition. Seeing these changes over time can give you a better insight into your overall health.

Weight

Showing Nov, 15 2015 - Feb, 15 2016

Height

Your height and weight are currently displayed using centimeters and kilograms.

Your height is 180.0. If this height is wrong, please update it.

History

You updated your weight on 2016-02-10
weight: 120.0

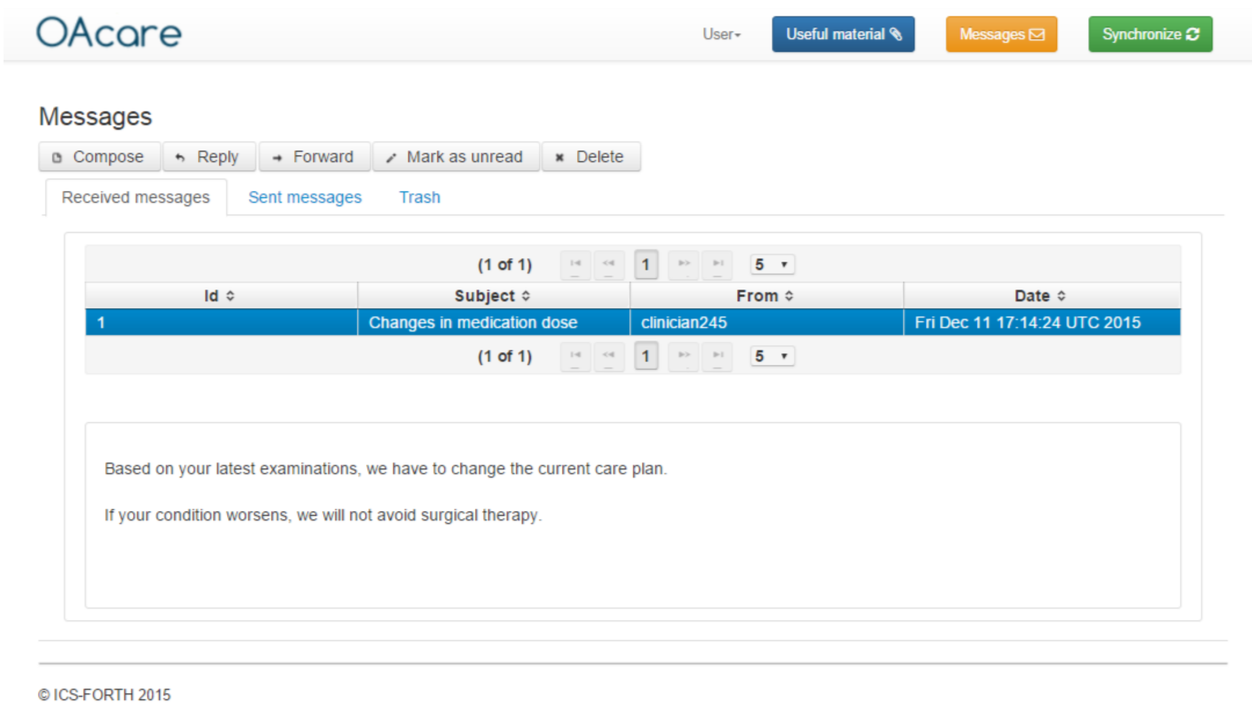
You updated your height on 2016-02-10
height: 180.0

You updated your weight on 2016-01-15
weight: 100.0

You updated your weight on 2016-01-15
weight: 120.0

© ICS-FORTH 2015

Figure 4-39: Update weight and height



Messages

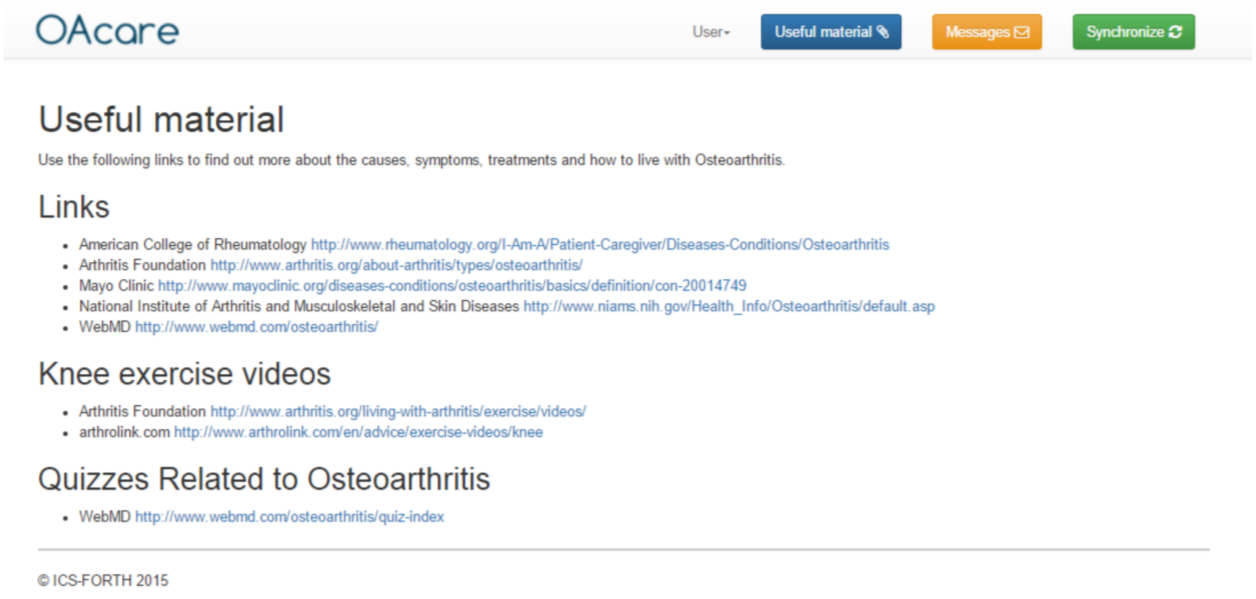
Id	Subject	From	Date
1	Changes in medication dose	clinician245	Fri Dec 11 17:14:24 UTC 2015

Based on your latest examinations, we have to change the current care plan.

If your condition worsens, we will not avoid surgical therapy.

© ICS-FORTH 2015

Figure 4-40: Patient can exchange messages with his physician directly



Useful material

Use the following links to find out more about the causes, symptoms, treatments and how to live with Osteoarthritis.

Links

- American College of Rheumatology http://www.rheumatology.org/I-Am-A/Patient-Caregiver/Diseases-Conditions/Osteoarthritis
- Arthritis Foundation http://www.arthritis.org/about-arthritis/types/osteoarthritis/
- Mayo Clinic http://www.mayoclinic.org/diseases-conditions/osteoarthritis/basics/definition/con-20014749
- National Institute of Arthritis and Musculoskeletal and Skin Diseases http://www.niams.nih.gov/Health_Info/Osteoarthritis/default.asp
- WebMD http://www.webmd.com/osteoarthritis/

Knee exercise videos

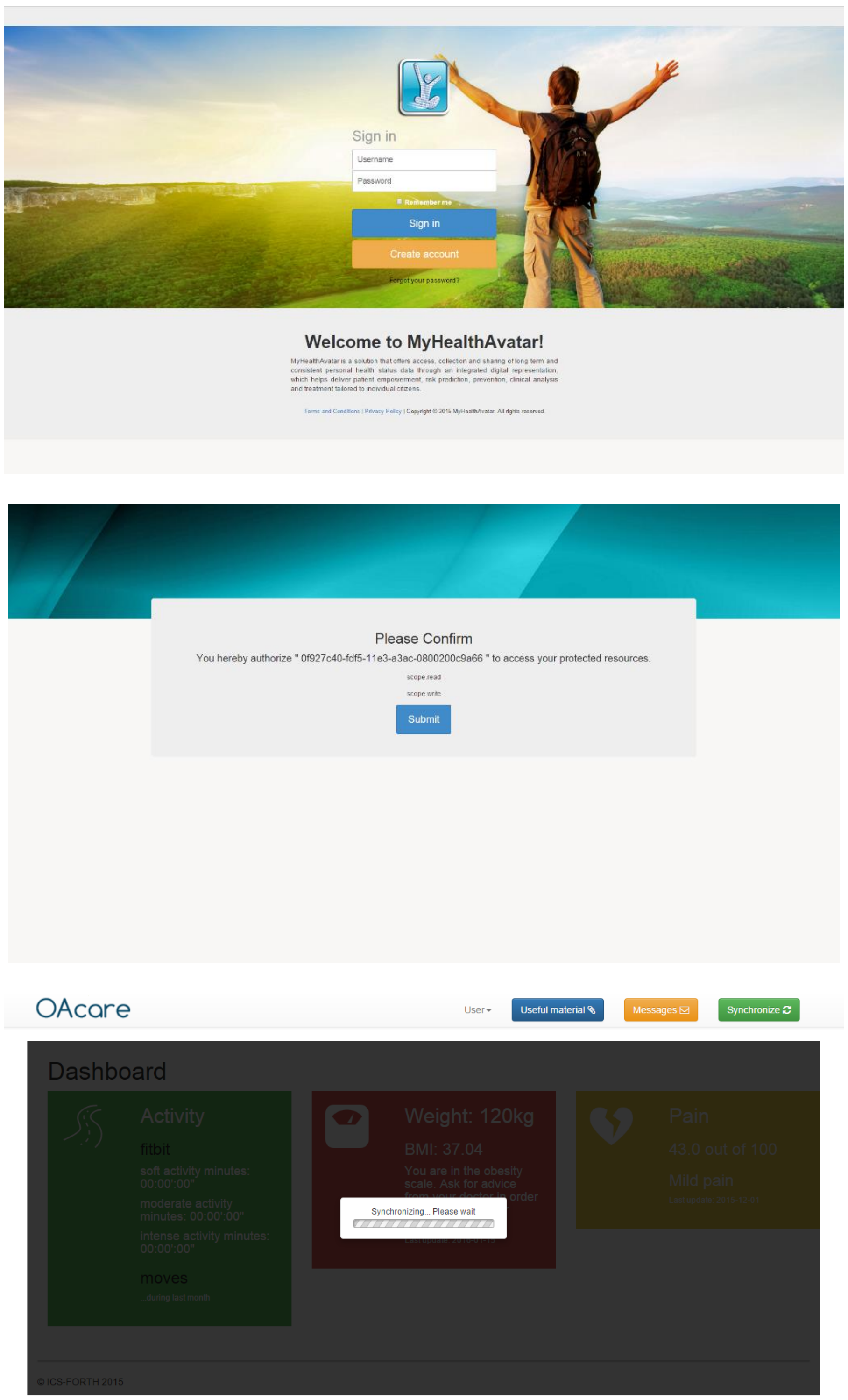
- Arthritis Foundation http://www.arthritis.org/living-with-arthritis/exercise/videos/
- arthrolink.com http://www.arthrolink.com/en/advice/exercise-videos/knee

Quizzes Related to Osteoarthritis

- WebMD http://www.webmd.com/osteoarthritis/quiz-index

© ICS-FORTH 2015

Figure 4-41: Educational material for the osteoarthritis condition



Welcome to MyHealthAvatar!

MyHealthAvatar is a solution that offers access, collection and sharing of long term and consistent personal health status data through an integrated digital representation, which helps deliver patient empowerment, risk prediction, prevention, clinical analysis and treatment tailored to individual citizens.

Terms and Conditions | Privacy Policy | Copyright © 2015 MyHealthAvatar. All rights reserved.

Please Confirm

You hereby authorize " 0f927c40-fdf5-11e3-a3ac-0800200c9a66 " to access your protected resources.

scope:read

scope:write

Dashboard

Activity

fitbit

soft activity minutes: 00:00':00"

moderate activity minutes: 00:00':00"

intense activity minutes: 00:00':00"

moves

...during last month

Weight: 120kg

BMI: 37.04

You are in the obesity scale. Ask for advice from your doctor in order

Last update: 2016-01-15

Pain

43.0 out of 100

Mild pain

Last update: 2015-12-01

Synchronizing... Please wait

© ICS-FORTH 2015

Figure 4-42: Synchronizing data with MyHealthAvatar platform

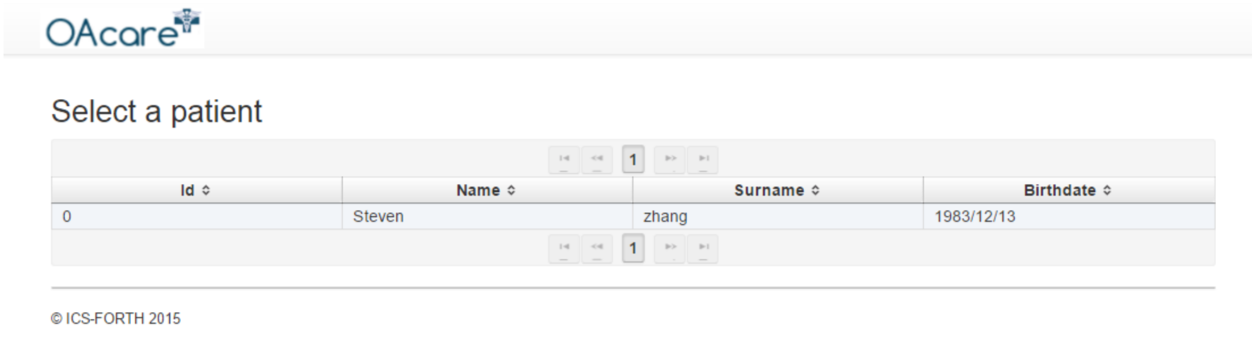
4.9.3.2 oaCARE for the clinicians (OAcare+)

The clinicians' view has been developed in order to provide a useful input to clinicians regarding the current patient's health status, as the related data will be properly visualized and presented.

At first, the clinician selects a patient to view their data (Figure 18) and then the app provides the main menu for easy navigation (Figure 5-43). The clinician can view the patient's imaging data (radiographs and MRIs) and up-date or edit the two radiographic scoring metrics (Figure 5-44). Moreover, the clinician can examine patient's lifestyle and pain data (Figure 5-45), as well as their profile (Figure 5-46) and update or send advice messages to patients (Figure 5-47).

Furthermore the clinician can set up a new care plan (Figure 5-48,5-49) and view the history of previously set up plans and their level of "success" based on the corresponding pain data. In addition, the clinician can upload educational material (links or files) that may be helpful for patients for better understanding the nature of their condition, or for further learning about recent advances in osteoarthritis management.

To summarize, the scope of the OAcare app is to provide patients an easy-to-use way of managing and monitoring their medical data related to the knee osteoarthritis, from the emerging of the condition until today, with the goal to enhance patients' engagement. On the other side, the OAcare app will benefit clinicians as they will be able to view the patient's medical data over the time, assisting them to better understand the patients' current health status and the progression of the condition. Next releases of the app may contain a genetic evaluation service for examining if an increased risk of developing osteoarthritis exists.



Select a patient

Id	Name	Surname	Birthdate
0	Steven	zhang	1983/12/13

© ICS-FORTH 2015

Figure 4-43: List of Patients

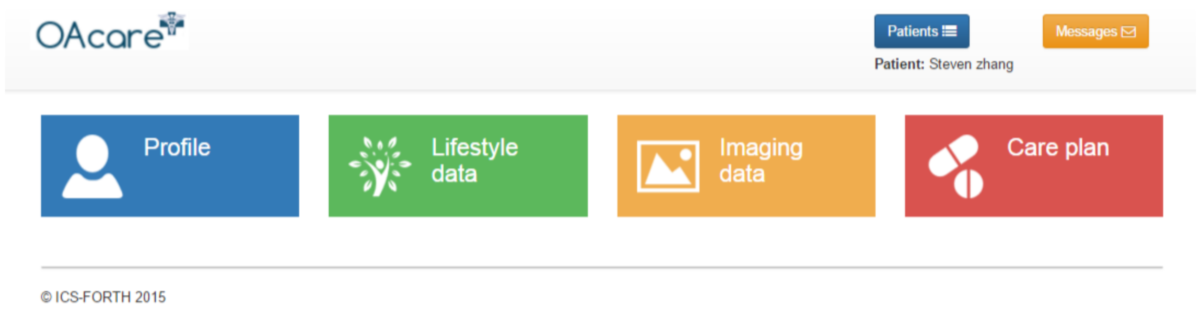


Figure 4-44: Main menu of OAcare+ app

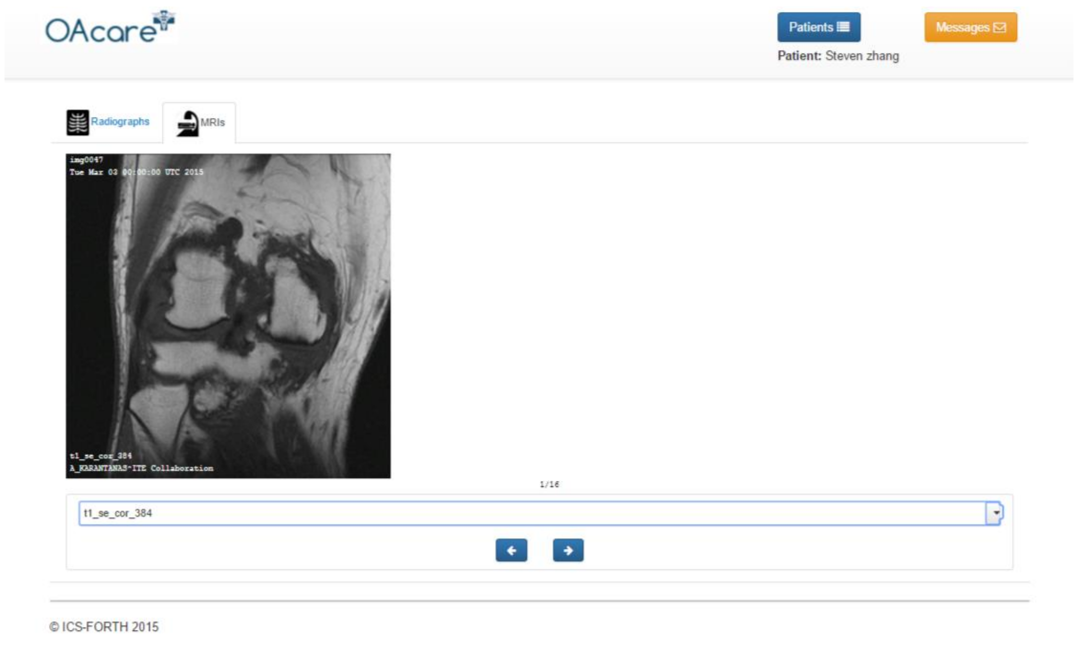


Figure 4-45: View the patient's imaging data and/or edit the two radiographic scoring metrics

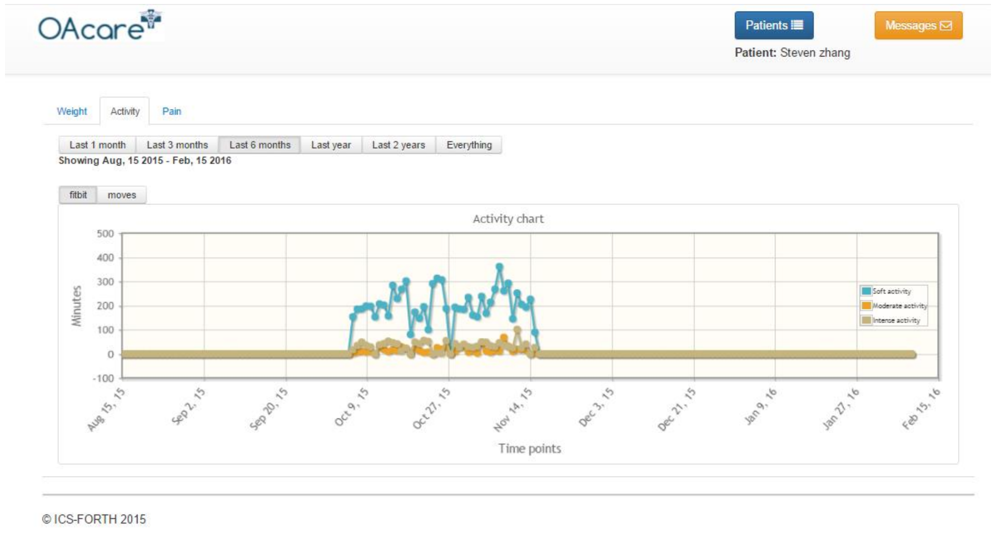
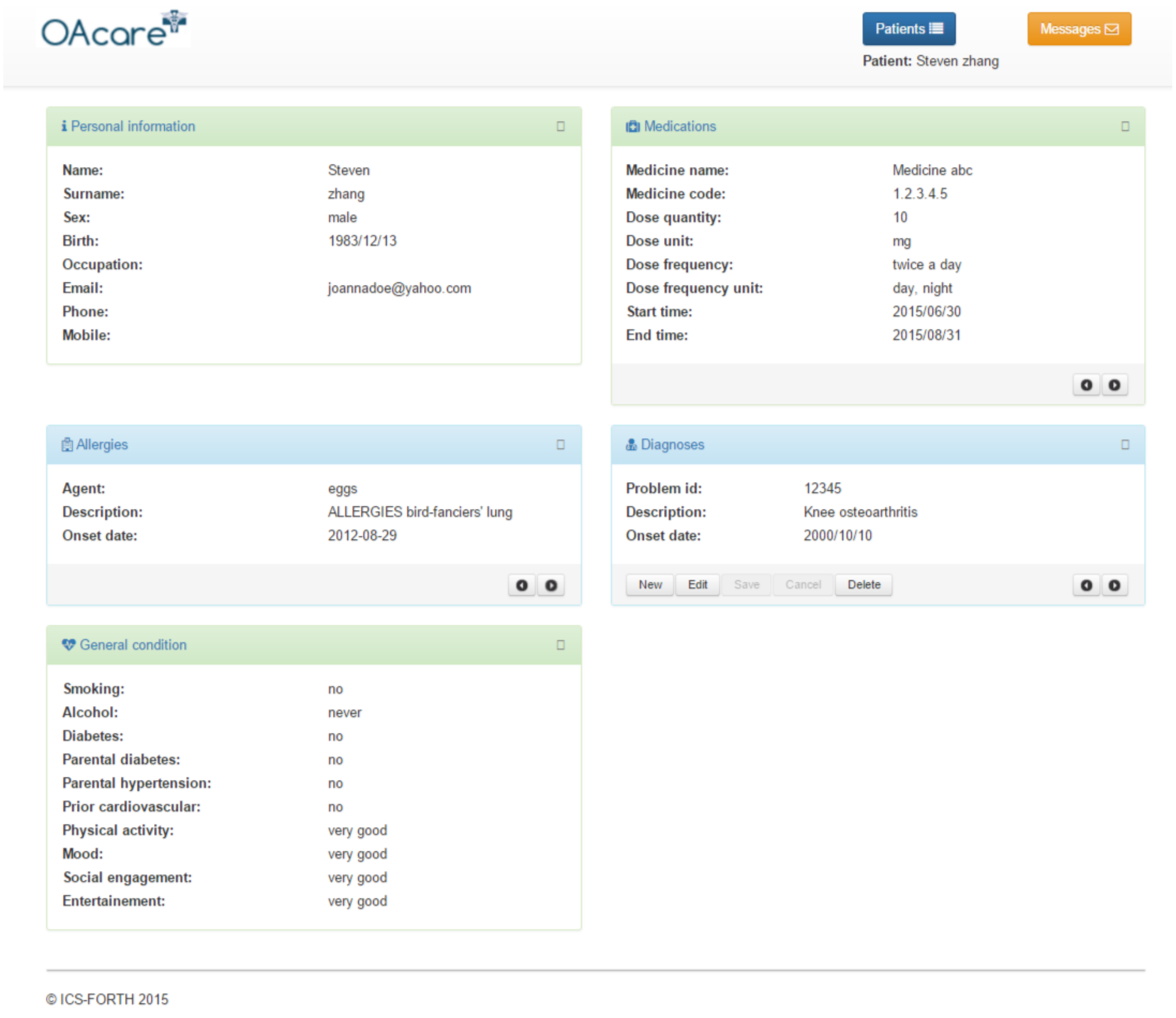


Figure 4-46: Patient's lifestyle and pain data



Patients | Messages

Patient: Steven zhang

Personal information

Name:	Steven
Surname:	zhang
Sex:	male
Birth:	1983/12/13
Occupation:	
Email:	joannadoe@yahoo.com
Phone:	
Mobile:	

Medications

Medicine name:	Medicine abc
Medicine code:	1.2.3.4.5
Dose quantity:	10
Dose unit:	mg
Dose frequency:	twice a day
Dose frequency unit:	day, night
Start time:	2015/06/30
End time:	2015/08/31

Allergies

Agent:	eggs
Description:	ALLERGIES bird-fanciers' lung
Onset date:	2012-08-29

Diagnoses

Problem id:	12345
Description:	Knee osteoarthritis
Onset date:	2000/10/10

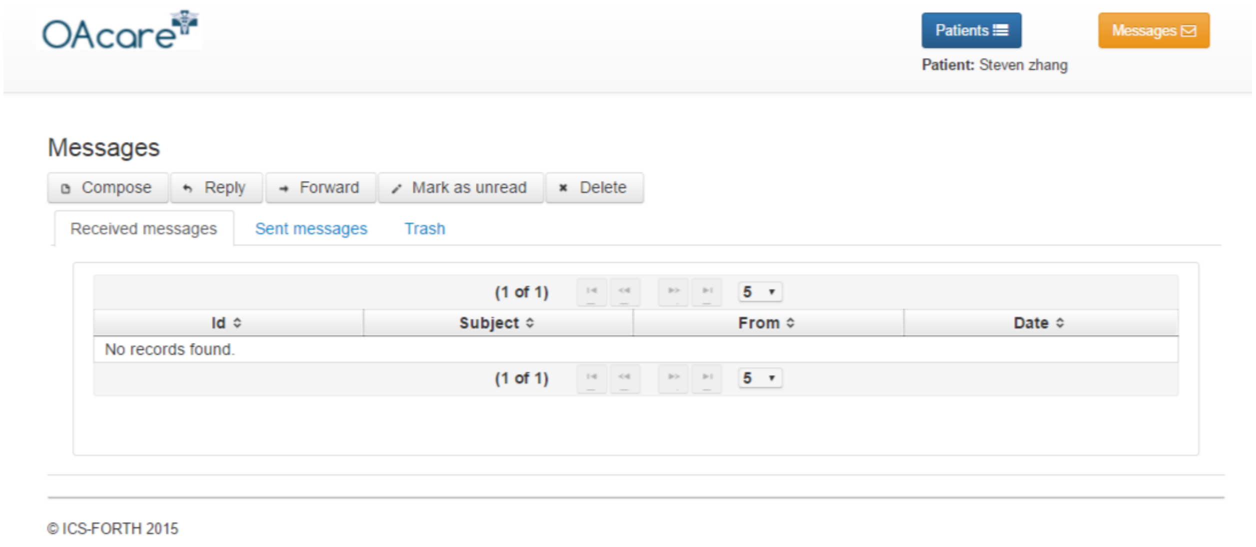
New | Edit | Save | Cancel | Delete

General condition

Smoking:	no
Alcohol:	never
Diabetes:	no
Parental diabetes:	no
Parental hypertension:	no
Prior cardiovascular:	no
Physical activity:	very good
Mood:	very good
Social engagement:	very good
Entertainement:	very good

© ICS-FORTH 2015

Figure 4-47: Patient's profile data



Patients | Messages

Patient: Steven zhang

Messages

Compose | Reply | Forward | Mark as unread | Delete

Received messages | Sent messages | Trash

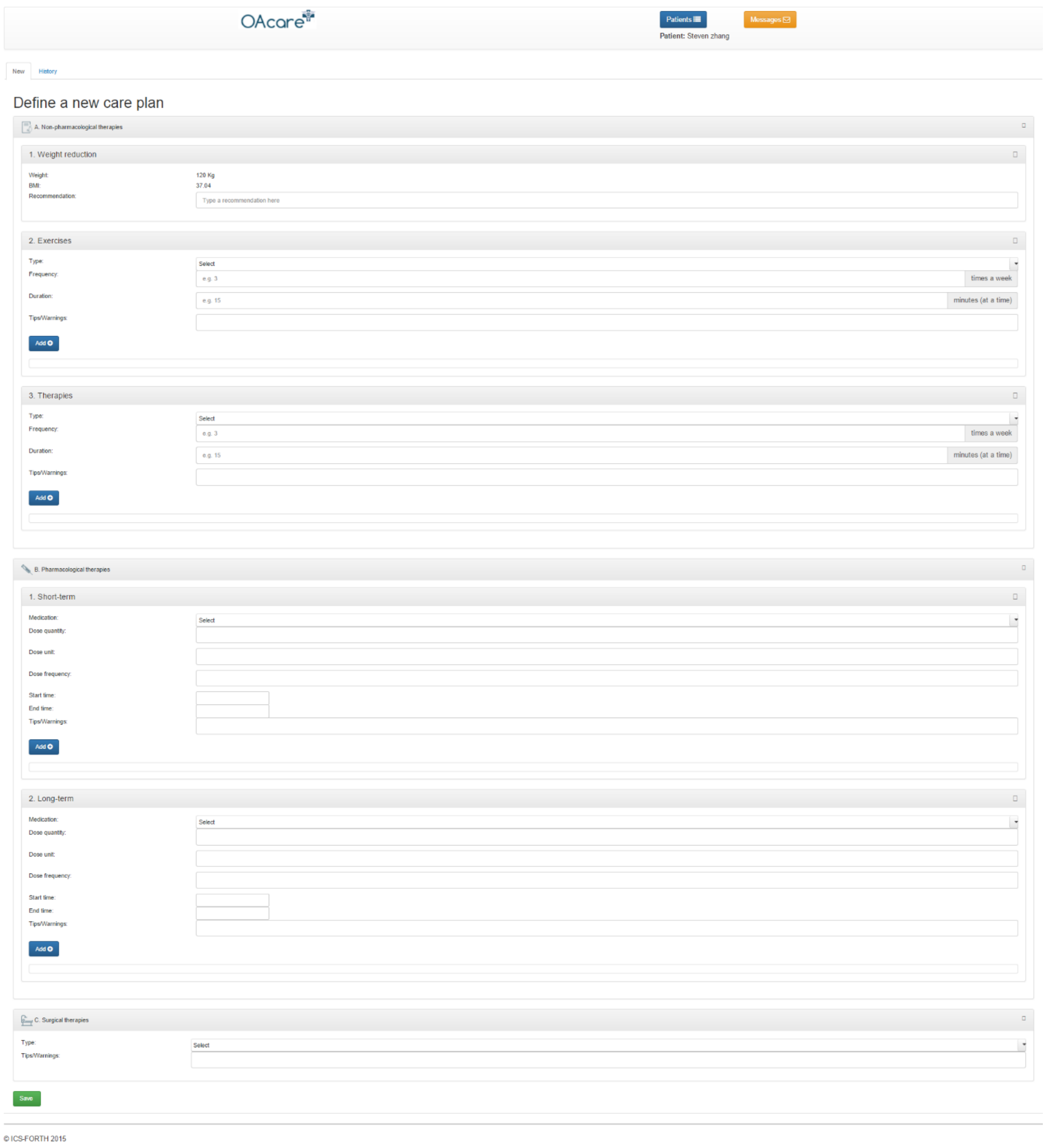
(1 of 1) 5

Id	Subject	From	Date
No records found.			

(1 of 1) 5

© ICS-FORTH 2015

Figure 4-48: Sending advice messages to patients



Patient: Steven zhang

New | History

Define a new care plan

A. Non-pharmacological therapies

1. Weight reduction

Weight:	120 Kg
BMI:	37.04
Recommendation:	Type a recommendation here

2. Exercises

Type:	Select
Frequency:	e.g. 3 times a week
Duration:	e.g. 15 minutes (at a time)
Tips/Warnings:	

Add

3. Therapies

Type:	Select
Frequency:	e.g. 3 times a week
Duration:	e.g. 15 minutes (at a time)
Tips/Warnings:	

Add

B. Pharmacological therapies

1. Short-term

Medication:	Select
Dose quantity:	
Dose unit:	
Dose frequency:	
Start time:	
End time:	
Tips/Warnings:	

Add

2. Long-term

Medication:	Select
Dose quantity:	
Dose unit:	
Dose frequency:	
Start time:	
End time:	
Tips/Warnings:	

Add

C. Surgical therapies

Type:	Select
Tips/Warnings:	

Save

© ICS-FORTH 2015

Figure 4-49: The clinician can set up a new care plan) and view the history of previously set up plans

4.9.3.3 oaCARE, oaCARE+ integration (through the MHA API)

MHA API calls

- Get account information, such as username
  - https://myhealthavatar.org/mha/api/me

- Get (all or after date) activities data, such as source, steps, duration, distance, calories, calories_bmr, calories_out, elevation, soft_activity_minutes, moderate_activity_minutes, intense_activity_minutes, sedentary_minutes
  - https://myhealthavatar.org/mha/api/v2/user/activities/
  - https://myhealthavatar.org/mha/api/v2/user/activities?from={yyyy-mm-dd}
- Get (all or after date) measurements, such as height and weight
  - https://myhealthavatar.org/mha/api/v2/user/measurements/
  - https://myhealthavatar.org/mha/api/v2/user/measurements?from={yyyy-mm-dd}
- Get (all or after date) medications, such as medicine_name, medicine_code, dose_quantity, dose_unit, dose_frequency_value, dose_frequency_unit,start_time and end_time
  - https://myhealthavatar.org/mha/api/v2/user/medication/
  - https://myhealthavatar.org/mha/api/v2/user/medication?from={yyyy-mm-dd}
- Get (all or after date) allergies, such as agent, description, onset_date
  - https://myhealthavatar.org/mha/api/v2/user/allergy/
  - https://myhealthavatar.org/mha/api/v2/user/allergy?from={yyyy-mm-dd}
- Get (all or after date) diagnoses, such as description, problem_id, onset_time
  - https://myhealthavatar.org/mha/api/v2/user/diagnosis/
  - https://myhealthavatar.org/mha/api/v2/user/diagnosis?from={yyyy-mm-dd}
- Get (all or after date) general health data, such as smoking, alcohol, diabetes, parental_diabetes, parental_hypertension, prior_cardiovascular, entertainment, physical_activity, mood, social_engagement,
  - https://myhealthavatar.org/mha/api/v2/user/general
  - https://myhealthavatar.org/mha/api/v2/user/general?from={yyyy-mm-dd}

4.9.3.4 Technical Details

OAcare and OAcare+ are dynamic web application projects, developed in Eclipse that run on Tomcat Server 7 and store data in MySQL database. In order to obtain authorization to MyHealthAvatar so as to synchronize data, OAcare uses OAuth 2.0 protocol.

4.9.3.5 Evaluation Results

This use case scenario was evaluated from 18 in total volunteers. The evaluation procedure is described in D9.3 deliverable. Below we present the results from the evaluation reports summarizing the mean values for each individual question. In general we see that the users believe that the platform has very good functionality and can efficiently respond to all tasks utilizing all necessary resources. The system is able to reliable and efficiently share information with MHA (including imaging data) and has a friendly and usable interface since users found easy to use it with not much effort. Security of great concern and portability (as expected is not an issue since this is a web based application). Lastly most of the users say that the software can deliver the intended goals and use without harming the people.

Question	Mean Value
Functionality	
Can software perform the tasks required?	4,6
Can software perform the tasks of providing to patients an easy-to-use way of managing and monitoring their medical data related to the knee osteoarthritis?	4,7
Can software perform the tasks of providing to clinicians the required functionality to view the patient’s medical data over the time?	4,7
Is the result as expected?	4,7
Does the system present activity, weight, pain and imaging data of the patient?	4,8
Does the system allow patients to update their profile and directly communicate with their physician via messages?	4,5
Does the system allow patients to fill out questionnaires for extracting pain and quality of life information?	4,5
Does the system allow physicians to view their patients' medical data?	4,4
Does the system allow physicians to set up new care plans and view history of existing care plans?	4,3
Does the system allow physicians to send messages to their patients?	4,4
Does the system allow physicians to upload educational material?	4,1
Can the system interact with MyHealthAvatar platform, sending and receiving data?	4,4
Does the system uses OAuth 2.0 protocol to access MyHealthAvatar APIs?	4,2
Efficiency	
How quickly does the system respond?	4,5
Does the system utilize resources efficiently?	4,3
Compatibility	
Can the system share resources without loss of its functionality?	4,7
Can the system share information/data with other MyHealthAvatar components?	4,6
Usability	
Does the user comprehend how to use the system easily?	4,5
Can the user learn to use the system easily?	4,5

Can the user use the system without much effort?	4,4
Does the interface look good?	4,7
Reliability	
Have most of the faults in the software been eliminated over time?	4,1
Is the software capable of handling errors?	4,5
Can the software resume working & restore lost data after failure?	4,3
Security	
Does the system provide identification access wherever is needed?	4,5
Are data accessible only to authorized users?	4,4
Can the system trace actions uniquely?	4,5
Does the system prevent unauthorized access?	4,5
Maintainability	
Can faults be easily diagnosed?	4,2
Can the software be easily modified?	4,5
Can the software continue functioning if changes are made?	4,6
Can the software be tested easily?	4,6
Portability	
Can the software be moved to other environments?	4,6
Can the software be installed easily?	4,7
Does the software comply with portability standards?	4,6
Can the software easily replace other software?	4,3
Quality in Use	
How accurate and complete is the software for the intended use?	4,6
Does the software improve the time or reduce resources for the intended goal?	4,5
Does the software satisfy the perceived achievements of pragmatic goals?	4,5
Can the software harm people in the intended contexts of use?	1,7

The figures below shows the opinion of the users in terms of the quality metrics presented in section 3.1.4 of this deliverable. As it is shown the application was able to deliver its goals according to the user's assessments. The users were concerned with the interaction of MHA but they found that the system can perform the described tasks in good quality of use and the general acceptance, on all of the different metrics, was high.

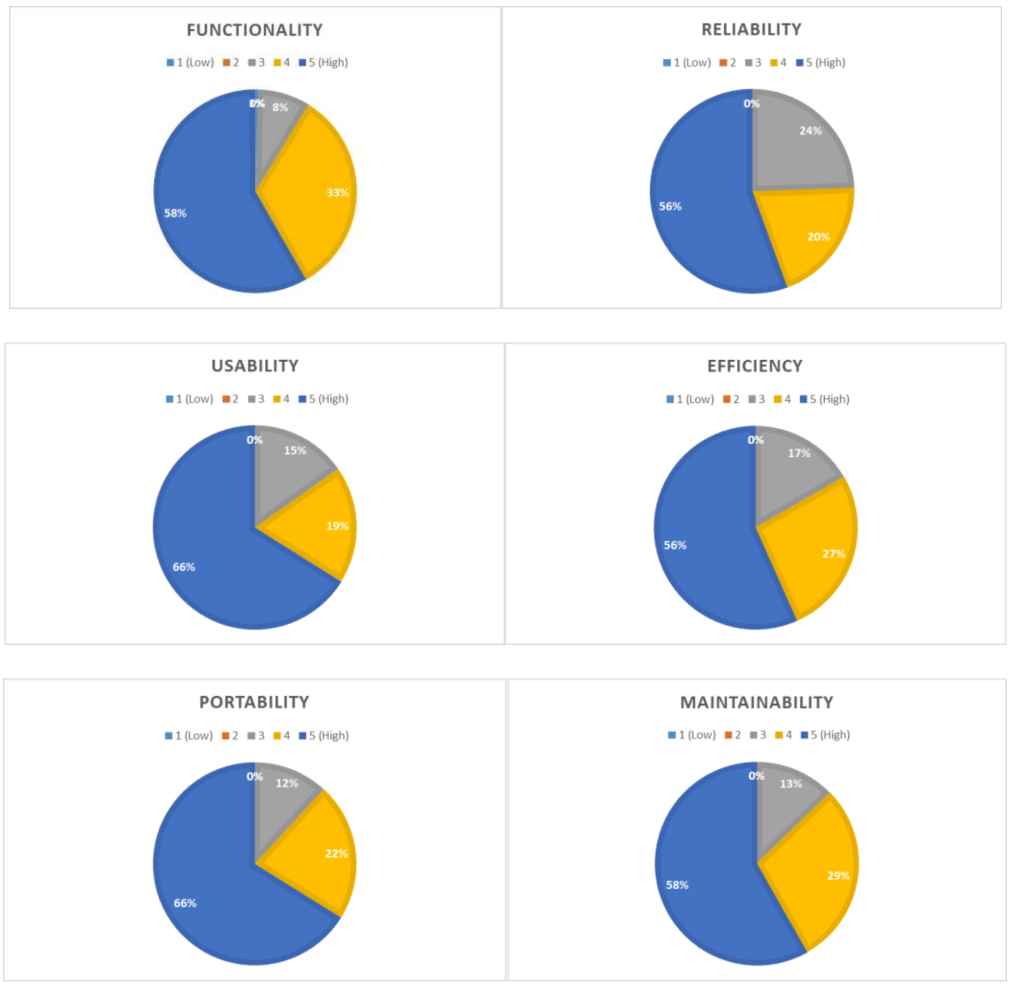


Figure 4-50: Evaluation scores Osteoarthritis use case

4.9.4 Nephroblastoma

This use case has been split up into two scenarios. The Nephroblastoma Educational Scenario is located within the MHA platform and is supported by the internal Generic Tool/Model Repository, whereas the Clinical Scenario is been implemented by the IAPETUS application built by ICCS and described in Deliverable 5.2.

4.9.4.1 Nephroblastoma Educational Scenario – Generic Tool/Model Repository

The Nephroblastoma Educational Module is supported in its operation by the Generic Tool/Model repository. The latter is built with Python's Django, which allows a variety of functionalities via its model-view-template system. In this case, a single template is responsible of serving as Graphical User Interface. In this template, the user has only to enter a set of values pertaining to the treatment schema and hit the "Run Simulation" button. The views in the backend, will contact the repository, automatically retrieve the model executable file, pass the template input to the model and execute it using a subprocess.call() command, which takes as an argument, the command that would be used in a command prompt environment. The results are finally collected and presented to the user.

Both of these modules are a part of the platform. The Generic Tool/Model repository resides within a virtual machine, created in the MHA private cloud part, residing at partner FORTH's premises. The repository is callable by using the URL http://139.91.210.38:9000/mhatools/index/ . The model for the Nephroblastoma educational Module has been uploaded into the repository by the platform administrators. The Django template that serves as its GUI is located at the URL http://139.91.210.38:9000/mhatools/wilms/. The platform users, however, do not have to enter this URL to access the module. It is accessible through the "Toolbox" menu in the MHA platform, which means that a user has only to log in to their account, and choose "Nephroblastoma Educational" from that menu. The platform will call the GUI's URL and present its content to the user using an IFrame tag.

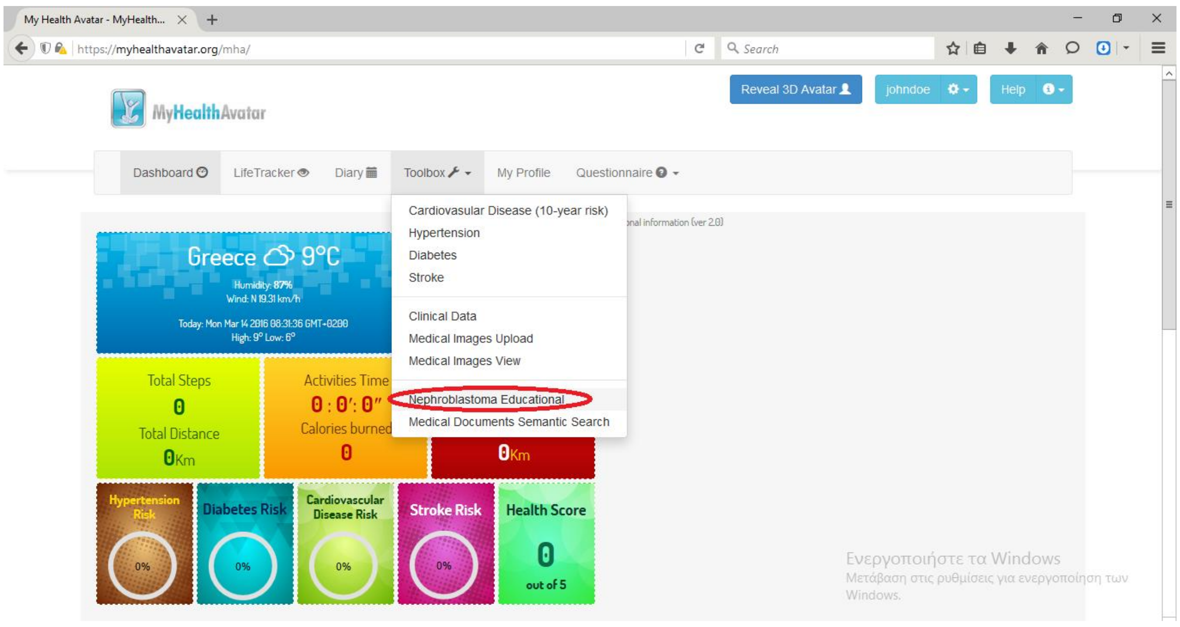
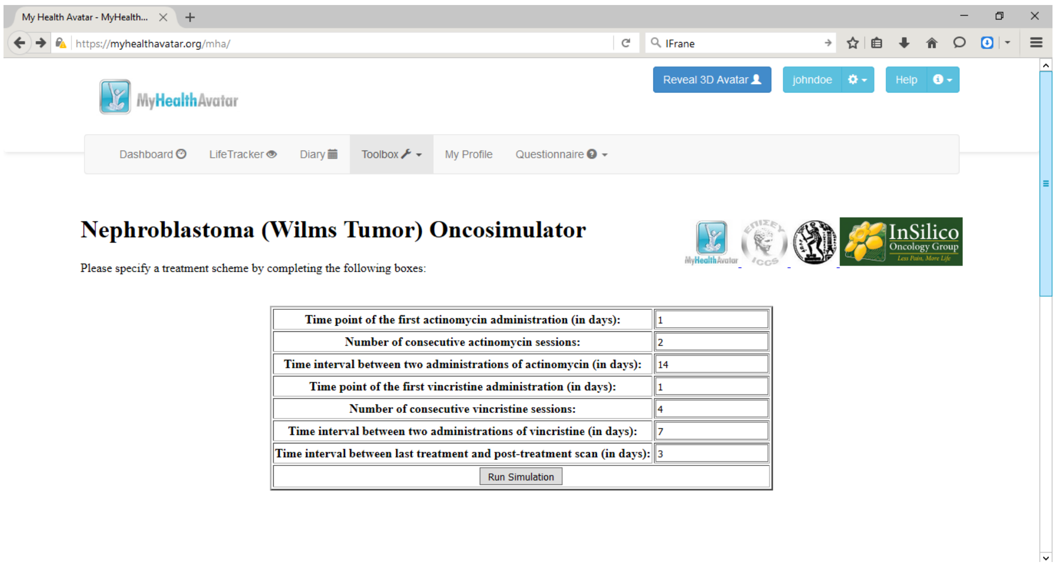


Figure 4-51: Choosing the Nephroblastoma Educational Module from within MHA platform



Nephroblastoma (Wilms Tumor) Oncosimulator

Please specify a treatment scheme by completing the following boxes:

Time point of the first actinomycin administration (in days):	1
Number of consecutive actinomycin sessions:	2
Time interval between two administrations of actinomycin (in days):	14
Time point of the first vincristine administration (in days):	1
Number of consecutive vincristine sessions:	4
Time interval between two administrations of vincristine (in days):	7
Time interval between last treatment and post-treatment scan (in days):	3
Run Simulation	

Figure 4-52: Nephroblastoma Educational Module interface

The Generic Tool/Model is currently available to the platform administrators and/or modelers with special clearance to upload their work. It is accessed directly through its URL, and its use follows the guidelines and procedures described in deliverable 5.2, albeit in a “lighter” and simpler form than IAPETUS. After uploading a model, a template can be constructed that can handle the inputs and the execution results, similar to the Nephroblastoma Educational Scenario

4.9.4.2 Nephroblastoma Clinical Scenario – IAPETUS

Introduction

IAPETUS is a prototype application built by ICCS that includes a Tool/Model repository, and a module with functionalities for executing Oncosimulator and other VPH models. It is described in D5.2 and carries the implementation of the Nephroblastoma Clinical Scenario, where a clinician and their patient and/or parents of a patient are involved. The inherent Tool/Model repository serves as the Onco-simulation Tool/Model repository described in section 5.4.3

Integration with MHA platform

IAPETUS is a standalone web application that resides outside the MHA platform and acts as a third-party external application. All of IAPETUS’s communications are done through API calls. The integration of the Nephroblastoma Clinical Scenario to the MyHealthAvatar platform is connected to the integration with the CHIC repository, as described in section 5.7.2. Since the data used in the Clinical Scenario’s execution are chosen from within the CHIC repository, using the MHA platform, then, according to figure 5-10, IAPETUS communicates with the platform in two occasions. First, a message is passed from the platform to the application, which informs the latter to begin the scenario execution by fetching the proper data from the CHIC repository. This is done using the CHIC repository API with the credentials created for the MyHealthAvatar project, as part of the CHIC-MyHealthAvatar agreement. The retrieved information is presented to the clinician, on the IAPETUS Oncosimulation wizard, as presented in D9.3.

After the Oncosimulator execution, the results are consolidated from the model’s output files into a PDF report. This report is sent to the platform using the URL https://myhealthavatar.org/mha/api/file/upload , which is part of the MHA API, through python with a CURL command of the following form:

curl -X POST -H "Authorization: Bearer <oath_token>" "Content-Type: multipart/form-data" -F "data=@<file_path>" https://myhealthavatar.org/mha/api/file/upload

where <file_path> is the full path of the report file and <oath_token> is the MHA access token. This token is obtained when the application is launched via a link from MHA platform. The result of this action is a JSON string, containing information about the uploaded file, along with a unique ID, that can be used to download it

In case the application wants to store additional string data (e.g. sets of result values), which are tied to a file, then a similar curl command is given through python

curl -X POST -H "Authorization: Bearer <oath_token>" "Content-Type: application/json" -F "data=<JSON_data>" https://myhealthavatar.org/mha/api/result

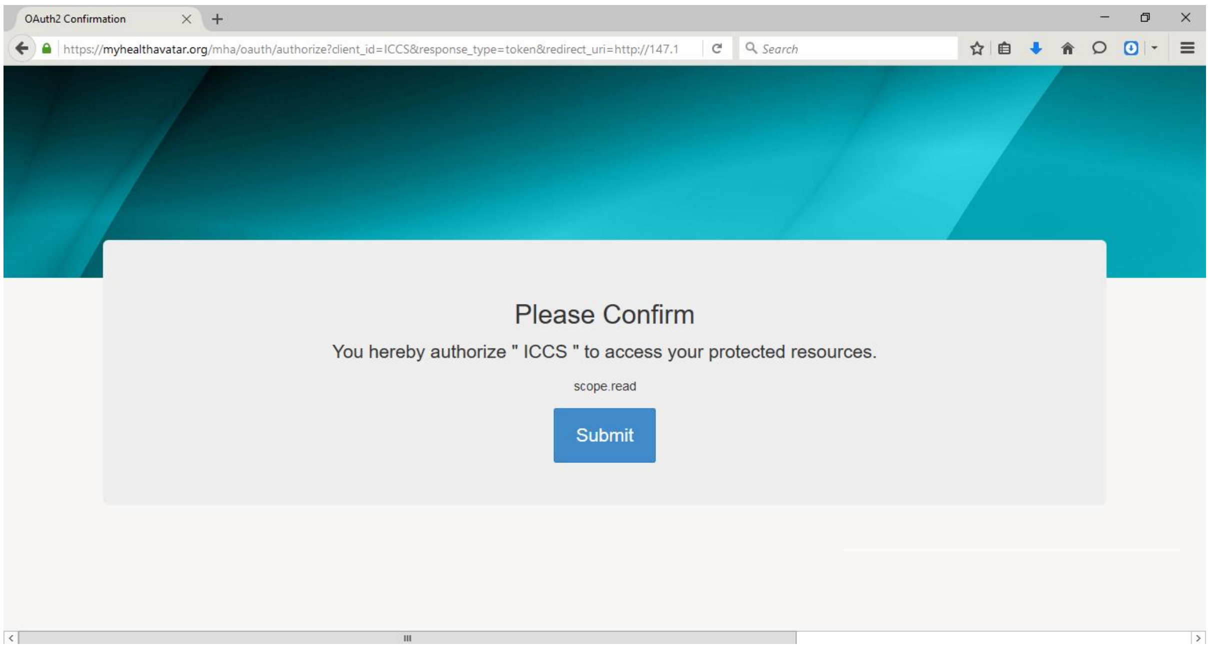
where <JSON_data> is a valid JSON string, containing the values to be uploaded.

To retrieve data, IAPETUS uses the following web services, in similar curl commands:

- List all files stored within MHA platform for a patient: https://myhealthavatar.org/mha/api/file/list
- Get the information of a selected file: https://myhealthavatar.org/mha/api/file/get/ID where ID is a number used to identify the file.
- Download a selected file: https://myhealthavatar.org/mha/api/file/download/ID where ID is a number used to identify the file.
- List all additional JSON information: https://myhealthavatar.org/mha/api/result/list
- Get additional JSON information about a file: https://myhealthavatar.org/mha/api/result/get/ID where ID is a number used to identify the file.

Scenario Implementation - workflow

For the scenario implementation and, the user will first log in to the MHA platform and click a link to enter IAPETUS. Then a confirmation message will appear (Figure 5-53). Upon, accepting, the user will enter IAPETUS, and an access token will be created and passed to the application.



OAuth2 Confirmation

https://myhealthavatar.org/mha/oauth/authorize?client_id=ICCS&response_type=token&redirect_uri=http://147.1

Please Confirm

You hereby authorize " ICCS " to access your protected resources.

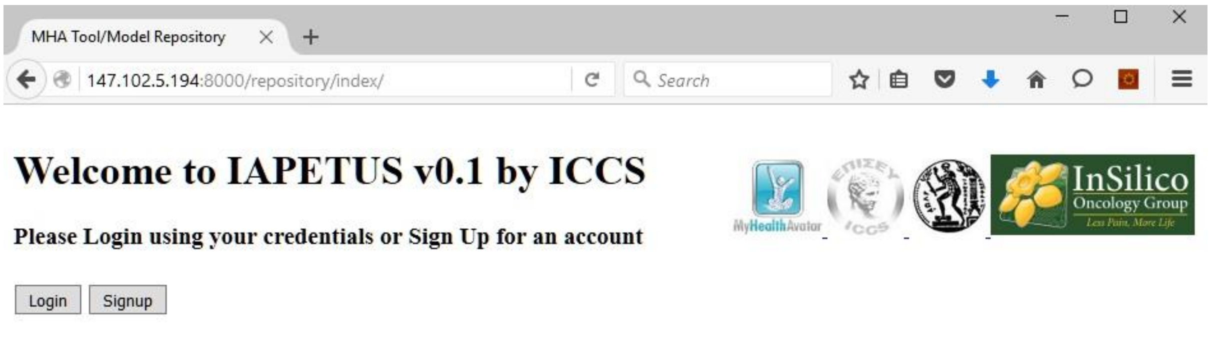
scope.read

Submit

Figure 4-53: Authorizing IAPETUS to access a patient's data.

After that, the initial screen of IAPETUS will be brought up, and the clinician, who is sitting next to the patient, will log in. Then, the clinician will use IAPETUS, to choose the cancer type, the Oncosimulator model that will be used and enter the necessary input. At a certain point, IAPETUS will connect with the CHIC repository and fetch the patient related files that will be fed to the Oncosimulator. The execution is carried out, the results are produced to the screen and the clinician can store the report as a pdf file, back to the MHA platform.

If the patient wants to download and view the pdf report from the MHA platform and view it via IAPETUS, this can be achieved by the Oncosimulation Module. Using the MHA API, as described above, the report can be downloaded and saved locally to the computer that is running the scenario.

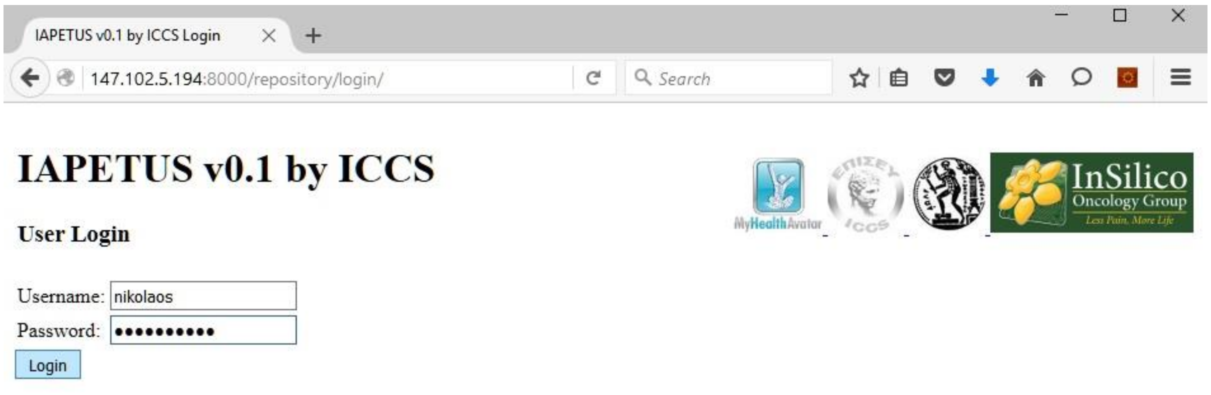

MHA Tool/Model Repository

147.102.5.194:8000/repository/index/

Welcome to IAPETUS v0.1 by ICCS

Please Login using your credentials or Sign Up for an account

Figure 4-42: IAPETUS initial screen


IAPETUS v0.1 by ICCS Login

147.102.5.194:8000/repository/login/

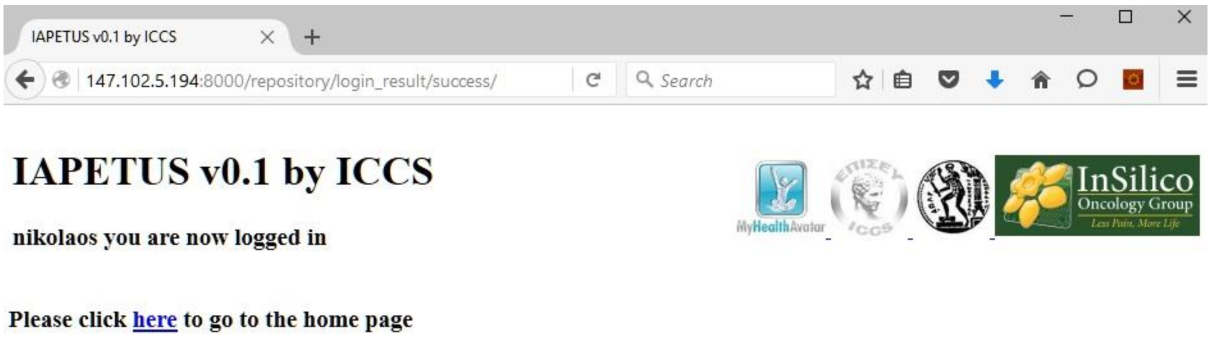
IAPETUS v0.1 by ICCS

User Login

Username: nikolaos

Password: ••••••••••

Figure 4-54: Clinician logging in


IAPETUS v0.1 by ICCS

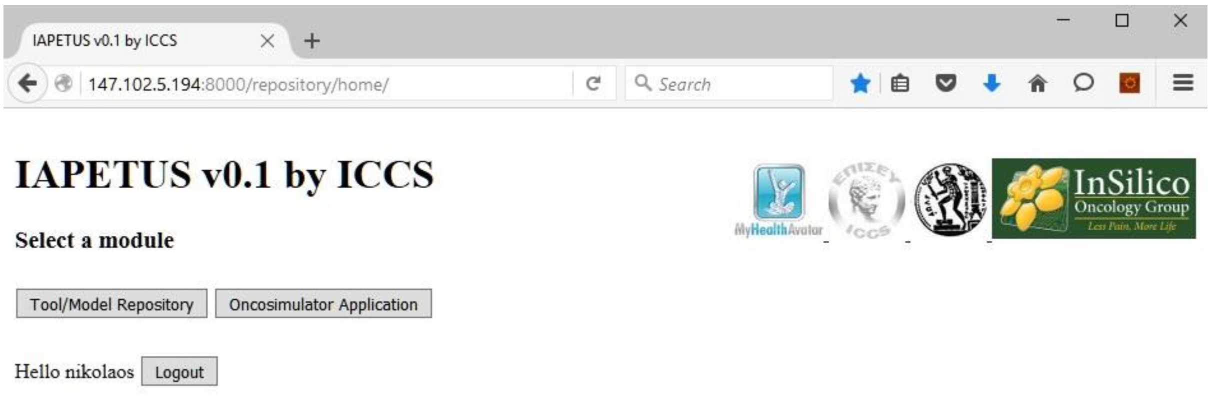
147.102.5.194:8000/repository/login_result/success/

IAPETUS v0.1 by ICCS

nikolaos you are now logged in

Please click here to go to the home page

Figure 4-55: Successful login


IAPETUS v0.1 by ICCS

147.102.5.194:8000/repository/home/

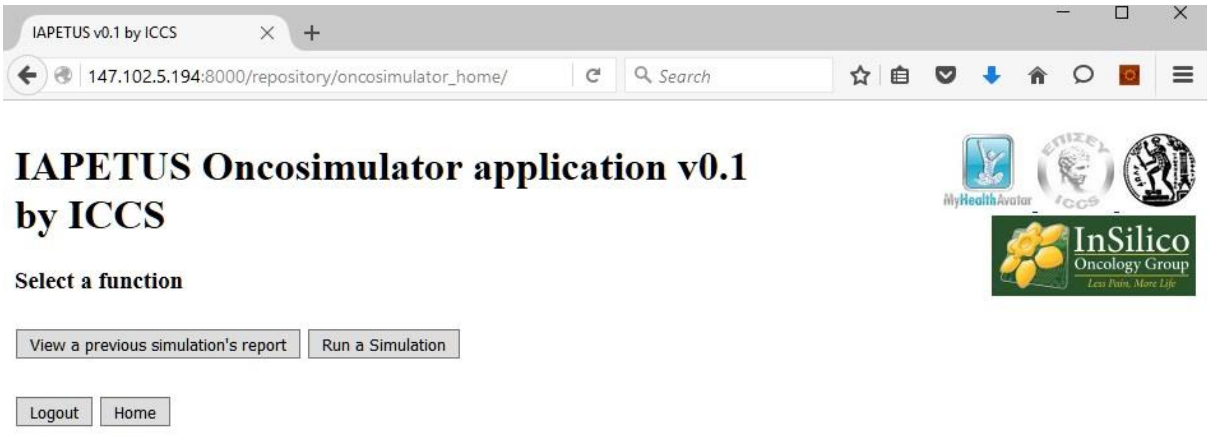
IAPETUS v0.1 by ICCS

Select a module

Tool/Model Repository | Oncosimulator Application

Hello nikolaos | Logout

Figure 4-56: IAPETUS main page


IAPETUS v0.1 by ICCS

147.102.5.194:8000/repository/oncosimulator_home/

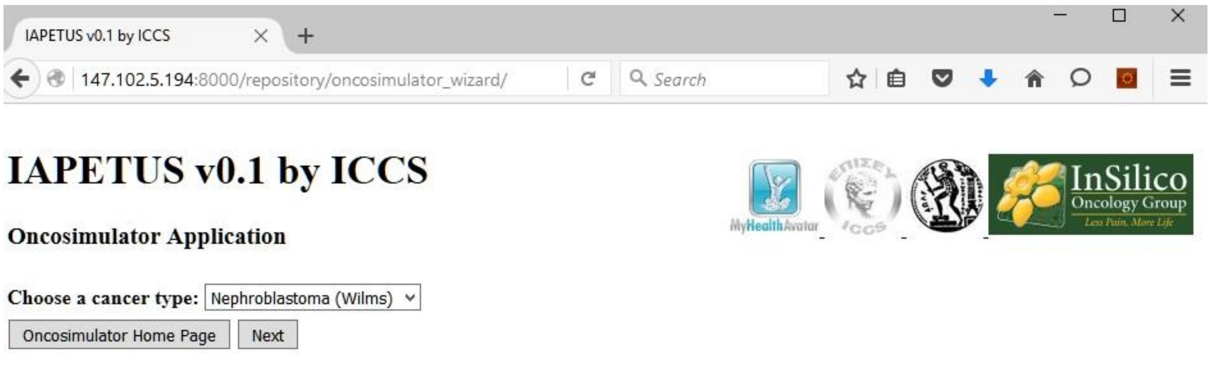
IAPETUS Oncosimulator application v0.1 by ICCS

Select a function

View a previous simulation's report | Run a Simulation

Logout | Home

Figure 4-57: IAPETUS Oncosimulator module main screen


IAPETUS v0.1 by ICCS

147.102.5.194:8000/repository/oncosimulator_wizard/

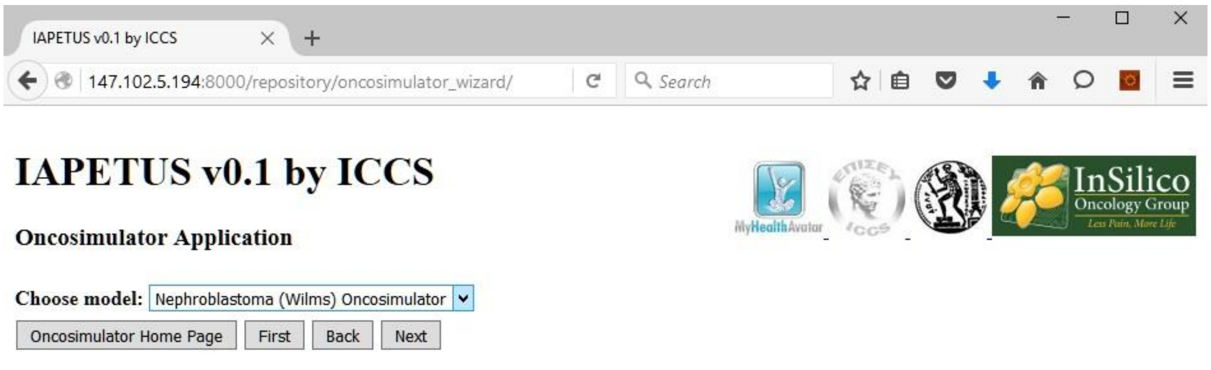
IAPETUS v0.1 by ICCS

Oncosimulator Application

Choose a cancer type: Nephroblastoma (Wilms)

Oncosimulator Home Page | Next

Figure 4-58: Choosing a cancer type

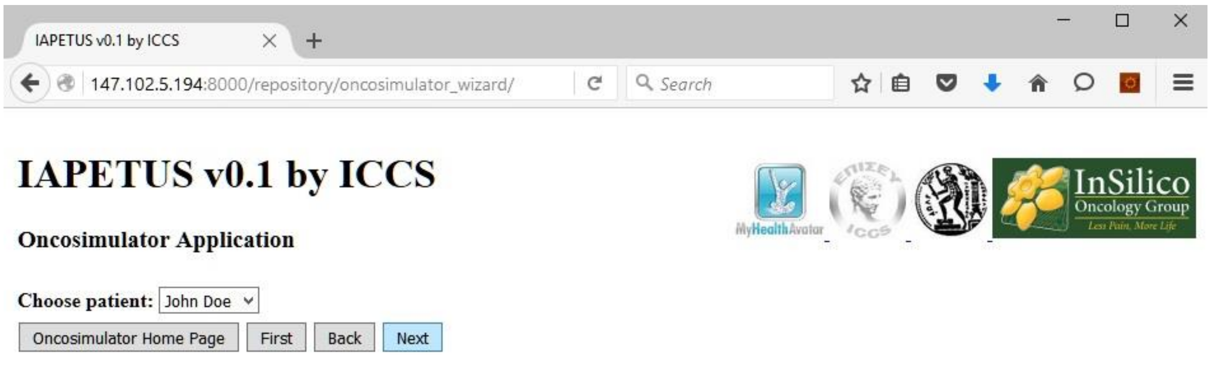


IAPETUS v0.1 by ICCS

Oncosimulator Application

Choose model: Nephroblastoma (Wilms) Oncosimulator

Figure 4-59: Choosing a simulation model (Wilms Oncosimulator)

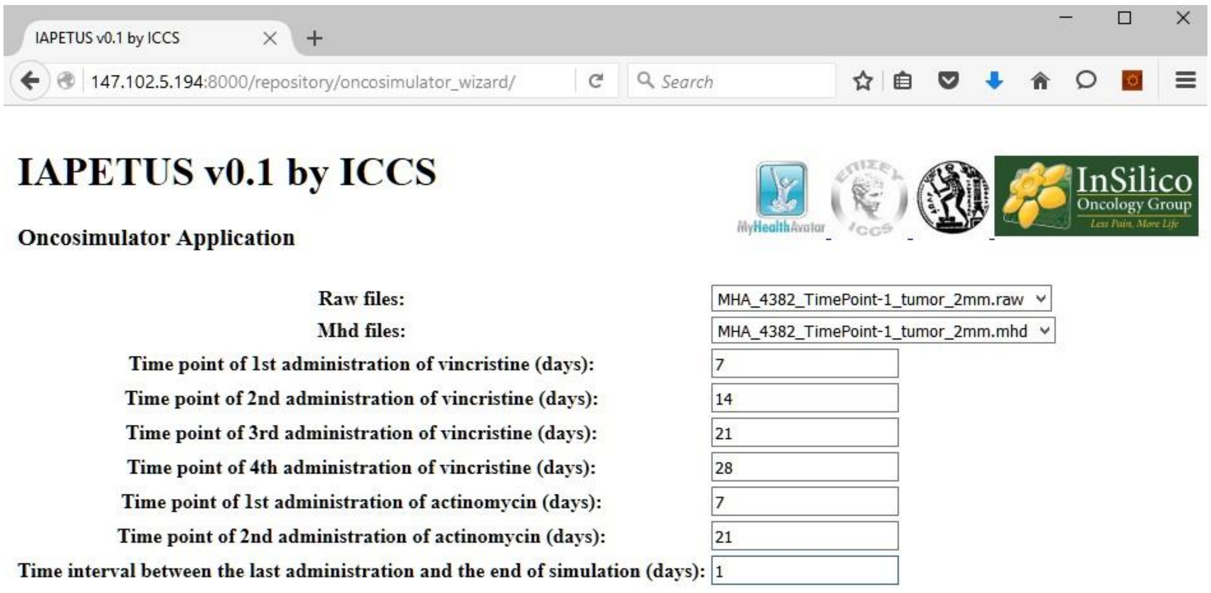


IAPETUS v0.1 by ICCS

Oncosimulator Application

Choose patient: John Doe

Figure 4-60: Choosing a patient.

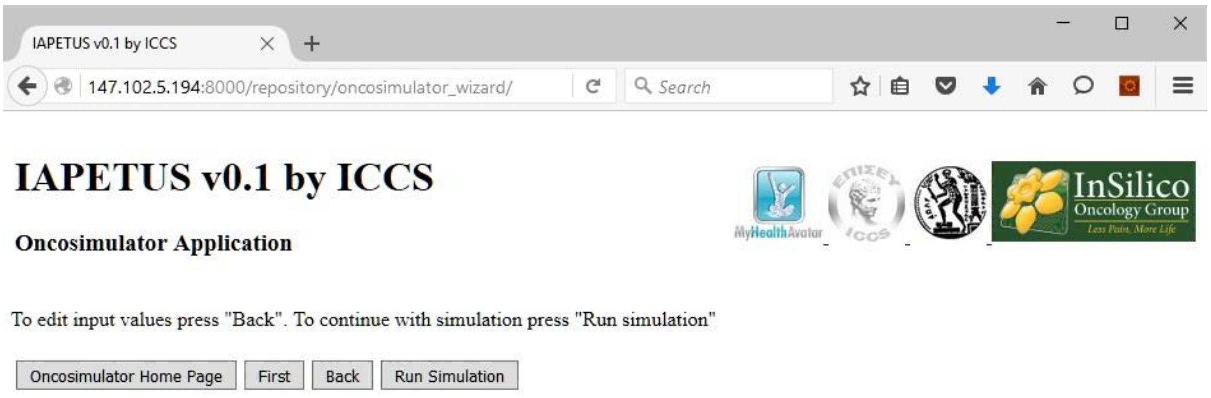


IAPETUS v0.1 by ICCS

Oncosimulator Application

Raw files:	MHA_4382_TimePoint-1_tumor_2mm.raw
Mhd files:	MHA_4382_TimePoint-1_tumor_2mm.mhd
Time point of 1st administration of vincristine (days):	7
Time point of 2nd administration of vincristine (days):	14
Time point of 3rd administration of vincristine (days):	21
Time point of 4th administration of vincristine (days):	28
Time point of 1st administration of actinomycin (days):	7
Time point of 2nd administration of actinomycin (days):	21
Time interval between the last administration and the end of simulation (days):	1

Figure 4-61: Giving the treatment scheme as input


IAPETUS v0.1 by ICCS

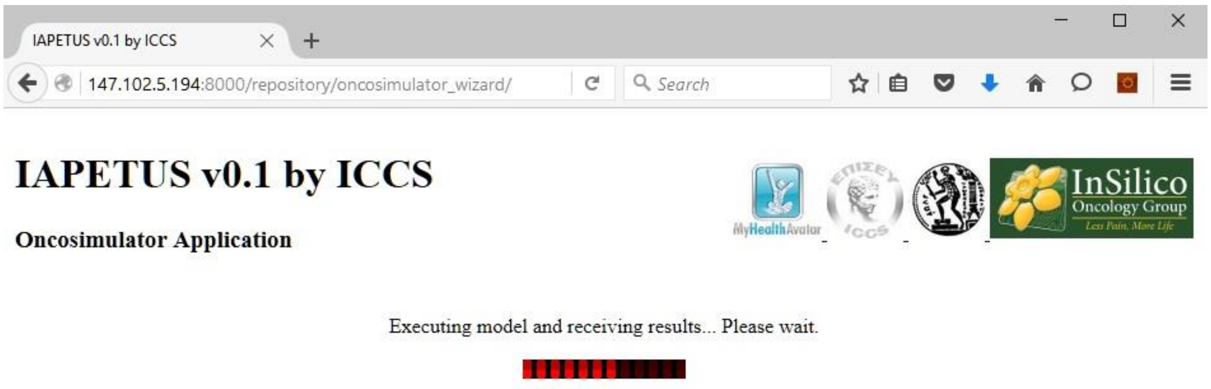
147.102.5.194:8000/repository/oncosimulator_wizard/

IAPETUS v0.1 by ICCS

Oncosimulator Application

To edit input values press "Back". To continue with simulation press "Run simulation"

Figure 4-62: Confirmation of input


IAPETUS v0.1 by ICCS

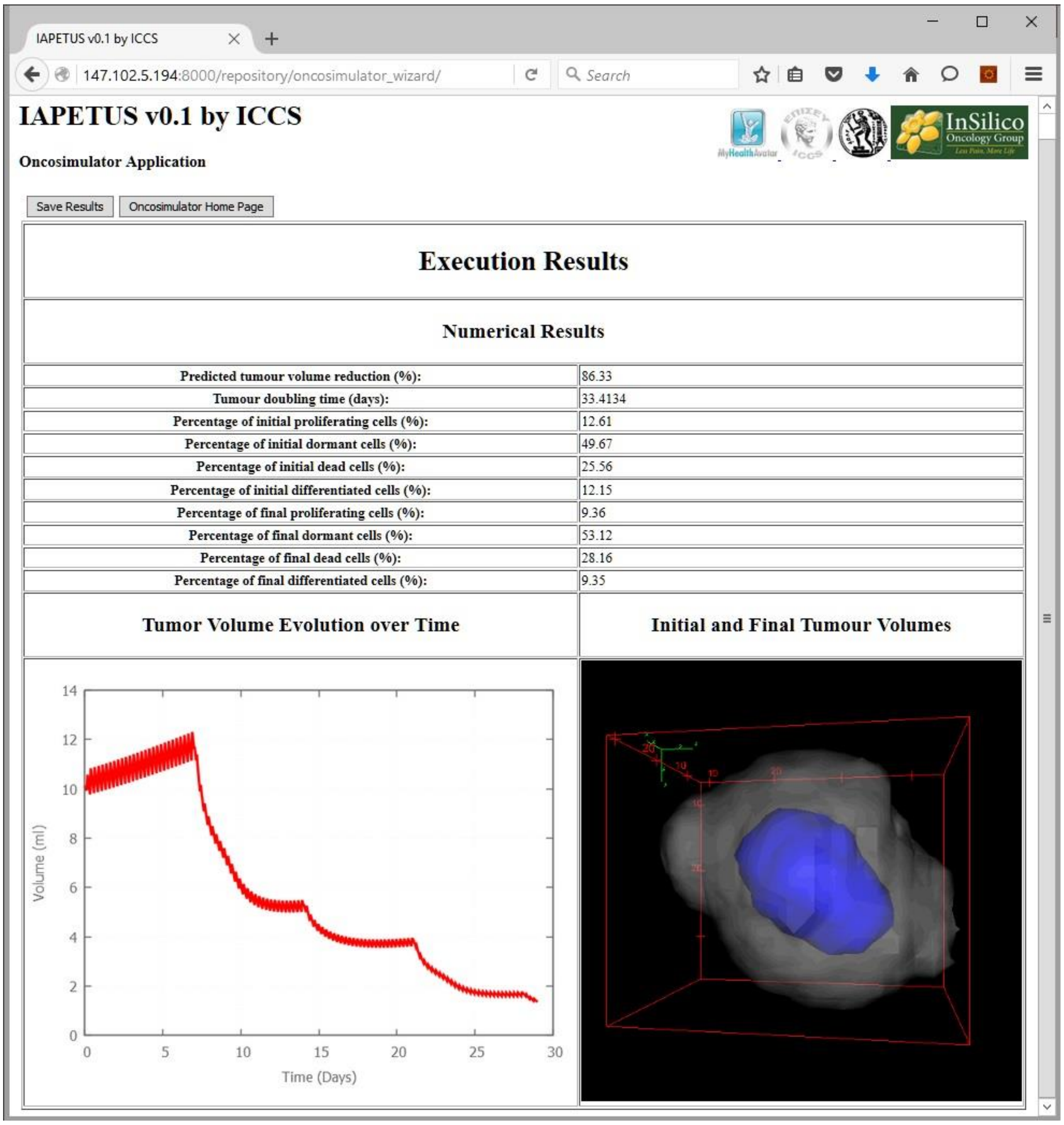
147.102.5.194:8000/repository/oncosimulator_wizard/

IAPETUS v0.1 by ICCS

Oncosimulator Application

Executing model and receiving results... Please wait.

Figure 4-63: Model execution.


IAPETUS v0.1 by ICCS

147.102.5.194:8000/repository/oncosimulator_wizard/

IAPETUS v0.1 by ICCS

Oncosimulator Application

Save Results | Oncosimulator Home Page

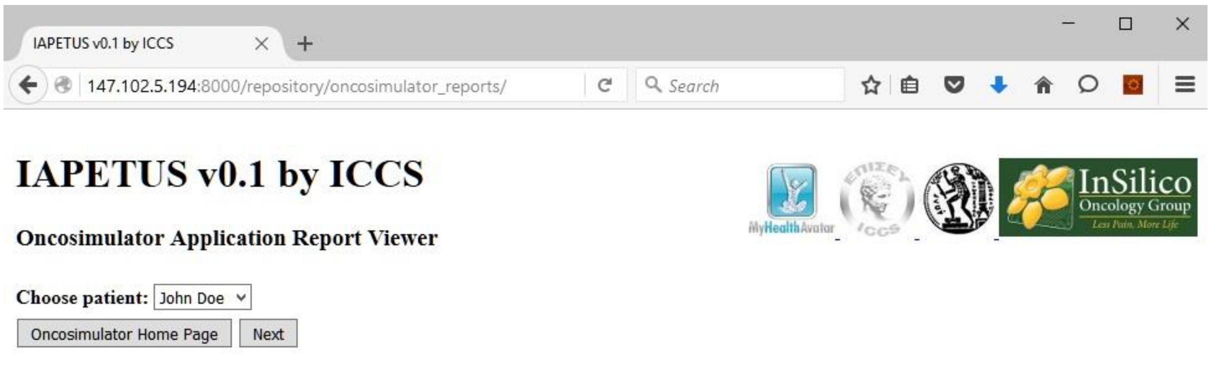
Execution Results

Numerical Results

Predicted tumour volume reduction (%):	86.33
Tumour doubling time (days):	33.4134
Percentage of initial proliferating cells (%):	12.61
Percentage of initial dormant cells (%):	49.67
Percentage of initial dead cells (%):	25.56
Percentage of initial differentiated cells (%):	12.15
Percentage of final proliferating cells (%):	9.36
Percentage of final dormant cells (%):	53.12
Percentage of final dead cells (%):	28.16
Percentage of final differentiated cells (%):	9.35

Tumor Volume Evolution over Time | **Initial and Final Tumour Volumes**

Figure 4-64: IAPETUS result screen


IAPETUS v0.1 by ICCS

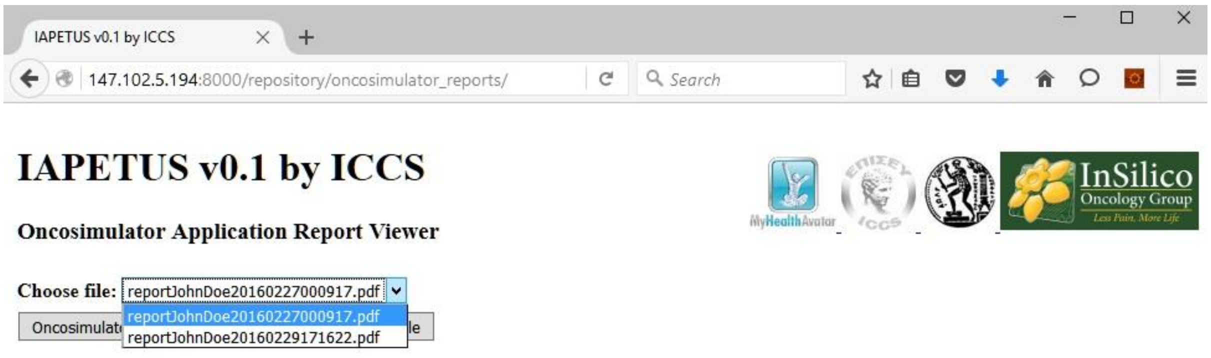
147.102.5.194:8000/repository/oncosimulator_reports/

IAPETUS v0.1 by ICCS

Oncosimulator Application Report Viewer

Choose patient: John Doe

Figure 4-65: Choosing a patient to download their report


IAPETUS v0.1 by ICCS

147.102.5.194:8000/repository/oncosimulator_reports/

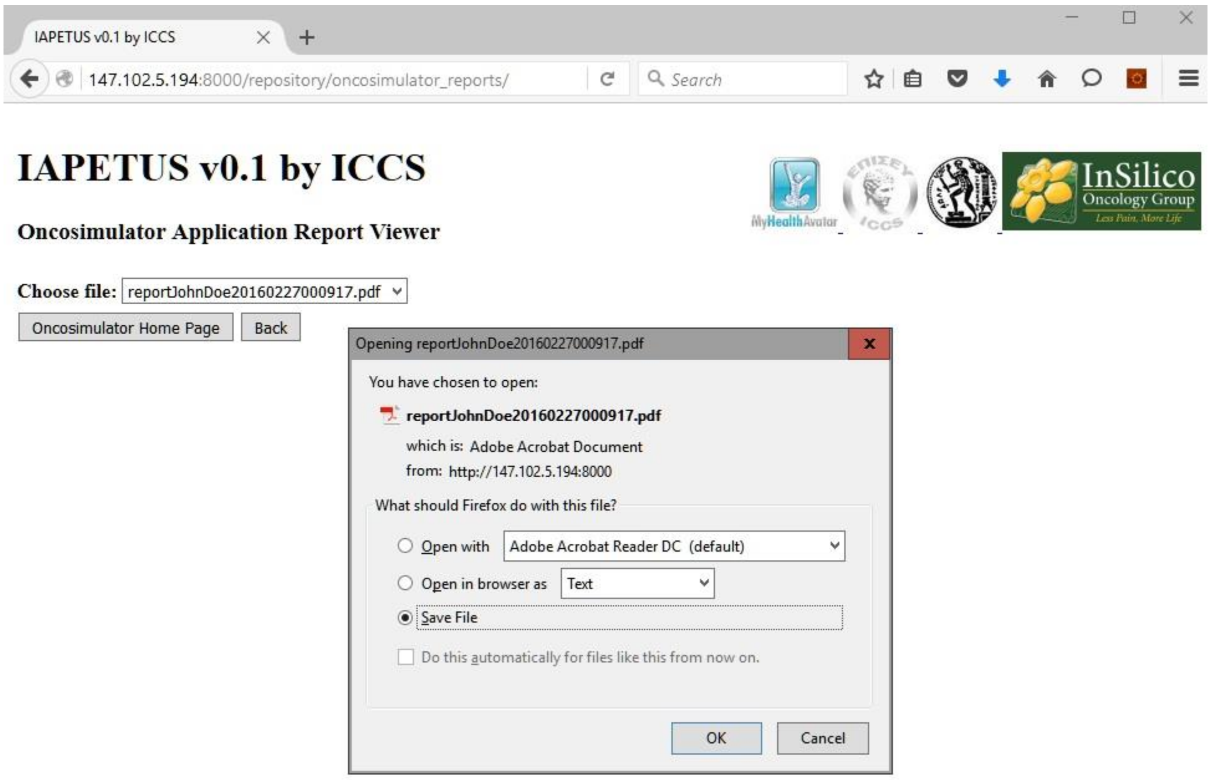
IAPETUS v0.1 by ICCS

Oncosimulator Application Report Viewer

Choose file: reportJohnDoe20160227000917.pdf

reportJohnDoe20160227000917.pdf
reportJohnDoe20160229171622.pdf

Figure 4-66: Choosing a report to download



IAPETUS v0.1 by ICCS

147.102.5.194:8000/repository/oncosimulator_reports/

IAPETUS v0.1 by ICCS

Oncosimulator Application Report Viewer

Choose file: reportJohnDoe20160227000917.pdf

Oncosimulator Home Page | Back

Opening reportJohnDoe20160227000917.pdf

You have chosen to open:

reportJohnDoe20160227000917.pdf

which is: Adobe Acrobat Document

from: http://147.102.5.194:8000

What should Firefox do with this file?

( ) Open with: Adobe Acrobat Reader DC (default)

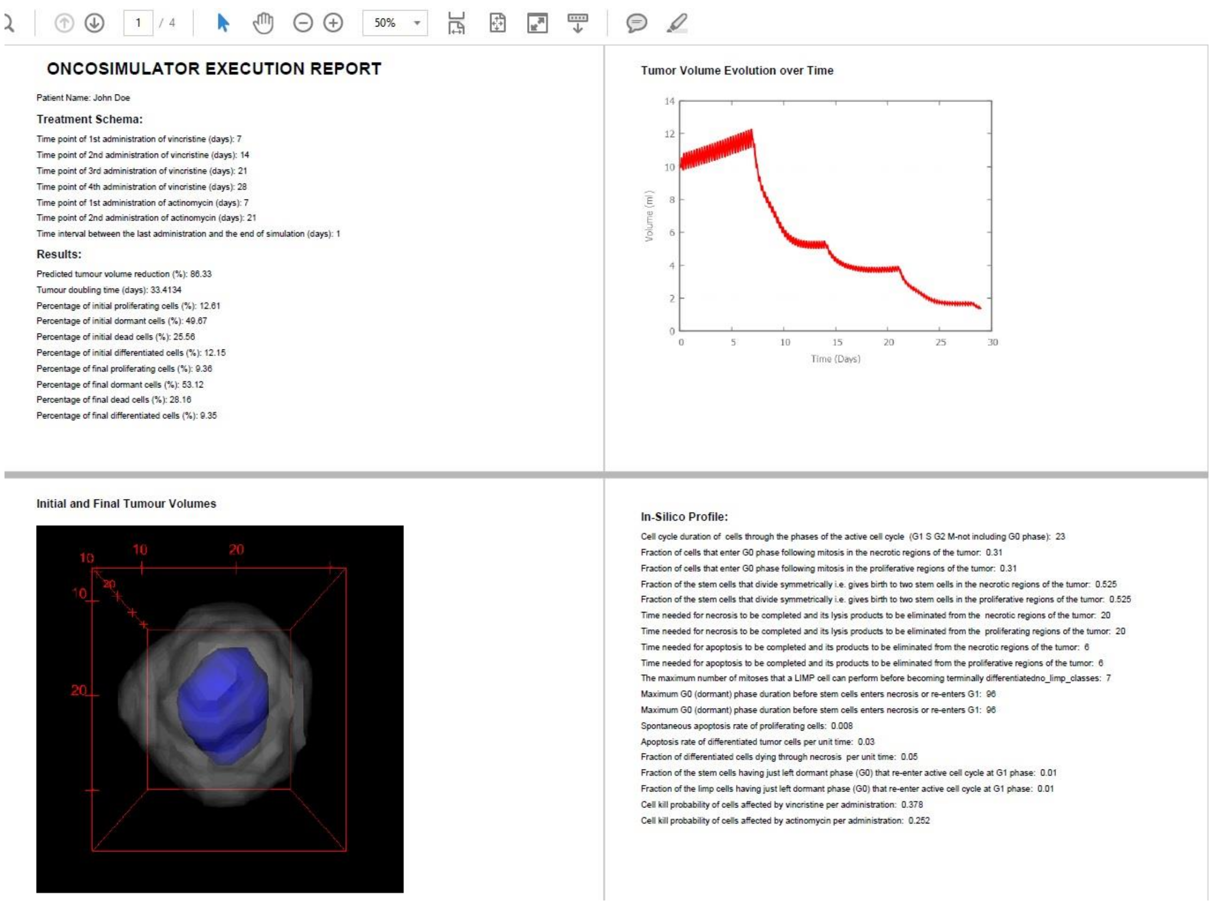
( ) Open in browser as: Text

(●) Save File

☐ Do this automatically for files like this from now on.

OK | Cancel

Figure 4-67: Downloading the report


1 / 4
50%
Tumor Volume Evolution over Time
Volume (ml)
Time (Days)
Initial and Final Tumour Volumes

ONCOSIMULATOR EXECUTION REPORT

Patient Name: John Doe

Treatment Schema:

Time point of 1st administration of vincristine (days): 7
Time point of 2nd administration of vincristine (days): 14
Time point of 3rd administration of vincristine (days): 21
Time point of 4th administration of vincristine (days): 28
Time point of 1st administration of actinomycin (days): 7
Time point of 2nd administration of actinomycin (days): 21
Time interval between the last administration and the end of simulation (days): 1

Results:

Predicted tumour volume reduction (%): 86.33
Tumour doubling time (days): 33.4134
Percentage of initial proliferating cells (%): 12.61
Percentage of initial dormant cells (%): 49.67
Percentage of initial dead cells (%): 25.56
Percentage of initial differentiated cells (%): 12.15
Percentage of final proliferating cells (%): 9.36
Percentage of final dormant cells (%): 53.12
Percentage of final dead cells (%): 28.16
Percentage of final differentiated cells (%): 9.35

In-Silico Profile:

Cell cycle duration of cells through the phases of the active cell cycle (G1 S G2 M-not including G0 phase): 23
Fraction of cells that enter G0 phase following mitosis in the necrotic regions of the tumor: 0.31
Fraction of cells that enter G0 phase following mitosis in the proliferative regions of the tumor: 0.31
Fraction of the stem cells that divide symmetrically i.e. gives birth to two stem cells in the necrotic regions of the tumor: 0.525
Fraction of the stem cells that divide symmetrically i.e. gives birth to two stem cells in the proliferative regions of the tumor: 0.525
Time needed for necrosis to be completed and its lysis products to be eliminated from the necrotic regions of the tumor: 20
Time needed for necrosis to be completed and its lysis products to be eliminated from the proliferating regions of the tumor: 20
Time needed for apoptosis to be completed and its products to be eliminated from the necrotic regions of the tumor: 6
Time needed for apoptosis to be completed and its products to be eliminated from the proliferative regions of the tumor: 6
The maximum number of mitoses that a LIMP cell can perform before becoming terminally differentiatedno_limp_classes: 7
Maximum G0 (dormant) phase duration before stem cells enters necrosis or re-enters G1: 96
Maximum G0 (dormant) phase duration before stem cells enters necrosis or re-enters G1: 96
Spontaneous apoptosis rate of proliferating cells: 0.008
Apoptosis rate of differentiated tumor cells per unit time: 0.03
Fraction of differentiated cells dying through necrosis per unit time: 0.05
Fraction of the stem cells having just left dormant phase (G0) that re-enter active cell cycle at G1 phase: 0.01
Fraction of the limp cells having just left dormant phase (G0) that re-enter active cell cycle at G1 phase: 0.01
Cell kill probability of cells affected by vincristine per administration: 0.378
Cell kill probability of cells affected by actinomycin per administration: 0.252

Figure 4-68: Oncosimulator report (in pdf)

Evaluation

In D9.3, the workshops and the evaluation procedure followed for the two Nephroblastoma Modules was described. In the following tables and figures, the mean values of the answers and pie charts showing the percentages of the answers, are presented. The scale is 1 to 5, 5 stands for "High" (best result) and 1 for "Low" (worst result).

Nephroblastoma Educational module

Educational Module of Nephroblastoma high-end scenario has been performed by 10 respondents (researchers and healthcare professionals) familiar with cancer models.

Question	Mean Value
Functionality	
Can the toolbox interact with the MHA platform?	4,1
Can the user set model input through the application?	5
Can the user submit an execution of the Nephroblastoma Oncosimulator model through the tool?	4,9

Is the user interface for execution submission of Oncosimulator user friendly?	4,4
Is the presentation of the results satisfying?	4,7
Is the use case educational?	4,3
Efficiency	
Is the tool comprehensible?	4,5
Can you learn how to use the system easily?	4,7
Is support of a technical person needed in order to use this tool?	1,6
Compatibility	
Is the tool running and the results presented independently of the software (windows version/web browsers) available on user's pc?	4,5
Do you know other similar tools? If yes is this tool better than the other you know?	2,2
Usability	
Is the execution of the model easy?	4,8
Is the execution time consuming?	2,3
Can the tool resume working & restore lost data after failure?	2,3
Does the interface provide all required information?	3,8
Reliability	
How accurate and complete is the software for the intended use?	4,1
Is the output trustful?	4,5
Are the results presented sufficient for educational purposes?	4,2
Quality in Use	
Does the software improve the time or reduce resource for the intended goal?	3,8
Does the software satisfy the perceived achievement of pragmatic goals?	4,1
Can the software harm people in the intended contexts of use?	1,4

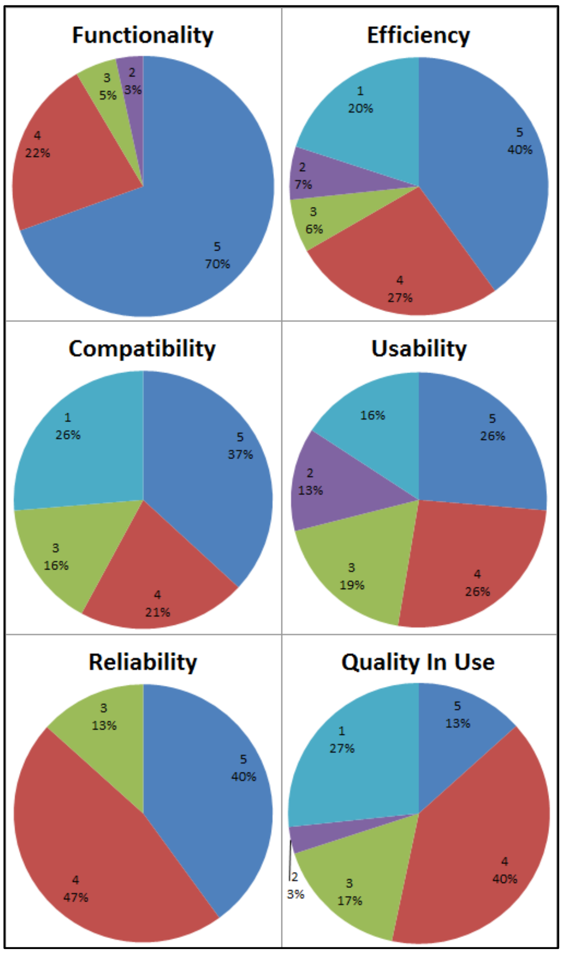


Figure 4-69: Results by percentages of the Nephroblastoma Educational Module evaluation

Nephroblastoma Clinical Module

Clinical Module of Nephroblastoma high-end scenario has been performed by 9 respondents (healthcare professionals) familiar with cancer models.

Question	Mean Value
Functionality	
Can the web application call the Nephroblastoma Oncosimulator?	4,8
Can the web application perform an execution of the Nephroblastoma Oncosimulator successfully?	4,9
Can the application fetch the clinical data (image files) from an outside source (CHIC data repository) successfully?	4,8
Can the user set model input through the application?	4,8
Can the user submit an execution of the Nephroblastoma Oncosimulator model through the web service?	4,8
Is the user interface for execution submission of Oncosimulator user friendly?	4,4

Is the presentation of the results satisfying?	4,1
Is the NEPH-UC clinically relevant?	4,4
Efficiency	
Is the application comprehensible?	4,4
Can you learn how to use the system easily?	4,7
Is support of a technical person needed in order to use this tool?	3
Compatibility	
Is the model running and the results presented independently of the software (windows version/web browsers) available on user's pc?	4,3
Can the system exchange data fluently with external modules?	4,1
Do you know other similar tools? If yes is this tool better than the other you know?	2,4
Usability	
Is the execution of the model easy?	4,9
Is the execution time consuming?	2
Can the tool resume working & restore lost data after failure?	3,9
Does the interface provide all required information?	4,2
Is the produced report useful?	4
Reliability	
How accurate and complete is the software for the intended use?	4
Is the output trustful?	3,9
Are the results presented sufficient for clinical purposes?	4
Portability	
Can the tool be easily accessed from any pc?	4,9
Security	
Do you think your data are secure?	4
Are data accessible only to authorized users?	4,1
Does the system prevent unauthorized access?	4
Quality in Use	
How accurate and complete is the software for the intended use?	4,3
Does the software improve the time or reduce resource for the intended goal?	4,7
Does the software satisfy the perceived achievement of pragmatic goals?	4,4
Can the software harm people in the intended contexts of use?	2,1

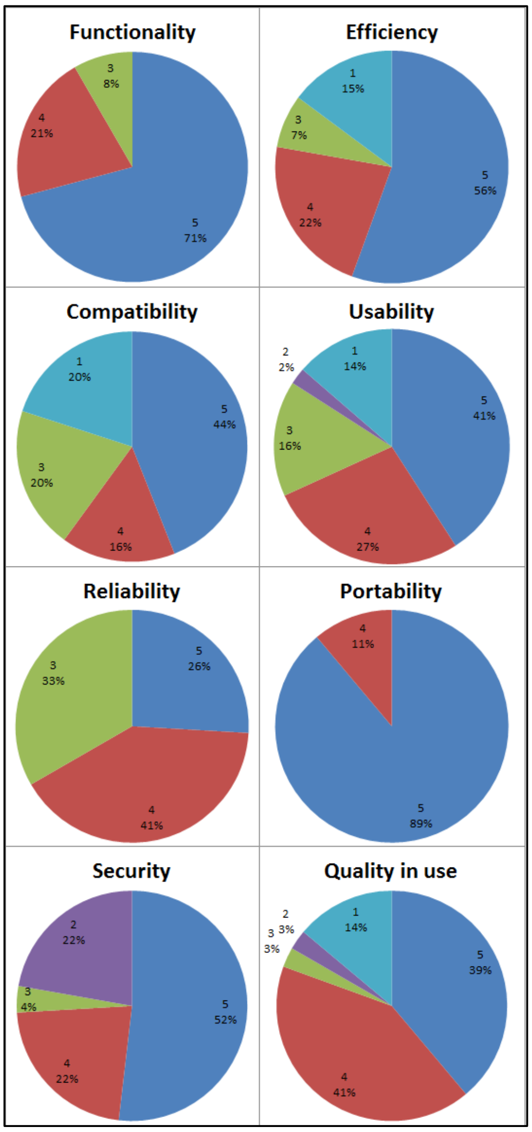


Figure 4-70: Results by percentages of the Nephroblastoma Clinical Module evaluation

5 Conclusion

The complexity of the MHA infrastructure and the corresponding software system is challenging. Complexity is a key concern that we addressed in providing intellectual intractability to make the system understandable and intellectually manageable by a) providing abstractions that hide unnecessary detail, b) providing unifying and simplifying concepts, c) decomposing the system into its elementary parts so as to allow for reasoning about its structural properties; and Management intractability. Our integration activities intend to make the development and deployment of the system easier to manage by enhancing communication, providing better work partitioning with decreased and/or more manageable dependencies. The evaluation of the integrate platform ensures that we *have all the necessary pieces,* supporting the functionality or services required and satisfying user requirements and that the *pieces fit together in terms of* interface and relationships between the pieces. This deliverable described the latest architectural consideration, integration activities towards a final platform formulation and a short evaluation.