Remote Accessibility to Diabetes Management and Therapy in Operational Healthcare Networks

REACTION (FP7 248590)

ID2-8-3 Change request and re-engineering report 2

Date 2012-06-01

Version 1.0

Dissemination Level: Restricted

Table of Content

Document control page

Code	ID2-8-3_Change-request-and-re-engineering-report-2_V10_IN-JET.doc
Version	1.0
Date	2012-06-01
Dissemination level	RE
Category	R
Participant Partner(s)	CNET, FHG-SIT, FORTH-ICS, FORTHNET, IMM, IN-JET, MSG, MUG, VUB
Author(s)	S. Asanin, M. Enzmann, F. Chiarugi, J. Karatzanis, V. Kontogiannis, D. Manoussos, M. Spanakis, S. Louloudakis, T. Klotzbücher, H. Udsen, B. Höll, G. Petrisch, S. Spat, K. Neubauer, P. Quinn, A-K Habbig
Verified and approved by	
Work Package	WP2
Fragment	No
Distribution List	All
Abstract	This deliverable contains an overview of Lessons Learned in cycle 2, a summary of validation results and the subsequent changes in requirements.
Status	☐ Draft ☒ Task leader accepted ☒ WP leader accepted ☒ Technical supervisor accepted ☐ Medical Engineering supervisor accepted ☐ Medical supervisor accepted ☐ Quality manager checked ☒ Project Coordinator accepted
Previous Versions	ID2-8-2 Change request and re-engineering report 1

Version Notes

Version	Author(s)	Date	Changes made
0.1	H. Udsen	2012-02-08	ToC, Section 2 and elements of Section 3
0.2	S. Asanin, M. Enzmann, F. Chiarugi, D. Manoussos, S. Louloudakis, S. Spat, T. Klotzbücher, H. Udsen, B. Höll, G. Petrisch, K. Neubauer, P. Quinn	2012-04-19	Contributions for Sections 1, 3, 4 and 5 plus Appendices A, B, C and D
0.3	A-K Habbig, F. Chiarugi, J. Karatzanis, V. Kontogiannis, M. Spanakis, H. Udsen	2012-05-03	Update of Sections 3.15/3.16, addition of Section 4.2, proofing and editing. Ready for internal review
1.0	H. Udsen	2012-06-01	Reviewer comments addressed; list of abbreviations added. Final version - submitted to EC

Internal review history

Reviewed by	Date	Comments made
P. Rosengren, CNET	2012-05-30	Approved with comments
T.F. Sørensen, IN-JET	2012-05-30	Approved with comments and suggestions

1 Abbreviations and acronyms

For the purposes of this deliverable, the following abbreviations and acronyms apply:

AHD	Application Hosting Device
BM	Bio Measurement
BMI	Body Mass Index
BPM	Blood Pressure Measurement
DB	Database
DCK	Device Connectivity Kit
EPR	Electronic Patient Record
GM	Glucose Measurement
GSM	Global System for Mobile Communication
HIS	Hospital Information System
HL7	Health Level 7
IHE	Integrating the Healthcare Enterprise
IHE-PCD	IHE Patient Care Device
IR	Infrared
JIRA	Issue tracker and requirement management tool
LIS	Laboratory Information System
MDD	Medical Device Directive
NMS	Network Monitoring System
P2P	Peer-to-peer
PID	Patient ID
PEG	Polyethylene Glycol
POCT	Point of Care Testing
RDMM	REACTION Data Management Model requirements (project in JIRA)
RSR	REACTION Security Requirements (project in JIRA)
SLED	Super Luminescent Light Emitting Diode
SMS	Short Message Service (Text message)
SOA	Service-Oriented Architecture
UI	User Interface

2 Introduction

2.1 Purpose, context and scope of this deliverable

This internal document describes the work performed in the second development cycle as part of the iterative requirements engineering process adopted for the REACTION project.

The document reports the Lessons Learned and the resulting change requests and re-engineering, as well as summaries of validation results and an assessment of the impact for the solution architecture.

The deliverable provides input for the analysis to be documented in *ID2-9-2 Updated requirements report 2*.

2.2 Content of the deliverable

Section 3 provides a brief summary of the research and development methodology defined for the REACTION project. A more detailed description can be found in *D2-8 The Requirement engineering process*.

Section 4 contains the first step in the re-engineering process; the collection and documentation of Lessons Learned. A total of 64 Lessons Learned have been collected during the second iteration cycle. The Lessons Learned are listed for each work package followed by an analysis and discussion of the changes in or new requirements derived from them.

Sections 5 and 6 include summaries of verification results (including unit tests, integration tests and system tests) and validation results (usability testing) of the enhanced In-hospital prototype and the first Primary care prototype developed in the second year. No field trials have been finalised at the present stage. An In-hospital field trial is in progress and a Primary care field trial is imminent.

Details of the testing and validation results for the In-hospital prototype are available in Appendices A through D.

Finally, Section 7 presents an assessment of the impact of the changes made on the system architecture and on compliance with the (revised) Medical Device Directive.

3 Research and development methodology

The REACTION project seeks to use the great potential that new technologies offer to address the major societal challenges of coping with the massive increase in number of citizens suffering from diabetes mellitus.

The REACTION solutions will be validated for use in both primary care (general practice) and secondary care (hospital general wards).

A description of the software engineering process and an overview of the iterative approach as they pertain to the REACTION project can be found in the internal deliverable *ID2-8-2 Change request and reengineering report 1*. A detailed account of the process is available in the deliverable *D2-8 The requirement engineering process*.

3.1 Re-engineering of requirements

After the successful completion of a prototype cycle, each work package analyses and reports their development results, RTD experiences, Lessons Learned in the development and integration work and other relevant knowledge gained during the development cycle. Knowledge gained from formal testing and system integration is also collected together with latest developments in technology, regulatory affairs and markets, which influence the REACTION solutions and their exploitability.

3.2 The REACTION approach to Lessons Learned

Lessons Learned help support project goals in the RTD work by promoting recurrence of successful outcomes and precluding the recurrence of unsuccessful outcomes.

The REACTION Lesson Learned process has six steps:

- Collection
- Verification
- Storage
- Dissemination
- Reuse
- Identification of improvement opportunity

The RTD work provides a large amount of Lessons Learned, by virtue of the many researchers participating in this. The Technical Management and the WP leaders have identified the Lessons Learned and verified them for correctness, significance, validity, and applicability.

All Lessons Learned have been entered into the Lesson Learned repository of the REACTION TWiki.

Again, further details can be found in internal deliverable ID2-8-2 Change request and reengineering report 1 and deliverable D2-8 The requirement engineering process.

4 Lessons Learned and requirements engineering

This section contains all Lessons Learned in cycle 2 and the subsequent requirements analysis. To facilitate referring to individual Lessons Learned they have been named LL followed by the relevant work package number and Lesson number (as they appear in the TWiki repository), e.g., LL WP4-1. The process results in the identification of a series of improvement opportunities and the need for new, changed and rejected requirements. The changes in requirements are commented and the impact on the present REACTION architecture is assessed.

The change requests are grouped per work package. The changes and updates to the requirements arising from the Lessons Learned are listed and discussed for each work package. The final re-engineering of the requirements will be analysed and discussed in detail in internal deliverable *ID2-9-2 Updated requirements report 2*, which will also contain the complete list of updated requirements.

A total of 64 Lessons Learned has been reported in the second iteration cycle, resulting in 119 new requirements, 2 updated requirements and 30 deleted requirements (mainly requirements that have been closed as being Out of Scope for REACTION).

Due to their non-technical nature no Lessons Learned have been reported in WP1, WP12 and WP13 in the second cycle. The same is true for WP11, because no demonstrations have yet taken place.

4.1 Lessons Learned in WP2

The work undertaken in WP2 relates to managing the process of requirements engineering and validation. IN-JET is the WP leader and five Lesson Learned have been collected and validated from this WP.

Org. No.	Experience and knowledge gained	Lesson Learned	Requirement affected
IN-JET 1	JIRA's potential as a requirement management tool has not been fully exploited	In addition to developing and implementing a tailor-made workflow, training of users and more frequent follow-up is required	(All)
FORTH-ICS 2	In order to have a high level management of all JIRA requirements it is necessary to have a main requirements project, instead of several JIRA sub-projects.	Use only one main requirements JIRA project and move (solving conflicts) all requirements from sub-projects (RDMM, RSR) to the main project. Do not generate any sub-project about requirements but simply use filters on the main requirements in order to select a specific subset.	(R-321 – R-468)
FORTH-ICS 3	Identification of a new architecture and component diagram.	Given the new list of main components each requirements has been reassigned to only one component of the new list.	(All)
FORTH-ICS 4	Higher involvement of users is necessary in the requirement management process, making clearer responsibilities and task assignments.	Upgrade the user licence in JIRA allowing the addition of further consortium members.	
FORTH-ICS 5	Detailed specifications for the primary care environment were finalized with some delay resulting in a significant delay in building components and all parts of the platform.	Higher commitment from partners is required in the specification phase in order to be able to match the deadlines in the development and tests.	

4.2 Change request and re-engineering originating from WP2

This Section provides an analysis of the five Lessons Learned from the work performed in WP2 in the second cycle. The resulting new requirements are reported under the associated work packages. No requirements were updated or deleted.

4.2.1 Analysis of Lessons Learned

LL WP2-1 recognised the need for a tailor-made workflow in the JIRA requirements database to improve monitoring the progress of the project. To utilise the potential of JIRA as a requirement management tool it was further recognised that user training and frequent follow-up is required. User training was offered in connection with the REACTION Plenary Meeting in Heraklion in September 2011. Follow-up is on-going.

LL WP2-2 also deals with monitoring of project progress. It was realised that one main project with high-level requirements was preferable to having to monitor several projects at the same time. Therefore two sub-projects, *REACTION Security requirements* and *REACTION Data Management Model requirements*, were copied or cloned into the main project.

LL WP2-3 introduced a new list of components based on the revised architecture. The principle of assigning one and only one of the new components meant changes to all requirements,

LL WP2-4 showed the need to extend the JIRA licence to allow for more dedicated JIRA users.

LL WP2-5 exposed the necessity of stronger adherence to internal deadlines and a more concerted effort in order to avoid further delays in building and testing of components.

4.2.2 New requirements

The copied or cloned requirements arising from the consolidation efforts of LL WP2-2 are numbered REACTION-321 through REACTION-468. New requirements considered to be Part of Specification are listed individually in the relevant work packages below.

4.2.3 Updated requirements

Though the implementation of a dedicated workflow has affected all requirements in the database it has not resulted in changes to the substance of the requirements. The same is true for the reassignment of new components to all requirements.

4.2.4 Deleted requirements

No requirements have been deleted.

4.3 Lessons Learned in WP3

The RTD work undertaken in WP3 involves the development of glucose sensors and monitoring and contextualisation of these. IMM is the WP leader and nine Lessons Learned have been collected and validated from this WP.

Org. No.	Experience and knowledge gained	Lesson Learned	Requirement affected
IMM 1	With the chip-based glucose sensor, design and shape of the fluidic channels are crucial for deployment with microdialysis concerning lag time	Lag time can be reduced by adapting the size of the fluidic channels and by using short tubings of the microdialysis needle. However, it has to be mentioned that the chip-based sensor is only an intermediate system for testing the optical functionality of the sensor principle. Later in the project the sensor is to be implemented in	R-268

		the needle directly which will drastically reduce the lag time.	
IMM 2	Thermal heating of electronic components, like operational amplifiers, resistors etc. can lead to signal drift of the glucose sensor	Use of components with low temperature coefficient (precision resistors and low offset drift amplifiers) improve the situation clearly	R-270, R-268, R-267, R-29
IMM 3	Glue bonding of the fluidic chips can lead to contamination of the channels with glue droplets, serving as seed crystal for bubble formation	Usage of other bonding methods like solvent bonding or thermal bonding	R-268, R-267, R-29
IMM, MUG 4	Bubble formation in the measuring channel has an enormous impact on the sensor signal (drift), especially if the bubble grows as function of time	This problem is generally an issue with microdialysis, it can be overcome only by applying suitable bubble traps, which are currently investigated	R-268, R-267, R-29
IMM 5	Differential heating of the liquids in the measuring and reference channels by IR radiation may be an issue for signal drift	Modulation of the light sources reduces the differential heating process	R-268, R-267, R-29
IMM 6	The fluidic chips used during the test and development phase have been milled, resulting in a comparatively large surface roughness of the channel walls, serving as potential seed crystals for bubble formation	For the testing phase a chip based on high surface quality glass components has been set up and is currently tested, for clinical trials hot embossed chips are envisioned	R-268, R-267, R-29
MUG 7	During microdialysis the recovery of the glucose concentration (ratio of the measured glucose concentration to the real glucose concentration) may change over time for several reason (e.g. sensor fouling, clogging of dialysis membrane).	To overcome this problem MUG has evaluated a method for correcting the measured glucose value by simultaneously measuring the ion density in the perfusate which changes in exactly the same manner as the measured glucose value. This technique is based on a conductivity measurement, alternative methods, based on an optical principal have to be thought of. Adding acetate to the perfusate and measuring the corresponding peaks in the IR spectrum might be a solution without the need of adding an additional measuring technique to the system.	R-268, R-267, R-29, R-478
IMM 8	With the fibre optic sensor problems with modal noise occurred when an SLED is used as a light source (single mode coupling), resulting in signal changes comparable or even bigger than those correlated with glucose concentration changes	To overcome this problem a multimode light source has to be used in combination with the fibre optic sensor. However, the disadvantage is a lower overall power requiring low loss connections between the sensor the source and the optical couplers.	R-268, R-267, R-29
IMM 9	The implementation of the fibre optic sensor into a micro needle causes problems with the filling of the sensor cavity, resulting in unwanted signal changes	The problem is not solved yet but it is suggested to apply a hydrophilic coating on the inner side of the optical cell to allow the water a full wetting of that area	R-268, R-267, R-29

4.4 Change request and re-engineering originating from WP3

This Section provides an analysis of the nine Lessons Learned from the work performed in WP3 in the second cycle. The work resulted in one requirement being added, one updated and one deleted.

4.4.1 Analysis of Lessons Learned

All Lessons Learned listed in the above table are dealing more or less with the glucose sensor accuracy so that all requirements referring somehow to the sensor accuracy are affected. However, only the requirement REACTION-270 was changed, taking into account that the IR difference spectroscopy glucose sensor can be maintained on a good accuracy level, as long as it is ensured that both the measuring and the reference channel are on the same temperature level. In that sense requirement REACTION-29 might be obsolete, since it covers the point of temperature stability of the IR glucose sensor.

LL WP3-7 was the basis for a new requirement REACTION-478. Since the application of a microdialysis based glucose sensor during the first clinical trials is most likely, the point of recovery change by sensor fouling was addressed in REACTION-478, introducing an additional sensor to monitor the recovery, if frequent recalibration of the sensor is going to be avoided.

4.4.2 New requirements

For the microdialysis based glucose sensor one new requirement REACTION-478 has been created.

Key	Requirement Type	Priority	Summary	Rationale	Fit Criterion
REACTION-478	Functional	Major	The recovery for microdialysis based glucose sensors should be monitored to avoid recalibration	Due to sensor fouling effects the recovery of a microdialysis catheter may change as a function of time, requiring sensor recalibration by a reference method. This could be avoided if an additional sensor is implemented measuring the change of recovery (e.g. ion density in the dialysate). This is relevant for microdialysis based glucose sensors.	Recovery detection implemented

4.4.3 Updated requirements

Requirement REACTION-270 has been changed.

Key	Requirement Type	Priority	Summary	Rationale	Fit Criterion
REACTION-270	Functional - REACTION platform	Major	Operating temperature of sensors should be specified and kept on equal level for the IR GM sensor reference and measuring channel	The temperature might influence the result of the measurement and its accuracy.	Either sensor manufacturers should specify the operating temperature of the sensors or the device should be able to adjust the measurement based on the temperature value (in this case a temperature sensor has to be integrated in the device)

4.4.4 Deleted requirements

REACTION-29 may be considered a Duplicate after the changes made to REACTION-270.

4.5 Lessons Learned in WP4

The RTD work undertaken in WP4 relates to data management and service orchestration. CNET is the WP leader and 11 Lessons Learned have been collected and validated from this WP.

Org. No.	Experience and knowledge gained	Lesson Learned	Requirement affected
FORTH-ICS 1	Early availability of data model and test population is fundamental for managing the prototype release in time.	Domain modelling has to be done prior to other development activities and be available together with the detailed specifications.	None
FORTH-ICS 2	To master problems of technical inaccuracies and errors in the database design to its next level, previously identified design deficiencies must be highlighted and prevent their perpetuation.	Either a shared repository (with versioning) for the data model or a selected partner in charge of applying modifications and distributing the new version to the consortium has to be used in the process.	-
FORTH-ICS 3	The population data for the primary care should be properly updated since we will obtain a population which makes sense both from a structural and logical point of view (e.g. reasonable values for the observations of the same patient, etc.). Several iterations have to be performed in order to reach this goal and to have useful data for the development, test and demos.	Continuous retrofits have to be provided by the involved partners in order to refine the population scripts (and eventually also the DB generation scripts). Versioning of the scripts and use of the repositories is mandatory.	-
CNET 4	In ID2-8-2 report 1, a decision was made to cooperate with the Continua Alliance although currently very few commercially available medical devices were Continua compliant. This allowed for additional development effort. Today, the estimation is that the number of additional Continua certified devices have not increased very much, leading the consortium in uncertainty regarding this issue.	Analyse new markets more in detail as adopting of standards and approaches are not always universal. A variable to look for is in particular the number of new adopters on the market as impact comes with its undertaking.	-
CNET 5	The design of a Continua Manager took a lot of effort but enabled us to quicker include additional agent specialisations when needed. However, having few new Continua devices to integrate in the platform became cumbersome.	The Continua Manager (i.e. REACTION Test Suite) provided instead support in overseeing the internal platform communication and behaviour.	-
CNET 6	Less than strong and devoted component contributions according to the Crete list affected the estimated REACTION infrastructure by not providing full described functionality.	The adoption of SOA in the Data Management subset still allowed us to use the Model-Driven Architecture and Web Services for seamless application development and deployment.	-
CNET 7	There is a need for patients to easily enter context information regarding	Experiments with different user interfaces for entering of context	

	nutrition and food intake. A first market survey showed a number of available apps for this. There is a need for the REACTION platform to be interoperable with these, as well as having our own solution for entering food data in a simple fashion.	information should be carried out	
CNET 8	It seems difficult to establish one standard for device integration/communication. In practice, IEEE11073-standards provide limited support for interoperability when it comes to device integration. For the end-users the same problems, can be solved more quicker and efficiently with freely available middleware solutions rather than waiting for new devices following the IEEE11073 standards. The time it takes to integrate a new device in a middleware solution is very short.	Reaction DCK (Device Connectivity Kit) has a promising commercial potential as device integration platform for home care/personal health monitoring application builders and efforts should be invested into improving guidelines, tutorials, and further support for more device types.	
CNET 9	Several new health devices has been launched under the paradigm “Device as a service”, for instance WiThings weight scale (which uploads weight, BPM and other data directly to services on the Internet). This moves the integration point from the device to a service on the net, and creates new interoperability at problems at the service level.	Service interoperability should be studied more in detail during iteration 3.	
CNET 10	There is a need for clinical users to express rules over the monitored signs and health status of the patient. Our analysis showed that a rule engine must go beyond simple rules acting on one incoming measurement, but also to deal with trends over time (such as if weight has increased by 5% the last 3 months then …)	The REACTION rule engine must be powerful and expressive yet flexible and configurable. It must also be extensible so that it integrates and can use existing services. At least the following rule types should be supported: - value with threshold. - Rules that process historical measurements - Rules that also access contextual data such as meal time. - Rules that access patient data.	
CNET 11	Several REACTION services are becoming available (data collection, context collection, long term risk engines, SMS alerting, et c). New mechanisms that allow a flexible combination of services in secure way are needed.	Service Orchestration must be easy to configure. It must also be easy to find available services to use and combine. Service Orchestration should build on the Rule engine developed in Reaction and extend it with service discovery.	

4.6 Change request and re-engineering originating from WP4

This Section provides an analysis of the 11 Lessons Learned from the work performed in WP4 in the second cycle. The work resulted in 80 requirements being added and 13 deleted. No requirements were updated.

4.6.1 Analysis of Lessons Learned

LLs WP4-1, WP4-2, WP4-3 and WP4-6 all are concerned with design, development and cooperation issues.

LL WP4-3 specifically stresses that all types of software files used within the consortium should be versioned for reliable refinement and retrofitting, while LL WP4-6 underpins the decision to use a SOA solution, because this approach makes it possible to build and add new services and components to the existing framework.

LLs WP4-4, WP4-5 and WP4-8 are related to the decision to cooperate with the Continua Alliance and adopt the Continua Guidelines for compliance. It goes without saying that the REACTION client environment should not just be composed of medical devices but also of environmental and context-giving sensors and/or user input. The baseline technology in REACTION is based on the LinkSmart Middleware, which offers the possibility of easy integration of a variety of devices including medical devices. Through the Device Connectivity Kit LinkSmart already has the ability to incorporate IEEE 11073 agent specialisations, but the market is not yet mature for full-scale adoption. Therefore, the LLs here are repeating what was stated in the previous version of this document: Continua guidelines and the availability of compliant devices are at an early stage of development and not widely adopted in the market.

LL WP4-7 concerns the relevance of context data. While context is relevant in the data collection stage of the primary care scenarios it is the most vital part of the data fusion functionality which is to be integrated in the next iteration.

LL WP4-9 describes the advent of the 'Device-as-a-service' concept and the need to take into account service interoperability and to consider how services are represented by the increasing number of advanced devices on the market. This further exposes the fragility of exclusively going for Continua compliance.

The rules referred to in LL WP4-10 and additional services described in LL WP4-11 imply that the Data Fusion Engine should automatically generate context data which are closely linked to the available services in the REACTION platform.

4.6.2 New requirements

80 new requirements have been added as a consequence of the consolidation described in LL WP2-2.

Key	Requirement Type	Priority	Summary	Rationale	Fit Criterion
REACTION-468	Functional - REACTION platform	Major	Provide regular update of data model for Health Care profile	Most application depends on current clinical data (e.g. blood glucose). A mechanism for regular data updates should be provided.	The Data Model for REACTION should provide a regular update mechanism for personal health care profiles.
REACTION-467	Functional - REACTION platform	Major	Semantics based data management	The monitoring and other data need to be properly annotated with ontological descriptions.	Relevant entries in the REACTION databases are annotated with semantic concepts.
REACTION-466	Functional - In-hospital pilot	Major	(Web) Service to present decision support for	After processing of data by the glucose prediction algorithm,	A service will be available to support

	application		glucose control to clinicians	the results should be presented by the system to the physician. The physician can use the result for decision support. The service uses data stored in the data storage and user additional user input as input for processing.	physician with glucose control of patients.
REACTION-465	Functional - In-hospital pilot application	Major	Clinical evaluation report	Supervision of glycaemia and associated treatment is performed once a day. The clinical evaluation report has to be produced daily. Adaptation of therapy or changes of medications has to be evaluated including by consultation with the duty-physician.	A daily clinical evaluation report has to be stored and available in the Inpatient application.
REACTION-463	Functional - In-hospital pilot application	Major	Context management for clinical (lab) values.	Contextualization of measured values (e.g. blood glucose values) is important in order to support REACTION applications like decision support. For example pre- or post-meal glucose values have very different meanings for treatment. Therefore the data management model has to provide context management.	The data management model support context management functionality for the inpatient prototype application.
REACTION-460	Functional - REACTION platform	Major	Measurements of HbA1c	The risk of developing diabetic complications is strongly affected by HbA1c. This parameter has to be measured every 2-6 months until the blood glucose level is stable on unchanging therapy in outpatient environment and at the patient enrolment in the inpatient environment (updates are decided	Specific fields have to be foreseen in data management.

				by clinicians).	
REACTION-459	Functional - In-hospital pilot application	Major	Ontologies and data management designed for the storage and multi-user availability of all relevant information, actions, treatments, events	Centrally managed data repositories shall be designed and implemented able to store and display (multi-user) all the relevant information for the diabetic patient management in the Inpatient environment.	Data insertion and/or update and data retrieval for patients shall be possible in multi-user way.
REACTION-458	Functional - Primary care pilot application	Major	Investigative stage	An investigative stage is required for all newly diagnosed diabetic patients. This stage (the duration of wich is determined by clinicians) is used to: confirm diagnosis, check effectiveness of lifestyle and medications, evaluate the optimal dosage of medications, carry out patient education , and reassure patients concerned about their blood sugar levels.	Specific fields have to be present in ontologies and data management.
REACTION-457	Functional - REACTION platform	Major	Privacy Enforcement Point	A component that could be added to the client side would be some kind of 'Privacy Enforcement Point'. Such a component could be examining outgoing data for information that the client did not authorize to be sent, yet. That is, the component would match the client's consents (with respect to the processing of her data) with the the kind of information from the outgoing message and, possibly, delay the transmission of certain information which the client has not decided on.	Privacy Enforcement Point is available for the REACTION client side.

				The component could stay hidden in other components for the time being, such as the Network Manager on the client side. The Privacy Enforcement Point should perform as a counterpart of the Consent Manager at the Reaction Device Hosting Server.	
REACTION-456	Functional - In-hospital pilot application	Major	Nutrition information has to be stored in the data management	Composition (proteins, fat and carbohydrates) of the meal has to be recorded and used for the insulin evaluation (the use of glycaemic index and load tables for various types of food might be taken into account). Also other parameters have to be taken into account (snacks in between, fasting, special diet, diarrhoea, vomiting, diminished/absence of appetite). Also special conditions related to nutrition have to be considered (PEG tube / parenteral feeding, fast adsorption of IV administered fluids).	The data management shall allow the insertion of time and composition of nutrition accompanied also by additional (context) parameters. The dosage of insulin shall vary with the variation of the nutrition.
REACTION-455	Functional - REACTION platform	Major	REACTION data storage	The REACTION platform should provide a storage module (database). Data gathered within REACTION should be stored here, as well as relevant data from external sources. The REACTION data storage should also use security mechanisms to include/exclude patient data access.	The REACTION platform provides a persistence layer for data storage with emphasis on data security and data access.
REACTION-454	Functional - REACTION platform	Major	Content formatter	A formatter for converting the acquired data to	Use a standard format or a verification

				useful information for the patient shall be available.	mechanism.
REACTION-453	Functional - REACTION platform	Major	Communication interface between REACTION Client and REACTION Server	A communication standard between REACTION client and server should be established (e.g. IHE-PCD01) in order to transport data from client to server side (and vice versa).	Communication interface between REACTION Client and REACTION Server will be available.
REACTION-451	Functional - REACTION platform	Major	In-hospital prototype communication with REACTION platform	The current design of the In-hospital prototype and the Primary care prototype does not consider the communication between these two prototypes (e.g. SOA). Thus, the data model should consider how the prototypes can be merged in future within the REACTION platform. A data/communication interface has to be defined.	Communication and transfer of data between In-hospital and Primary care prototypes are possible.
REACTION-449	Functional - Primary care pilot application	Major	Personalized care plan	A personalized care plan must be defined (and updated if necessary) for each patient. It includes disease management, risk management and lifestyle plan. Personalization methods must be defined.	Care plan can be personalized.
REACTION-448	Functional - REACTION platform	Major	Alert / notification messages should be short enough in order to be delivered as SMS messages if necessary	User's terminal mobile device will likely be used as a GSM mobile phone. Considering the advantages of Short Message Service (fast delivery, provides an alternative data path when an Internet connection is not available etc) the time critical messages for the patients should be	functional tests when user is away from broadband connection.

				able to be forwarded as SMS messages.	
REACTION-445	Functional - In-hospital pilot application	Major	Registration of specific interfering drugs (including their dosage)	Some drugs interfere with glycaemia management: systemic interference (e.g. cortisone by increasing blood glucose), analytical interference with glucose monitoring devices (e.g. fructose, maltose-interference). Their administration should be registered.	The data management shall allow for the insertion of specific interfering drugs (including their dosage).
REACTION-444	Functional - Primary care pilot application	Major	6-month clinical checks	Every 6 months the following tests have to be performed: blood tests as in the annual clinical checks (except for the thyroid function tests), BMI, blood pressure measurements, check smoking status, review of medications (including diet and lifestyle measures).	Specific fields (entries) have to be foreseen in ontologies and data management.
REACTION-442	Functional - Primary care pilot application	Major	Management of complications	Apart from the diabetic management, the other managements for diabetic patients will be around the complications (cardiovascular, renal, ophthalmology, management of foot and neuropathy problems).	Data management should include the necessary structures for assuring the storage of all necessary information for the management of complications.
REACTION-441	Functional - In-hospital pilot application	Major	Basic workflow in In-hospital environment	The basic workflow is based on measurement of blood glucose and evaluation of the necessary insulin (bolus or basal), based also on additional parameters and insulin administration.	There should be the possibility of acquiring, storing and retrieving all the information generated during any basic workflow performed during any time of the day/night.
REACTION-439	Functional - REACTION platform	Major	Information should be cashed in local storage to	In case of network error the client application should be able to store	The functional test should include specific tests in order to

			prevent loss in case of a node or communication failure.	temporary data. This will a) allow user to continue the process later and b) prevent corrupted / incomplete data to be uploaded to the main server.	ensure that there is no data loss in case of network failure.
REACTION-435	Functional - Primary care pilot application	Major	Outcomes of regular visits at primary healthcare centres	Outcomes of regular visits at the primary healthcare centre shall be registered through the data management.	The outcomes of each visit have to be stored as much as possible in a structured way.
REACTION-434	Functional - In-hospital pilot application	Major	Interface to Lab Information System (LIS) for glucose data import	In order to perform decision support, the blood glucose value has to be imported from the Lab Information System (LIS). A standardized interface from inpatient pilot application to the LIS has to be defined. HL7 would be a suitable standard.	Standardized Interface (e.g. based on HL7) to Lab Information System (LIS) for glucose data import.
REACTION-433	Functional - REACTION platform	Major	Results of screening, symptoms and types of diabetes or hyperglycaemia	At the diabetic patient enrolment his/her symptoms or results of screening confirming presence of diabetes should be registered. Symptoms can be: polydipsia, polyuria, blurred vision, weight loss, tiredness, recurrent skin infections. Results of screening can be: glucosuria or elevated BMs (both have to be confirmed with a diagnostic blood glucose measurement). Type of diabetes should be registered (if available data can be taken from the HIS/EPR).	Possible classifications should be present in the knowledge base & ontology and in the database fields for multiple selections from the classifications. Does the data need to be stored at each subsequent visit or evaluation?
REACTION-432	Functional - In-hospital pilot application	Major	Special examinations/treatments to be registered in fever chart	For some examinations/treatments in the hospital the patients have to be in a fasting and/or euglycaemic condition. Iin such cases treatment	These events (special examination/treatments) have to be registered in the data management where they can

				must therefore be adjusted to the particular needs (e.g. during fasting conditions the insulin dose is decreased). However a problem may arise if the patient has to wait longer than expected due to unforeseen delays. This may result in glycaemic excursions (hyper- or hypoglycaemia). The dose of insulin and/or OADs will therefore need to be adapted, the patient receives some food in the event of hypoglycaemia and receives insulin by injection in the event of hyperglycaemia.	be retrieved for the composition of the fever/sugar chart.
REACTION-430	Functional - REACTION platform	Major	REACTION Hosting client scheduler	TODO (Peter Rosengren)	TODO (Peter Rosengren)
REACTION-428	Functional - In-hospital pilot application	Major	Drug administration data (OAD and/or insulin)	Drug administration (time, insulin type, administration type - IV or SC-, dosage and other relevant information) has to be immediately registered in the data management by the administering nurse.	Data on drugs administered have to be stored in the data management where they can be also retrieved as part of the fever/sugar chart.
REACTION-426	Functional - REACTION platform	Major	Co-morbidities have to be registered	Co-morbidities are almost always present in diabetic patient and their presence can affect the overall management of the diabetic patient.	In the design of data management and ontologies the possibility of registering the co-morbidities with a basic set of attributes has to be guaranteed. Co--morbidities with their attributes have to be registered at the patient enrolment and at each subsequent visit or evaluation when new co-

					morbidities take place.
REACTION-425	Functional - REACTION platform	Major	Set of action rules	Action rules define what should be done if an event occurs, e.g. who should be notified and how.	Action rules can be defined and stored.
REACTION-424	Functional - REACTION platform	Major	Contextualization of measurements	The availability of all measurements (and mainly blood glucose levels) shall be accompanied also by the context of the measurements.	Measurements before any usage have to be contextualized.
REACTION-423	Non-functional - Operational	Major	Sensor quality parameters	The REACTION data management model should consider data storage for sensor quality parameters from devices reports like for example mis-calibration, or low battery. The parameters should be used for QoS.	Data fields for sensor quality parameters are available in the data management model.
REACTION-422	Functional - REACTION platform	Major	User transparency in case of communication failure	In case of network error the client application should be able to store temporary data (RDMM 76). The system should detect problems on the network and start the local storage. From the client's viewpoint, failures should be perfectly masked, and service should be completely fault-tolerant.	User transparency refers to a combination of user friendliness' and 'high efficiency'.
REACTION-419	Functional - REACTION platform	Major	Set of event rules	Event rules define the criterions of different events. Events are detected based on these rules. Personalization is possible through the use of individual thresholds and other parameters.	Event rules can be defined and stored.
REACTION-413	Functional - REACTION platform	Major	Connection with external services like MS HealthVault[1]	External interfaces to services of MS HealthVault should be taken into account in the REACTION platform.	Interfaces to MS HealthVault will be available.

[1] www.microsoft.com/en-us/healthvault

REACTION-410	Non-functional - Performance	Critical	Collecting measured data ("many to one" traffic pattern)	Different sensors can have different acquisition rates and relay data at different frequencies. Specific policy for data aggregation/fusion has to be defined.	Check the measurements collected by different sensors (times & values) and evaluate if there are critical delays.
REACTION-408	Functional - REACTION platform	Major	Non-pharmacological and/or pharmacological treatment	Non-pharmacological (diet, lifestyle, education) and pharmacological (OAD, insulin and interfering drugs) treatments have to be assigned to each patient and can be modified at each check.	In the ontologies and data management there should be the possibility of registering the different types of treatment for each patient and of modifying them at each check.
REACTION-404	Functional - REACTION platform	Major	Workflow Orchestration Manager	TODO (Peter Rosengren)	TODO (Peter Rosengren)
REACTION-402	Functional - In-hospital pilot application	Major	Measurements of blood glucose and insulin injections in In-hospital environment	In In-hospital environment, the blood glucose level measurements are, in most cases, performed by nurses with treatment performed by clinicians and/or nurses.	Measurements of blood glucose and insulin injections are tasks performed by clinicians and/or nurses. They have to store the relevant data in the system or to start the procedure for the storage of the relevant data in the system.
REACTION-401	Non-functional - Operational	Critical	Device specialization	Based on the necessary information to be monitored from the patient, a complete list of IEEE 11073 device specialization has to be completed. Measurements which cannot be collected using IEEE 11073 device specialization are also to be mentioned in this list. The complexity of the IEEE 20601 manager also depends on the number of device specializations to be	For each device the supported standard has to be specified (or the company documentation).

				managed.	
REACTION-399	Functional - Primary care pilot application	Major	Ongoing management	Ongoing management follows investigative stage. This stage is used to: support patients with difficulties in managing their diabetes, check effectiveness of lifestyle and medications, evaluate the optimal dosage of medications, perform patient education on diabetes, support changes in patient lifestyle, identify better diabetes management for patients.	Specific fields have to be present in ontologies and data management
REACTION-396	Functional - Primary care pilot application	Major	Consider patient's preferences, wishes and decisions	The data set should allow documentation of patient's preferences, wishes and decisions. This information should also be considered in the evaluation of rules etc., so that no recommendations against the will of the patient are made.	Patient's preferences, wishes and decisions can be documented and rules consider this data.
REACTION-395	Constraint - End-User Workplace Environment	Major	A REACTION application needs to be executed in the patient surgery independent from the EPR	As it is not possible to influence/modify many EPR systems, REACTION features inside the GP surgery have to be provided by a dedicated and independent application. This application communicates with - the REACTION platform over the Internet. - other systems in the surgery (EPR, lab, etc.) This application can be server-based and always on, for a prototype also an application client could be used.	An easy to run possibility to run and access REACTION features inside the GP surgery is available.

REACTION-393	Functional - Primary care pilot application	Major	Management of referrals to and responses from other physicians (via EHR interface)	Referrals are part of clinical pathways and treatment plan. Referrals should be documented and the recommendation of referrals should be considered in decision support rules... Summary letters and other "responses" from other healthcare professionals should be managed. - Optimal solution would be an interface to a regional or national EHR infrastructure (e.g. IHE-XDS) from where documents can be received.	Referrals can be documented and are considered in decision support, summary letters can be received via an appropriate data interface.
REACTION-389	Functional - Primary care pilot application	Major	Different stages for the patient management in primary care environment	Different actions have to be performed at different stages (newly diagnosed / medication titration / investigative stage, ongoing management) of patient management in primary care environment.	The data management has to allow the storage of the stage of management for each patient.
REACTION-388	Functional - REACTION platform	Major	Insulin sensitivity and insulin resistance	Insulin sensitivity and insulin resistance have to be used in the evaluation of the insulin dosage. However, these two parameters cannot be directly measured and have to be estimated by the clinicians. Their value can vary depending on the context (physio-psychological status of the patient, usage of specific drugs, etc.). Glucose control algorithm and physiology models should use these two parameters.	The data management has to allow for the insertion and subsequent modifications of these values by clinicians.
REACTION-387	Functional - REACTION platform	Critical	Information related to informed consent stored in the	An ethical approved declaration of informed consent has to be signed	The enrolment procedure shall allow the storage of the

			platform	(either digitally or in paper form) by patients before they can be enrolled in the REACTION platform.	digitally signed informed consent or of a scanned copy of the signed paper This procedure shall be completed before any other operation can be performed.
REACTION-386	Functional - REACTION platform	Minor	Medical knowledge base	Contains the relevant medical knowledge or is able to connect to external sources, e.g. evidences, drug information etc.	A medical knowledge base is built.
REACTION-384	Functional - REACTION platform	Major	Communication failure recovery	In case of communication failure, the connection has to be restored ASAP and the information should not be duplicated or corrupted.	Ensure that there is no data loss in the event of communication failure.
REACTION-383	Functional - Primary care pilot application	Major	Self-management and lifestyle support	Support of the patients' self-management by lifestyle (diet, exercise etc.) advices, therapy advices, health status assessment.	Self-management is supported.
REACTION-381	Functional - REACTION platform	Minor	Definition of a common ontology to refer to data, metadata, interfaces and models	A common ontology facilitates components integration and maintain a common language among the technical people and stakeholders.	All logical entities in software components should correspond to terms from the ontology (or to a published source which justifies their introduction).
REACTION-380	Functional - REACTION platform	Major	Set of alerts and reminders	A set of possible alerts and reminders. These can be thought as "prototypes". Action rules can define when and how they must be sent with which parameters.	Alerts and reminders can be defined and stored.
REACTION-378	Functional - Primary care pilot application	Major	Energy friendly data aggregation for mobile devices	Aggregation techniques should be used for reducing the overall data traffic to save energy.	The functional test should include specific tests on battery consumption

				Depending on the need for a real-time response, the redundancy of the data, etc., specific data-propagation strategies should be defined.	using different communication methods with the sensors.
REACTION-376	Non-functional - Security	Critical	Integrity check for the stored data	To guarantee the integrity of the stored data in the case of an unwanted happening.	Use of adequate methods like e.g. Hash keys or redundancy codes for the data stored.
REACTION-375	Functional - In-hospital pilot application	Major	Therapy scheme in In-hospital environment	Decision on therapy has to be performed immediately after performing any measurements based also on patient history and associated parameters. It might imply changes in the therapy scheme.	The pharmaceutical and non-pharmaceutical treatment (or therapy scheme) has to be stored in the data management and can be modified during any clinical evaluation of the patient. It has to be initialized immediately after the patient enrolment.
REACTION-374	Functional - Primary care pilot application	Major	Annual clinical checks	The annual clinical checks for the outpatient environment includes (with the necessary attributes): foot check, retinal screening (photograph of patient's retinae), test for protein, height and weight, BMI, blood pressure measurement, check smoking status, blood test (glucose level, HbA1c, etc.), check/administer flu injections, depression screening, review of medication (including diet and lifestyle measures).	Specific fields have to be present in ontologies and data management.
REACTION-372	Functional - REACTION platform	Major	Context of observations	The middleware of the REACTION platform should	The REACTION platform supports context

				support context management for observed values.	management on the client side.
REACTION-371	Functional - REACTION platform	Critical	Use of activity patterns for context annotations	Context has to be expressed synthetically in some way. A possible and common option is through activity patterns (to be specified for the two environments).	Collect measurements about physical activity, environmental data, additional information and evaluate the activity patterns verifying their correctness.
REACTION-369	Functional - In-hospital pilot application	Major	Storage of hyperglycaemic or hypoglycaemic episodes	Reasons for any cases of hypoglycaemia have to be registered (overdosing of insulin, change in nutrition, vomiting, changes in insulin sensitivity and/or resistance, etc.) and adequate treatment has to be provided and registered. Should the blood glucose level rise above a certain threshold, a hyperglycaemic episode has occurred. The reasons for such an episode have to be registered along with ensuing changes in treatment..	Specific procedures have to be present for the management of hyperglycaemic or hypoglycaemic episodes. These procedures shall also allow for the recording of the significant parameters and actions.
REACTION-367	Functional - Primary care pilot application	Major	Insertion of baseline physiological measurements at the first visit	At the first visit, baseline physiological measurements (the exact set has to be clearly defined) have to be inserted in the platform.	The data management shall foresee the possibility of introducing the baseline physiological measurements at the first visit (just after the patient enrolment).
REACTION-365	Functional - Primary care pilot application	Critical	Data should be stored in proper way in order to be easily transmitted over mobile networks in case that broadband network is not	In the event that the hosting client is not connected through a broadband connection, the patient will be able to upload data using GPRS / 3G data networks. In this	Functional test uploading data over slow mobile networks.

			available.	case we need to examine possible limitations.	
REACTION-363	Functional - In-hospital pilot application	Major	Interface to Hospital Information System for clinical data import/export	In order to exchange clinical data between In-hospital pilot application and Hospital information System (HIS) an interface based on HL7 has to be provided.	Standardized Interface (HL7) to HIS / EPR to exchange clinical data.
REACTION-362	Functional - In-hospital pilot application	Major	Interface to patient demographic register	In order to import demographic data from the patient demographic register has to be imported from the HIS. A standardized interface e.g. HL7 has to be used for data interchange. Required data fields are: - unique PID - name - age (data of birth) - sex - address	Standardized interface (HL7) to patient demographic register is available for the In-hospital pilot application
REACTION-361	Functional - REACTION platform	Major	Baseline and clinical history handled in the data management	Immediately after patient recruitment, his/her baseline and clinical history has to be entered in the platform. This can be done by extracting this information from the HIS/EPR (if available and interoperable) and completing manually (through a proper UI) the missing information.	The data management should allow the storage of baseline and clinical history and these data can be extracted from the HIS/EPR (if available and interoperable).
REACTION-358	Functional - REACTION platform	Major	Network manager for hosting client	TODO (Peter Rosengren) incl. security mechanism ("the Network Manager would be configured to encrypt the data") "The LinkSmart Network Manager has two roles, it takes care of the P2P between different nodes. It also keeps a list of	TODO (Peter Rosengren)

				LinkSmart Identifiers for different devices/services and there local endpoints. In this way it "virtualizes" devices, services, and applications behind identifiers."	
REACTION-357	Functional - Primary care pilot application	Major	Power management techniques to decrease power consumption	Power management techniques can be used to decrease the power consumed by sensors. Some non-critical sensors can power down when activity is not required, waking up in time to receive and transmit messages as necessary.	The functional test should include specific tests in order to ensure that power consumption is at an acceptable level.
REACTION-356	Non-functional - Usability	Major	Manual data insertion	In case of no connectivity with the sensor or medical device or use of a non-interoperable medical device, the mobile device should offer the possibility of manual insertion of measurement data .	Check that measurements can be inserted manually using the mobile device .
REACTION-355	Functional - REACTION platform	Major	Computer interpretable guidelines	Evidence based guidelines as important constituents of the knowledge base must be encoded in a computer-interpretable way for decision support.	Guidelines are encoded.
REACTION-352	Non-functional - Maintainability and portability	Major	Scalable / easy to use solution for REACTION software in GP surgery	The REACTION software which is executed in the GP surgery has to be usable for practices in different setting with different EPR systems. It should provide a user interface for disease management as well as Web Services which can be implemented by EPR manufacturers to easily integrate REACTION features into their products.	REACTION software is easy to run beside an EHR application or EHR manufacturer is satisfied with ease of integration of REACTION

REACTION-351	Functional - Primary care pilot application	Major	Telemonitoring data should be visualized to patients and professionals in a flexible and performant way	GPs and nurses as well as patients and their carers use the telemonitoring data to get an impression of the patient status. So telemonitoring data needs to be visualized in a flexible way (aggregation level, combination of parameters ...) Data has to be handled in a way that this visualization can be generated on-demand with good performance.	Data can be visualized flexibly and with good performance to professionals
REACTION-349	Non-functional - Usability	Major	Patient questionnaires (lifestyle, physio-psychological conditions, checking medication compliance, adherence to clinical pathways, education, self management)	Questionnaire for patients in order to collect qualitative (or quantitative but not directly measurable) information related to the parameters to be monitored has to be available. They are part of the monitoring (using a frequency set) administered by the responsible clinician.	The mobile device shall have user interfaces allowing completion of these questionnaires.
REACTION-348	Functional - REACTION platform	Major	High-level data fusion	Besides low-level data fusion on the client side a high-level data fusion should be available for the REACTION platform. The high-level data-fusion should provide the integration of external gathered information to the REACTION platform data structure and the fusion of REACTION-internal processed data.	High-level data fusion functionality will be available for the REACTION hosting server.
REACTION-347	Non-functional - Operational	Major	Continuous blood glucose monitoring	Using the acquired values, the mobile device must be able to analyze the glycaemic variability and to generate alarms or alerts (hypo or hyper), based on configurable	This functionality can be tested using the device simulator and simulated sequences of values-

				thresholds.	
REACTION-345	Functional - REACTION platform	Major	Two-way communication between REACTION server and client	There is a need for two-way communication between server and client e.g. for remote configuration of the end-user application running in the AHD. The data fusion engine also needs to be configured based on which values the clinician wants to observe. There is also a need for 2-way communication from the point of view of error handling. If the observed values suddenly appear out-of-range it might be necessary to check with the client if this is an error state. Other devices/sensors, e.g. the Continua devices, might also require different types of communication. It might be necessary to reverse a patient's consent that had to be given 'remotely', e.g. at the doctor's surgery, because the hosting client at the patient's home is simply a 'box' with no display or input capabilities. In this restricted 'boxed case', it would be hard to change the patient's privacy settings, once they are initially configured, if we were unable to push data back to the box.	Two way communication between Client and Server will be available for the REACTION platform in order to perform: e.g. data fusion configuration, error-handling, data security (consent management).
REACTION-344	Non-functional - Look and feel	Major	Display of acquired measurements (values, time, context (if available))	Provide immediate and consistent (if possible also contextualized) information to the patient.	The user interface on the mobile device shall have this functionality.
REACTION-	Functional -	Major	Low-level data	The REACTION	Low-level data

342	REACTION platform		fusion	platform should support low-level data fusion in order to interpret measurements occurring in PAN. The Data Fusion needs to take place close to the patient/user.	fusion will be available for the REACTION platform (middleware).
REACTION-340	Functional - REACTION platform	Major	Storage of insulin administration	Insulin administrated to patient has to be stored with time, dosage (units), type of insulin and modality of administration (always subcutaneous for outpatient environment).	Specific fields have to be foreseen in data management, ontologies and user interfaces (also portable).
REACTION-338	Non-functional - Security	Critical	All data entered must be checked for format, consistency and validity	Unintended user actions should not harm data integrity and the overall functioning of the platform. In case of doubt, the user must be warned and asked how to proceed. The user must be able to correct mistakes easily.	The functional test should include specific tests in order to verify such circumstances.
REACTION-336	Functional - REACTION platform	Major	Patient enrolment (or recruitment)	When an interoperable HIS or EPR is present in the managing organization, the patient data at the patient enrolment should be obtained from the HIS or EPR through interoperable user interfaces.	When an interoperable HIS/EPR is present, a new diabetic patient cannot be created in the REACTION platform if not present in the HIS/EPR. When a diabetic patient is created, his/her data have to be taken from the HIS/EPR.
REACTION-334	Functional - Primary care pilot application	Major	Devices should be able to operate anywhere in the home	To make a system that is ubiquitous and fits patient lifestyle	Device specification
REACTION-333	Functional - Primary care pilot application	Major	Devices should be single communication technology	Single communication technology will reduce cost of end	Device specification

				system and simplify use for end user	

4.6.3 Updated requirements

No requirements have been updated

4.6.4 Deleted requirements

REACTION-10, REACTION-11, REACTION-13, REACTION-16, REACTION-21, REACTION-27, REACTION-36, REACTION-137, REACTION-164, REACTION-167, REACTION-260, REACTION-368 and REACTION-377 have been resolved or closed as being Out of Scope.

4.7 Lessons Learned in WP5

The RTD work undertaken in WP5 is related to network management and service execution. FORTHNET is the WP leader and six Lessons Learned have been collected and validated from this WP.

Org. No.	Experience and knowledge gained	Lesson Learned	Requirement affected
FORTH-ICS 1	Delays in the final selection of the medical devices to be used in the primary care site generates delays in the integration phase and may produce several alternative (but not optimal) BAN/PAN to be used in the primary care prototype.	A clear identification of the medical devices to be used should be performed sufficiently in advance in order to allow at least a purchase of a single sample and the execution of all the integration tests sufficiently before the release of the prototype.	
FORTH-ICS 2	There are very few medical devices enabled with the Continua protocol. About glucometers (we must focus on diabetes management and not on comorbidity management) only the Roche Smart Pix Device adapter is available. Thus, from one side the use of standards should be promoted, but on the other side the very limited amount of medical devices compliant with the standard is discouraging.	The consortium (and mainly the involved clinical partners) should take a definite decision quite early in each iteration cycle about the medical (and other) devices to be used in the prototype under development in that iteration cycle. That would definitely help in more focused development and tests.	
FORTHNET 3	The Network Monitoring System for PC clients using Windows OS has been developed during the second year, but the specific software has been carefully researched and developed, in order to easily provide additional support for other operating systems, apart from Windows, through minor software modifications. The main purpose was not to limit the OSs of the PC clients.	Since major parts of the REACTION platform are being developed in a specific way (web-services), in order to provide its functionalities through various OSs (OS independent), it was crucial to develop the NMS client accordingly and to provide support for various OSs through minor modifications of the NMS client Windows edition.	
FORTHNET 4	The templates that can be provided through the Network Monitoring interface have been developed in order to support and help end-users of the software that have limited or	During the development of the NMS interface, it was important to consider that the software was not going to be used only by IT professionals and network	

	no administrator expertise. The main visual aids provided through the software are the graph templates, where the end user can easily detect and diagnose network problems, by observing graphs.	administrators, leading to the development of graphical representations for additional support for the non-expert end users.	
FORTHNET 5	Following the development of the NMS, a number of tests had been performed in order to examine the functionalities and the general operability and implementation of the system. Network related alert notifications system was also introduced to the overall implementation, where, during the test period, it was noticed that the majority of the generated alerts were related to network availability.	Since the most common alerts are related to network availability, it was decided that the process for the network alert generation should be fully automated, in order to provide even more transparent end user interaction.	
FORTHNET 6	Throughout the development of the NMS system (up to the second year of the project), a more general approach related to the software implementation was adopted, in order to cover as many scenarios as possible, regarding mainly the possibilities of software and hardware components to be used in the reaction platform, especially related to the servers. In the late months of the second year of the project it was finally clear that this approach, regarding the NMS system was not necessary or even ideal. As a result, a single-server approach has been adopted, saving crucial time for the NMS development.	The general approach adopted for the NMS development has caused some drawbacks related to time constraints, since it was not necessary to consider the design of a multiple server environment for REACTION. The related requirements of the project were stating that a general approach regarding the server distribution was not necessary. Despite the lost amount of time, it was stated that all requirements and reports of the project should be considered in a more careful way.	

4.8 Change request and re-engineering originating from WP5

This Section provides an analysis of the six Lessons Learned from the work performed in WP5 in the second cycle. No requirements were added or updated, while two were deleted.

4.8.1 Analysis of Lessons Learned

The first two Lessons Learned in WP5 are related to the medical devices to be used in the REACTION framework and more specifically to time delays in their selection as well as their compliance with the Continua protocols.

The remaining four Lessons Learned are related to the Network Monitoring System (NMS).

LL WP5-3 refers to the additional support of the NMS PC client to other Operating Systems through minor modifications, apart from the already developed client for Windows OS.

LL WP5-4 is related to end-user friendliness and ease of use, considering users that have limited or no administrator expertise, in order to easily detect any network problems.

LL WP5-5 refers to the changes that took place during the development and testing of the NMS generated alerts, where it was discovered that most of them are related to network availability. As a consequence, it was decided that the process for the network alert generation should be fully automated in order to provide even more transparent end user interaction.

Finally, LL WP5-6 regards the REACTION single-server approach that was adapted in the development of the NMS, since it was not necessary to consider the design of a multiple server

environment for REACTION. This alteration has resulted in some minor delays in the development of the NMS, without causing any drawbacks.

4.8.2 New requirements

No requirements have been added.

4.8.3 Updated requirements

No requirements have been updated.

4.8.4 Deleted requirements

REACTION-19 and REACTION-257 have been resolved or closed as being Out of Scope.

4.9 Lessons Learned in WP6

The RTD work undertaken in WP6 involves risk assessment and feedback. MSG is the WP leader and seven Lessons Learned have been collected and validated from this WP.

Org. No.	Experience and knowledge gained	Lesson Learned	Requirement affected
MSG 1	Constraints for the implementation of a dosing protocol into an electronic system are time consuming.	Transformation of paper-based dosing protocol in an electronic workflow and decision support system demands for intensive study of constraints. The identification and translation into technical needs is time costly and demands for an interdisciplinary team.	GMSIP[2]-28
MSG 2	Adaption of a specific protocol (e.g. for a different patient group) needs much effort for a generic and modular workflow support.	The current implementation needs effort to separate the protocol from the workflow support (mainly at the user device). Moreover, actions have to be taken to provide a flexible workflow configuration system.	GMSIP-123
MSG 3	The current developed REACTION insulin dosing protocol is basically not flexible in order to fit it without effort to other medical wards.	The adaption of the REACTION protocol to other medical fields than the general ward demands for separate effort in terms of workflow analysis, dosing protocol pretesting and clinical trials.	
FORTH-ICS 4	The data collection protocol of the DCCT study has been carefully evaluated, for better understanding the structure and the semantic of the DCCT data.	The DCCT data can be employed for deriving several long term risk assessment models. However, since the DCCT data were collected in North America around 20 years ago, the applicability of these models in nowadays European populations is questionable.	
FORTH-ICS 5	Six different Long Term Risk Assessment models are derived from the DCCT data. Each model is	In order to derive a meaningful model, it is necessary to carefully define the complication	

[2] The GMSIP (Glucose Management System In-hospital Prototype) notation refers to a separate development project in JIRA

	designed to evaluate the risk of a specific diabetes related complication.	under study in terms of the parameters that are available in the DCCT dataset.	
FORTH-ICS 6	The long terms risk assessment model for the Retinopathy complications was embedded within a Web Service interface. A simple Web Based Interface was developed for the Retinopathy complication for testing and demonstration purposes.	Web Services seem to be quite effective in order to allow the integration of the long term risk assessment models in the REACTION risk assessment engine. Building a simple Web-based Interface is an effective strategy in order to illustrate the operation of the models to any audience that does not known Web Services technical aspects.	
FORTH-ICS 7	A uniform approach for the development of the risk models should be applied and this approach should be in-line with the main guidelines for software developments agreed in the consortium.	A web service approach has to be applied for the implementation of the risk models in order to allow their easier integration in the higher level shell (risk assessment and decision support systems).	

4.10 Change request and re-engineering originating from WP6

This Section provides an analysis of the seven Lessons Learned from the work performed in WP6 in the second cycle. The work resulted in 10 requirements being added and five deleted. No requirements were updated.

4.10.1 Analysis of Lessons Learned

The Lessons Learned focus on two different aspects of the work in work package 6. LL WP6-1 to LL WP6-3 address the difficulty of transforming the insulin dosing protocol – which is usually used in paper form in clinical practice – in a computerized system meeting the requirements of users and especially the very strict regulations of the Medical Device Directive. Moreover, the provision of a flexible, configurable workflow support system is another important issue of the project. The addressed problems can be solved with additional effort but in fact result in a secure, safe and comfortable way for clinical end users (nurses, doctors) to have a reliable decision support system in their daily work of finding the optimal insulin treatment of diabetic patients with diabetes type 2.

LL WP6-4 through LL WP6-7 deal with the second aspect, which focuses on the long-term risk modelling based on publicly available diabetes data sets. These data sets are available under some pre-conditions but their usage for risk modelling is only possible with technical restrictions due to the specific patient cohort or the age of the dataset. These factors have to be taken into account when designing end user applications based on modern user interfaces for risk prediction and patient communication.

4.10.2 New requirements

10 new requirements have been added as a consequence of the consolidation described in LL WP2-2.

Key	Requirement Type	Priority	Summary	Rationale	Fit Criterion
REACTION-421	Functional - REACTION platform	Major	Models and rules for insulin dose prediction (In-hospital)	A physiologic model and calculation rules/algorithm must be stored for insulin dosing support based on clinical protocols.	Necessary models and rules are defined and stored.
REACTION-409	Functional - Primary care pilot	Major	Risk assessment models and	Models and rules must be defined to determine personal risks.	Models and rules for risk assessment are

	application		rules		present.
REACTION-392	Functional - Primary care pilot application	Major	Personal Health Status Profiles	Personal Health Status Profile for each patient must be generated, stored and regularly updated. It serves as an input for risk assessment and disease management.	Personal Health Status Profiles can be generated.
REACTION-391	Constraint	Major	Data fields for the In-hospital glucose control prototype (eDSS).	Following data fields should be provided: - administrative data (patient name, address, PID, ward, hospital bed, physician(s) in charge, nurse(s) in charge) - demographic data (age, sex, date of birth) - medical history (type of diabetes, medication, comorbitities, former complications, pre-existing conditions) - anamnesis data (fever, infections, diarrhea, vomiting, hypo-hyperglycemia) - lab data (glucose level, HBA1c, ...) - external input (food intake, insulin sensitivity, ...) - context data (time of glucose measurement, what device, ...)	Required data fields will be provided by data structure.
REACTION-364	Functional - REACTION platform	Minor	Case base	The case base contains a set of cases generated in the platform and/or imported from existing case bases. It can be used together with other knowledge elements (e.g. evidences) to discover new knowledge.	A case base is present.
REACTION-360	Functional - REACTION platform	Major	Mechanistic model and rules for insulin dose prediction (primary care)	A physiologic model and calculation rules/algorithm must be stored for insulin dosing support.	Necessary models and rules are defined and stored.
REACTION-353	Functional - REACTION platform	Minor	Case generation	From the data of individual patients, a depersonalized case description is built which will be put in the case base.	Cases can be generated.
REACTION-346	Functional - REACTION	Major	Knowledge Discovery	EPRs often contain unstructured text	REACTION provides a

	platform		from unstructured clinical text information	information. In order to use this information for decision support or diabetes management the information has be pre-processed. NLP-technologies to find relevant information for REACTION applications from these data bases (annotation of text information: anamnesis information, co-morbidities, medical history, ...) can be a useful tool.	knowledge discovery module to process unstructured information and store this information in the data storage for further processing.
REACTION-337	Functional - Primary care pilot application	Major	Health status model	The health status model serves as a generic prototype for Personal Health Status Profiles, i.e. defines its data content. This helps to define personal models (profiles), which permit the personalised disease management.	A health status model is present.
REACTION-321	Non-functional - Operational	Major	Risk analysis	Risk Analysis has to be started in the very early stage of the development. The identified risks have to be identified and assessed.	All risks must be in an acceptable range according to the assessment criteria.

4.10.3 Updated requirements

No requirements have been updated.

4.10.4 Deleted requirements

Due to findings during the research and development process (strict requirements of the MDD) but also as the outcome of meetings with domain and medical experts, requirements REACTION-379, REACTION-417, REACTION-418 and REACTION-429 have been resolved as being Out Of Scope, while REACTION-359 was resolved as being not implementable,

4.11 Lessons Learned in WP7

The RTD work undertaken in WP7 revolves around security, safety and privacy issues. FHG-SIT is the WP leader and one Lesson Learned has been collected and validated from this WP.

Org. No.	Experience and knowledge gained	Lesson Learned	Requirement affected
FHG-SIT 1	The EU Privacy Directive 95/46/EC permits processing of medical data when it is being done by health professionals for the purpose of diagnosis, preventive medicine, or treatment – see Article 8.3. The scenarios and use cases of REACTION fall within these	Consents are not required for the REACTION use cases.	R-370, R-398, R-407, R-412

	categories. Since the processing is explicitly allowed by the Directive, the patient's consent is not required.		

4.12 Change request and re-engineering originating from WP7

This Section provides an analysis of the Lesson Learned from the work performed in WP7 in the second cycle. The work resulted in the addition of 16 new requirements and the deletion of nine requirements. No requirements were updated.

4.12.1 Analysis of Lessons Learned

LL WP7-1 deals with the issue of consent. The EU Privacy Directive 95/46/EC in general forbids the processing of personal data under Article 8.1. However, there are a number of derogations from this general prohibition. One of these derogations is if the data subject, such as a patient, gives her/his consent to the processing of her/his data, see Article 8.2 (a). Since the REACTION platform will process personal, medical data of patients, it was expected that managing patient consents would be necessary. However, after the scenarios and use cases of REACTION became clearer, it turned out that the processing being done in REACTION is permitted by another derogation, namely Article 8.3, which says that a patient's consent is not required

"where processing of the data is required for the purposes of preventive medicine, medical diagnosis, the provision of care or treatment or the management of healthcare services, and where those data are processed by a health professional subject under national law or rules established by national competent bodies to the obligation of professional secrecy or by another person also subject to an equivalent obligation of secrecy."

Therefore, the requirements concerning patient consent have been deleted.

4.12.2 New requirements

16 new requirements have been added as a consequence of the consolidation described in LL WP2-2.

Key	Requirement Type	Priority	Summary	Rationale	Fit Criterion
REACTION-452	Non-functional - Security	Major	Communication between the Reaction Device Hosting Server and the EPR/EHR System MUST be authentic (entity authentication), with integrity, and confidential.	It must be assumed that data transmission from the Reaction Device Hosting Server to the EPR/EHR System and vice versa takes place over an insecure channel, i.e., data might be overheard or tampered with. Since personal data is to be transmitted it MUST be ensured that the communication channel is authentic, with integrity, and confidential.	Availability of mechanisms to provide communication channels with authenticity, integrity, and confidentiality.
REACTION-438	Non-functional - Security	Major	Communication between the Reaction Device Hosting Server and the GP EPR MUST be authentic (entity	It must be assumed that data transmission from the Reaction Device Hosting Server to the GP EPR and vice versa takes place	Availability of mechanisms to provide communication channels with authenticity, integrity, and

			authentication), with integrity, and confidential.	over an insecure channel, i.e., data might be overheard or tampered with. Since personal data is to be transmitted it MUST be ensured that the communication channel is authentic, with integrity, and confidential.	confidentiality.
REACTION-437	Non-functional - Security	Major	Each role MUST be assigned to a set of permissible actions.	Since some actions are reserved for specific roles it has to be decided which actions are permissible for which role.	According to the roles' needs, each role is assigned to a set of appropriate permissions.
REACTION-431	Non-functional - Security	Major	Data/messages exchanged between the Reaction Device Hosting Server and the GP EPR SHOULD be authentic (message authentication), with integrity, and confidential.	The security of messages transferred between the Reaction Device Hosting Server and the GP EPR must be ensured even after the message was received - this is true even if the message was received over a secure communication channel. To guarantee this, the messages themselves MUST be self-contained with respect to authenticity, integrity, and confidentiality.	Availability of mechanisms to provide data authenticity, integrity, and confidentiality
REACTION-415	Non-functional - Security	Major	Each person MAY only perform actions permitted by her role.	Before a requested action is performed, a control mechanism has to check whether the requested action is part of the requester's set of permissible actions according to its role.	Availability of a control mechanism which decides whether a requested action may be granted or denied according to the requester's role.
REACTION-414	Non-functional - Security	Major	Communication between the Reaction Hosting Client and the Reaction Device Hosting Server MUST be authentic (entity authentication), with integrity, and	It must be assumed that data transmission from the Reaction Hosting Client to the Reaction Device Hosting Server and vice versa takes place over an insecure channel,	Availability of mechanisms to provide communication channels with authenticity, integrity, and confidentiality.

			confidential.	i.e., data might be overheard or tampered with. Since personal data is to be transmitted it MUST be ensured that the communication channel is authentic, with integrity, and confidential.	
REACTION-403	Non-functional - Security	Major	Each entity in the Reaction platform MUST be representable by a digital identity.	In the Reaction platform, entities must be uniquely identifiable and recognisable in order to allow repeated communication, referrals, accountability of actions, exclusion of ill-behaving entities, etc.	Availability of a digital identity mechanism.
REACTION-400	Non-functional - Security	Major	Data/messages exchanged between the Reaction Device Hosting Server and the EPR/EHR System SHOULD be authentic (message authentication), with integrity, and confidential.	The security of messages transferred between the Reaction Device Hosting Server and the EPR/EHR System must be ensured even after the message was received - this is true even if the message was received over a secure communication channel. To guarantee this, the messages themselves MUST be self-contained with respect to authenticity, integrity, and confidentiality.	Availability of mechanisms to provide data authenticity, integrity, and confidentiality
REACTION-385	Non-functional - Security	Major	Digital identities for the Reaction platform MUST only be issued or revoked by trusted (third) parties, e.g., a certification authority (CA).	Without a trusted party (TP), anyone could produce its own digital identity and someone relying on such an identity would have to trust that the claimed identity is genuine. By incorporating a TP, relying parties trust that the TP ensures that its issued digital identities are	Availability of a party which is trusted to orderly issue and revoke digital identities.

				genuine. This makes life easier for relying parties as they only have to establish a single trust relationship (with the TP) as opposed to having a multitude of trust relationships with others. The same goes for parties that had been excluded from the Reaction platform, as each relying party would have to determine by itself if another party is still part of the Reaction platform or not. In case of a trusted party, the relying part could simply query the TP if some identity is still valid or had been revoked, e.g., because its owner left the platform.	
REACTION-382	Non-functional - Security	Critical	Privacy enhancing technology	Protect the privacy of users personally identifiable information (PII) and further more personal data.	It must not be possible for any third party to determine the relation between a measurement and the measured patient's real world identity.
REACTION-354	Non-functional - Security	Major	Data/messages exchanged between the Reaction Host Client and the Reaction Device Hosting Server MUST be authentic (message authentication), with integrity, and confidential.	The security of messages transferred between the Reaction Host Client and the Reaction Device Hosting Server must be ensured even _after_ the message was received - this is true even if the message was received over a secure communication channel. To guarantee this, the messages themselves MUST be self-contained with respect to	Availability of mechanisms to provide data authenticity, integrity, and confidentiality

				authenticity, integrity, and confidentiality.	
REACTION-343	Non-functional - Security	Major	Every person represented in the Reaction platform MUST be assigned to one or more roles.	In order to interact with the Reaction platform, persons need certain rights. As rights are associated with roles, persons MUST have at least one role to interact with the Reaction platform.	Each person is assigned to at least one role.
REACTION-341	Non-functional - Security	Major	Roles MUST be defined for stakeholders of the Reaction platform, e.g., doctor, nurse, patient, informal carer, administrative personnel etc.	Each person in the Reaction platform has the right to perform a certain set of actions. In order to simplify the administration of these rights, each person is assigned to a role and roles are assigned to permissible actions. The advantage of this approach is that it is easier to manage the rights of a role than managing individual rights for each person.	Roles are defined for every actor from the Reaction use cases.
REACTION-339	Non-functional - Security	Major	Communication between the Reaction Device Hosting Server and the patient's/GP's web browser MUST be authentic (entity authentication), with integrity, and confidential.	It must be assumed that data transmission from the Reaction Device Hosting Server to the patient's/GP's web browser and vice versa takes place over an insecure channel, i.e., data might be overheard or tampered with. Since personal data is to be transmitted it MUST be ensured that the communication channel is authentic, with integrity, and confidential.	
REACTION-324	Functional - Primary care pilot application	Major	Providing a secure log in and log out for the user	The system shall be protected with a secure login for each user on the web portal, users shall be required to log out upon the end of the	

				task. The system shall have a clear hierarchy for different type of users (Patient, Clinic, etc) and each user logging into the system shall be logged into the correct user type.	
REACTION-323	Functional - Primary care pilot application	Major	Providing a complete audit trail for each user's data and action taken on the system	There must be a complete audit trail of all actions taken in the system by any user. No user shall have the permission to permanently delete data from the system. This refers to the system logging and all actions taken by different users. The system shall also provide traceability of each action to the user taken those actions.	

4.12.3 Updated requirements

No requirements have been updated.

4.12.4 Deleted requirements

Requirements REACTION-102, REACTION-103, REACTION-370, REACTION-373, REACTION-398, REACTION-407, REACTION-412, REACTION-429 and REACTION-464 were closed with resolution Out of Scope, as patients' consents need not be dealt with for the purpose of REACTION.

4.13 Lessons Learned in WP8

The RTD work undertaken in WP8 relates to clinical practice and associated field trials. MUG is the WP leader and five Lessons Learned have been collected and validated from this WP.

Org. No.	Experience and knowledge gained	Lesson Learned	Requirement affected
MUG 1	The number of incorrect calculated insulin doses of the first general ward clinical field trial occurs more often than expected.	The electronic decision support should counteract wrong calculations. Based on that the quality of correct insulin doses will be improved.	
MUG 2	Preliminary conclusions of the first clinical field trial have shown that the blood glucose values at midday of some patients are often not in the recommended target range.	The REACTION algorithm has space for improvement, especially to optimize the blood glucose values during lunch time.	
MUG 3	Preliminary conclusions of the first clinical field trial have shown that there were glucose measurements	A data analysis has to be performed to find the cause of low glucose measurements.	

	below 70 mg/dl in the first general clinical field trial.	Based on these findings preventive interventions to avoid low blood glucose values have to be implemented or a change of the algorithm has to be done.	
MUG 4	The continuous glucose data of the iPro2 sensor of the first general clinical field trial had no influence on the treatment, because the data were analysed retrospectively.	The results of continuous glucose data should be used during the treatment period in order to enable optimized interventions for safe glycaemic control assuming that the continuous data are reliable.	
MUG 5	The analysis of the standardized online survey of in hospital glucose management of nurses has shown that no standardized operating procedures of glycaemic control are defined and that different treatment approaches are used.	The REACTION in hospital application and the REACTION algorithm should support standardized procedures (target ranges, therapy adaption, correctional schema, etc.) to improve glycaemic control.	

4.14 Change request and re-engineering originating from WP8

This Section provides an analysis of the five Lessons Learned from the work performed in WP8 in the second cycle. No requirements were added, updated or deleted.

4.14.1 Analysis of Lessons Learned

These five Lessons Learned are gained from the first general ward clinical field trial. The aim of the study is to investigate efficacy, usability and safety of an enhanced version of the REACTION algorithm to control glycaemia in hospitalised patients with diabetes Type 2 and to compare to results obtained with standard care.

LL WP8-1 reports a preliminary result of the first general ward clinical field, i.e., that the number of incorrect calculated insulin doses occurs more often than expected. Therefore the electronic decision support should counteract wrong calculations.

LL WP8-3 detected that blood glucose values below 70 mg/dL were measured. Therefore preventive interventions or a modification of the algorithm has to be done after data analysis.

LL WP8-4 concerns the continuous glucose monitoring. The iPro2 sensor is inserted at the beginning of the treatment period and removed at the end of the treatment period of the first clinical trial. The continuous glucose data are analysed retrospectively and no treatment is based on the data. The results of continuous glucose data should be used during the treatment period in order to enable optimized interventions for safe glycaemic control assuming that the continuous data are reliable.

LL WP8-5 was observed at the beginning of the first clinical trial in connection with a standardized anonymous online survey about in hospital glucose management and its needs and problems, which was performed in August 2011 at the Medical University of Graz. The nurses of two wards (Endocrinology and Cardiology) were invited to participate. The analysis has shown that no standardized operating procedures of glycaemic control are defined and that different treatment approaches are used. This outcome confirms the need of the electronic decision support system.

The five Lessons Learned in the first clinical field trial will be taken into account in the development of the REACTION in-hospital application and for the optimisation of the REACTION algorithm to achieve safe glycaemic control.

4.14.2 New/updated/deleted requirements

No requirements have been added, updated or deleted.

4.15 Lessons Learned in WP9

The work undertaken in WP9 relates to the socioeconomic framework of the REACTION project. VUB is the WP leader and 12 Lessons Learned have been collected and validated from this WP.

Org. No.	Experience and knowledge gained	Lesson Learned	Requirement affected
VUB 1	Visibility of components and electronic visibility of network components can lead to stigmatization of diabetes patients.	The visibility of those components should be decreased to a minimum to avoid stigmatization. REACTION should therefore try to develop components with a minimal visibility.	R-470
VUB 2	The possibility to socialize with others having a similar condition (and suffer stigmatization because of it) is normally appreciated.	REACTION should not eliminate this possibility. Eventually, REACTION could reinforce this possibility by using new ways of communication (social networks).	R-471
VUB 3	The feeling of stigmatization can be an important inhibitor in the uptake of products like a portable sensor patch.	The visibility of those devices should be as reduced as technically feasible.	R-472
VUB 4	The issue of visibility and stigmatization do not only apply to portable devices but also to the public networks used in REACTION.	The visibility of those networks should be as low as possible. Furthermore, the highest feasible level of security should be implemented.	R-473
VUB 5	Personal data can be leaked or lost	Data Breach Notification mechanism should be incorporated in REACTION architecture	R-474
VUB 6	The data controller should implement appropriate organizational and technical measures to ensure to protect personal data against unauthorized disclosure or access.	Log and log-in system are needed to identify and authenticate persons accessing the system and in order to keep records who did what and when in an audit log.	R-475
VUB 7	Data protection impact assessments are to be carried only in when data processing operations 'are likely to present specific risks to the rights and freedoms of data subjects by virtue of their nature, their scope or their purposes.	Data protection impact assessment should be envisaged	R-476
VUB 8	Data controllers should be held liable for damages due to unlawful processing and the user/patient should receive prompt and adequate compensation	Rules of liability for unlawful processing should part of the contract agreement between health service provider and patient. In case of damage, compensation should be adequate.	R-477
VUB 9	Consent procedures should be reviewed over time with the involvement of carers. The patient should not be left alone with technology	The agreement of the patient of having data rating to him/her being processed should be assessed over time and involving carers on a regular basis	
VUB 10	REACTION must meet the Requirements of the Medical Device	Standards can be used to demonstrate that the essential	

	Directive	requirements have been met.	
VUB 11	REACTION should consider the reimbursement requirements for acts of eHealth.	Some states do not as of yet reimburse acts of health. Others do but have specific requirements.	
VUB 12	REACTION should take into account the cultural variations in patient preferences that exist.	Significant cultural variations exist in patient treatment preferences.	

4.16 Change request and re-engineering originating from WP9

This Section provides an analysis of the 12 Lessons Learned from the work performed in WP9 in the second cycle. The work resulted in the addition of eight requirements. No requirements were updated or deleted.

4.16.1 Analysis of Lessons Learned

LL WP9-1 describes the possible stigmatization of diabetes patients that can arise if components are visible and network components are visible electronically. Diabetes patients often experience their disease as a "*discreditable"* condition. Decreased visibility of their condition is therefore of particular importance for patients. The Lesson Learned is that the visibility of components should be decreased to a minimum to avoid stigmatization. This resulted in the creation of REACTION-470.

The subject of LL WP9-2 is the possibility of socialising with others having a similar condition (and suffering stigmatization because of it). This possibility is highly valued by most patients. The conclusion is therefore that REACTION should not eliminate this possibility. If limitations arise because of REACTION the Lesson Learned constitutes that new ways of communication like social networks could be used to reinforce inter-patient communication. This led to the creation of REACTION-471.

LL WP9-3 elaborates on the visibility of portable sensor patches, possibly resulting in a feeling of stigmatization. These feelings can be strong inhibitors in the uptake of products. Hence, the visibility of these devices should be as reduced as technically feasible. This Lesson Learned corresponds with REACTION-472.

The issue of reduced visibility is also illustrated by LL WP9-4. In addition to visibility and stigmatization in relation to portable devices, the visibility of public networks used in REACTION (e.g., Wi-Fi) is of importance. The visibility of these networks should also be as low as possible. As an additional part of this Lesson Learned it is emphasized that the highest feasible level of security should be implemented. REACTION-473 was created to accomplish this.

LL WP9-5 deals with requirements for Data Breach Notification, which are not explicitly foreseen in the Data Protection Directive. However, a number of countries, such as Germany and Norway, have introduced a notification requirement for data breaches. In addition, the Article 29 Working Party has argued that an extension of personal data breach notifications, beyond telecoms firms, to Information Society Services is necessary given the ever increasing role these services play in the daily lives of European citizens, and the increasing amounts of personal data processed by these services, including access to medical records. Accordingly, the Proposed Data Protection Regulation foresees the duty of notification of a data breach. It is therefore very likely that there will be a general European-wide data breach notification in future (EC COM 2012 11/4 draft)[3]. REACTION-474 was created in response to this Lesson.

LL WP9-6 describes the need for Log and log-in system. According to Data Protection Directive, the data controller "must implement appropriate technical and organizational measures to protect personal data against accidental or unlawful destruction or accidental loss, alteration, unauthorized disclosure or access, in particular where the processing involves the transmission of data over a network, and

[3] Proposal for a Regulation of the European Parliament and of the Council on the protection of individuals with regard to the processing of personal data and on the free movement of such data; (General Data Protection Regulation) http://www.statewatch.org/news/2012/jan/eu-com-general-dp-regulation-com-12-3-12.pdf

against all other unlawful forms of processing" (Directive 95/46/EC)[4]. In addition, in the 'I v. Finland' judgment of 17 July 2008, the European Court of Human Rights held that it is a positive obligation of states to ensure that information systems used in a hospital are transparent and allow assignation of responsibility in case of wrongdoings or mistakes. LL WP9-2 resulted in the adding of REACTION-475.

In LL WP9-7 Data Protection Impact Assessment is considered. Article 33 of the Proposed Regulation on Data Protection and concerns the obligation of data controllers to carry out Data protection impact assessments. Data protection impact assessments are to be carried only in certain circumstances, e.g., when data processing operations "*are likely to present specific risks to the rights and freedoms of data subjects by virtue of their nature, their scope or their purposes*". This Lesson resulted in the creation of REACTION-476.

LL WP-9-8 involves Liability and compensation for unlawful data processing. The proposed Regulations introduce a new regime for penalties and administrative fines. The considerable pecuniary penalties and fines elevate the significance of data protection on a par with corporate compliance with other topics such as competition law, anti-corruption, and money laundering (article 77). As a consequence REACTION-477 has been added.

In LL WP9-9 it is described how consent procedures should be reviewed over time with the involvement of carers. The patient should not be left alone with technology. The use of consent for legitimizing data processing is significantly restricted "*where there is a significant imbalance between the position of the data subject and the controller*", says article 7 of the Proposed Directive. In the many instances when it is not clear whether consent is genuinely given, it may be sensible to conceive of supportive or ancillary measures. One option that the REACTION platform may consider is the adoption of cooling off periods and regular interviews to verify whether, over time, the user still wants to signify his or her agreement to participate in the platform and have data related to him or her being processed. Inspiration could be taken also from mediation services in public hospitals which assist users in understanding the consequences of giving or refusing consent, or afford the possibility to renegotiate the contract of service.

LL WP9-10 states that where needed, REACTION must comply with the medical device directive. Where components of REACTION meet the definition of a medical device they must be compliant with the relevant essential requirements of the directive. This will be the case both software and physical components. Compliance with the relevant essential requirements can be demonstrated by meeting available standards.

The subject of LL WP9-11 is Reimbursement Problems. Certain member States do not recognize eHealth activities as reimbursable acts. REACTION should take this into account. Where possible a REACTION platform should be crafted in a manner that would allow it to meet existing reimbursement criteria. This issue may also create problems where REACTION is to be used on a cross border basis as the new Patient Right's Directive requires that an act of medicine to be reimbursable in the Member State of Residence for a patient to have the right to be reimbursed for it in another Member State.

LL WP9-12 states that significant cultural variations in Patient Preferences exist across Europe. A REACTION platform should be designed in a manner that takes into account such variations.

4.16.2 New requirements

The Lessons Learned in the second cycle have resulted in the addition of eight new requirements.

Key	Require-ment Type	Priority	Summary	Rationale	Fit Criterion
REACTION-477	Non-functional - Legal	Major	Liability of controller for damages due to	REACTION could consider the allocation of compensation or insurance schemes in case errors occur in the processing of medical data.	EU case law Armonias v Lithuania

[4] European Parliament and the Council of the European Union (1995) Official Journal of the European Communities L281, article 17. http://ec.europa.eu/justice/policies/privacy/docs/95-46-ce/dir1995-46_part1_en.pdf

			unlawful processing		
REACTION-476	Non-functional - Legal	Major	Data protection impact assess-ment	Data protection impact assessments are to be carried only in certain circumstances, e.g., when data processing operations 'are likely to present specific risks to the rights and freedoms of data subjects by virtue of their nature, their scope or their purposes.' Recital 71 indicates that the requirement to conduct them should apply in particular 'to newly established large scale filing systems, which aim at processing a considerable amount of personal data at regional, national or supranational level and which could affect a large number of data subjects.' The foregoing suggests that an institution operating a system such as REACTION, for instance a hospital or a national health service, should carry out an impact assessment	Proposed regulation on data protection
REACTION-475	Non-functional - Legal	Major	Log and log-in system	One requirement is a 'log-in system' used to identify and authenticate a given person when s/he accesses the medical data. Another requirement is a 'log system' that records who did what and when in an audit log. This would contribute to realise those "technical and organizational measures" capable of ensuring the traceability of those who access the data of patients. Besides robust log-in and log systems showing who has accessed information and when.	EU case law (e.g. I v Finland) D9.2
REACTION-474	Non-functional - Legal	Major	Data breach notification duty	Data Breach Notification requirements are not explicitly foreseen in the Data Protection Directive. However, a number of countries, such as Germany and Norway, have introduced a notification requirement for data breaches. In addition, the Article 29 Working Party has argued that an extension of personal data breach notifications, beyond telecoms firms, to Information Society	Requirements of the proposed regulation

				Services is necessary given the ever increasing role these services play in the daily lives of European citizens, and the increasing amounts of personal data processed by these services, including access to medical records. Accordingly, the Proposed Data Protection Regulation foresees the duty of notification of a data breach. It is therefore very likely that there will be a general European-wide data breach notification in future.	
REACTION-473	Non-functional - Legal	Major	All public networks (e.g. WIFI) created as a result of REACTION should have as low visibility and as high security as is technically possible.	The lower the visibility of such equipment (in terms of network visibility) the less the chance that an individual's condition might become apparent to those who do others in situations where the parent does not wish this to happen. This is important in connection to issues of stigmatization - see task 9.2	Networks should not be unnecessarily visible and should be secure.
REACTION-472	Non-functional - Legal	Major	A portable sensor patch should have as reduced visibility as is technically feasible.	The lower the visibility of such equipment the less the chance that an individual's condition might become apparent to others in situations where the patient does not wish this to happen. This is important in connection with issues of stigmatization - see task 9.2	Equipment should have as low a level of visibility as is technically possible.
REACTION-471	Non-functional - Legal	Major	Individuals that suffer stigmatisation (including through conditions such as diabetes) often value the ability to socialise with others having a similar condition or sympathetic healthcare professiona	Individual patients often use the opportunity to meet such groups as a coping mechanism for the stigmatising effects there condition can entail. A REACTION platform should not reduce such possibilities too much. Where such possibilities are drastically reduced alternatives should be offered, for example online social networking possibilities.	Individuals should not feel that a REACTION platform has eliminated their access to other patients and sympathetic health care professionals which represent an important coping mechanism for individuals that feel stigmatised.

			ls. REACTION should not eliminate this possibility.		
REACTION-470	Non-functional - Legal	Major	The potential stigmatising effect of REACTION due to visibility should be decreased to a minimum.	Most diabetic patients experience diabetes as 'discreditable' individuals in terms of stigmatisation (See ethical analysis - task 9.1). For such individuals control over personal information is extremely important. Individual patients will not want to unnecessarily increase visibility of their condition as this will mean that they have reduced level of control over their personal information and will therefore have less ability to control who they reveal their condition to. Visibility of components and also electronic visibility of networking components should therefore be kept to a minimum.	Patients should feel that a REACTION like platform will not result in a overall increase in the visibility of their condition and consequently a reduction in their ability to conceal it, it they should wish to do so.

4.16.3 Updated requirements

No requirements have been updated.

4.16.4 Deleted requirements

No requirements have been deleted.

4.17 Lessons Learned in WP10

The work undertaken in WP10 involves platform integration and implementation. FORTH-ICS is the WP leader and eight Lessons Learned have been collected and validated from this WP.

Org. No.	Experience and knowledge gained	Lesson Learned	Requirement affected
FORTH-ICS 1	Significant differences have emerged between the specifications for the in-hospital and the primary care environments. Such differences lead to profoundly different data models and components.	Reusability can unlikely be applied between different applied environments, thus requiring much more efforts in terms of resources.	
FORTH-ICS 2	As put into evidence also in the first iteration cycle, strict planning has to be applied for leading to the successful release of the prototypes in the expected times. However, partners should commit more themselves to comply with the agreed schedule.	Delays in the release of the components either in binary or in source mode in the repositories lead unavoidably to delays in the release of the prototypes. Higher commitment is required to all involved partners.	
FORTH-ICS	The release of components verified in terms of unit tests does not	Once again the availability in time of the components in the	

3	guarantee an easy and immediate integration in FORTH-ICS test environment. In fact, the involved factors can be countless and many efforts and tries have to be done before making the released component work in the test environment (with many retrofits in the installation and configuration procedures). The installation guidelines of the various released components were very difficult and several times we had to go back and revisit the solution trying to find a solution. This was also due to the fact that unit testing had been done but not for the operating environment of the server.	repository is fundamental and components should be also accompanied by a first document describing the installation and configuration procedures. All developers must perform the unit test of the components under Windows Server 2008 (this is what is agreed) as the operating environment of the server. Furthermore, all developers must take into account what has already been used to avoid possible conflicts. Well defined instructions with configuration guidelines (wherever is essential) for each web server, service/web service, component, middleware, etc. are required. More specifically, when components are released for integration the following data must be provided (at this moment they have been provided only on demand): 1) Manuals on how to deploy them on the server; 2) Manuals/instruction on how to use the components; 3) Unit test results with the results.	
FORTH-ICS 4	Binary and source repositories are becoming a key part of the REACTION project life cycle.	Taking routine backups of the repositories is necessary for being able to recover data, components, configuration and the system itself in case of any problem, thus guaranteeing a smooth and solid life cycle of the project.	
FORTH-ICS 5	We have been very flexible on the development tools and the technologies we used (as a consortium) for the development of the various components and prototypes (in order to allow the highest degrees of freedom to all partners). However, currently we are using 3 Database Management Systems (Postgressql, MySQL, Microsoft SQL) and 3 different web servers (Apache, Apache tomcat, IIS). Problems are (from the integration point of view): - Difficulty in creating a running environment to deploy the working components (installation is hard); - Difficulty in configuring this environment; - Difficulty in integrating heterogeneous components and perform test.	The absence of any rule or constraint in the target and development environment increases the flexibility but at the same time also the complexity, making the integration very cumbersome. At this moment it seems rather difficult to create a unified platform and probably some main rules should be defined from now on.	

FORTH-ICS 6	The additional constraint of setting up a software life cycle compliant with the amended MDD, led to the choice of a specific development framework (i.e. AndroMDA) for the in-hospital application. Although it uses an automated mechanism to create source code it is not transparent for manual configuration. Because AndroMDA creates automatically the basic stack of the source code all the configuration is made indirectly to the AndroMDA framework and not directly to the development source code components, making things quite complex. The development of the in-hospital application requires the knowledge of an additional framework (i.e. AndroMDA) and unfortunately the community and the documentation of the AndroMDA framework are quite poor.	Indirect control of the source code framework is quite complex and requires either very skilled developers or a strong training and learning phase. The integrated environment increases the control but reduces flexibility.	
FORTH-ICS 7	In the in-hospital prototype development, several upgrades in terms of target platforms (different version of tablet PC and Android) or in terms of the development platform (different versions of MagicDraw and UML) have led to frequent adaptations to the new environments with significant efforts in terms of involved resources.	A more long-term view should be adopted in the selection of the target platforms and consequently the development platform and tools thus avoiding too frequent adaptation of the already developed software to the new platforms.	
FORTH-ICS 8	Profoundly different requirements have been finally expressed for the in-hospital and the primary care environments leading to different solutions in terms of software and prototypes.	Impossibility to realize applications that can be configured for different environment and necessity to increase the development efforts in order to build different applications for different environments.	R-69

4.18 Change request and re-engineering originating from WP10

This Section provides an analysis of the eight Lessons Learned from the work performed in WP10 in the second cycle. The work resulted in four requirements being added and one updated. No requirements were deleted.

4.18.1 Analysis of Lessons Learned

The main outcomes of Lessons Learned are:

LL WP10-1 assesses the large differences between the in-hospital and the primary care environments after the release of the detailed specification for both environments. These large differences minimize the opportunities for software reusability and consequently increase the required resources.

LL WP10-2 refers to the various time plans set up by the consortium for the release of the building blocks to be assembled for generating the prototypes. A building block must be considered as released when stored either as source code or in binary form in the software repositories created by FORTH-ICS for the integration support. Each building block must have overcome the unit tests before its release in the repository. Since the complexity of the platform is very high, a successful integration cannot be obtained without partners fully comply with the established time and modalities.

LL WP10-3 describes the complexity of the integration phase and identifies a reference environment for the execution of unit tests and set up the necessary accompanying documentation to be provided together with the released building block.

LL WP10-4 identifies the need for regular back-ups to be performed in the software repositories being these ones a fundamental element of the software life cycle.

LL WP10-5 puts into evidence the high flexibility allowed to partners in the selection of the deployment environment. This high flexibility facilitates the work of the partners optimizing and exploiting very well their expertise, but makes very complicated the integration phase and the vision of the final platform.

LL WP10-6 deals with the tools selected for the in-hospital environment for the optimal support of the software life cycle. These tools increase the control on the software development but significantly flexibility and require more skilled developers.

LL WP10-7 describes how during the development several times the target devices or versions of the target operating systems have been changed introducing changes and adaptations to the already developed software. Even if this can be interesting from the point of view of pure research on software development, it increases the development efforts and a more long-term view should be applied in the selection of the target platform from a hardware and software point of view.

LL WP10-8 describes the huge differences in the specifications for the primary care and for the in-hospital environments. This produced an impact in the software development making impossible the implementation of applications which can be simply configured for the different environments. Now different applications have to be built increasing significantly the required effort.

4.18.2 New requirements

Four new requirements have been added as a consequence of the consolidation described in LL WP2-2.

Key	Requirement Type	Priority	Summary	Rationale	Fit Criterion
REACTION-462	Functional - In-hospital pilot application	Major	Interface for user inputs from portable computer in order to store data in In-hospital data storage	For the In-hospital prototype user input should be possible. The user data should be stored in the data storage.	User input can be stored in the In-hospital prototype storage for further processing.
REACTION-452	Non-functional - Security	Major	Communication between the Reaction Device Hosting Server and the EPR/EHR System MUST be authentic (entity authentication), with integrity, and confidential.	It must be assumed that data transmission from the Reaction Device Hosting Server to the EPR/EHR System and vice versa takes place over an insecure channel, i.e., data might be overheard or tampered with. Since personal data is to be transmitted it MUST be ensured that the communication channel is authentic, with integrity, and confidential.	Availability of mechanisms to provide communication channels with authenticity, integrity, and confidentiality.
REACTION-443	Functional - REACTION platform	Major	Data exchange with third party systems	Ideally integrates information from outside the REACTION platform	Should be able to import and export data in an interoperable

				(e.g. Laboratory Information Systems in hospital or primary care with blood glucose and glycated haemoglobin).	way (e.g. HL7) to third-party systems.
REACTION-379	Functional - In-hospital pilot application	Major	Interface for transmission of glucose values from POCT system to In-hospital prototype	As decision support is a time critical process, data from the POCT device should be transferred directly (without detour to LIS) to the In-hospital prototype in order to speed up the transmission process. Therefore an interface has to be provided.	Interface to POCT device is available for the In-hospital prototype.

4.18.3 Updated requirements

Requirement REACTION-69 has been reviewed because it refers to easy configurability of applications in different environments. Since this cannot be obtained as explained in LL WP10-8 an increased implementation effort has been required in order to solve this Major issue (which was initially considered Critical). However, given the detailed specifications for the in-hospital and primary care environments this requirement cannot be implemented and has been resolved with this resolution.

Details of the modified requirement are shown below.

Key	Requirement Type	Priority	Summary	Rationale	Fit Criterion
REACTION-69	Functional - REACTION platform	Major	System Configuration	The components and applications should be made in a way that makes easy the configuration	Theoretically without any recompilation, the application should be easily configurable for the different environments

4.18.4 Deleted requirements

No requirements have been deleted.

5 Validation results – In-hospital prototype

5.1 Summary of verification results

These verification results refer to the 2nd year prototype development process. Specifically this verification phase has been performed using the release-1.2.1 (in the tags folder of the SVN repository) of the REACTION components (i.e., Back end and Front end). The development process has been conducted meeting the procedures listed in *ID2-6-2 Prototype Application Specification 2*. The In-hospital prototype has been structured into two main layers. The back end layer contains the implementation of several services (GlucoManSys) and the front end layer contains the user interface (UI) implementation (GlucoManSysFrontEnd). Table 1 presents a mapping of the implemented services (in the left column) of the back end mapped in the system main functionalities (in the right column) required for the second year prototype.

Back End (server) functionalities	REACTION system functionalities for In-hospital prototype
UserService	User Management
EnrolmentService **FacilityService** **PatientDataManagementService**	Ward Management
TherapyAdjustmentService **MeasurementService** **PatientService** **DrugService** **ProposedMedication Service** **MedicationService** **RecentActivitiesService**	Data Management (i.e., Glucose and Drug Management)
TaskManagementService	Open Task Management
BasalBolusTherapyRegimentHandlerService	Decision Support System

Table 1: Mapping between implemented services and main functionalities

The complete list of the implemented services (left column) in the back end can also be mapped into the use cases shown in Figure 1, presenting in more details the required functionalities to be accomplished by the in-hospital prototype. Each specific implemented service consists of sub-functions (methods) which are able to address and satisfy the sub-cases shown in Figure 1 (from the *ID2-6-2* use case diagram).

5.1.1 Description of the implementation services

The list below reports a short description of the developed services in the back end that have been implemented for the **release-1.0** of the in-hospital prototype.

Enrolment Service: This service performs the enrolment procedure for the patients inside the Glucose Management System (GMS). The service provides 3 functionalities of starting, cancelling and updating enrolment for a specific patient (based on the patientID).

Facility Service: This service provides information about the rooms in a specific ward; it contains 1 function for loading room information (description, ward name).

Measurement Service: This service is for managing measurement records. A measurement record can also be filled in manually by a user.

Drug Service: This service is used in order to load drug records that have been inserted in the system. It returns a list of registered drugs.

Medication Service: This service performs operations for storing, edit and finds medication records for a specific patient (i.e., medication type, units, unit type nutrition, etc.).

Proposed Medication Service: This service manipulates proposed medication records. The proposed medication record holds information on how many units and what time range a medication should be administered.

Patient Service: This service performs patient finder and load patient enrolment function in order to get information about the enrolled patients in the GMS.

Therapy Service: This service implements the therapy adjustment. It contains a set of therapy adjustment functions which add, update or change an existing therapy.

User Service: This service performs user administration. It contains a functionality of user finding which returns a list of the users of the system.

Patient Data Management Service: This service will be used as an endpoint for the HL7 adapter for the HIS/LIS integration.

Task Management Service: This service is for providing alerts and notifications to the medical staff in for the patients that are enrolled for glucose management.

Basal Bolus Therapy Regimen Handler Service: This service is a first implementation of the decision support mechanism that will assist the physicians for the proper treatment of the patients with diabetes.

The above services can be mapped in the use cases defined in *ID2-6-2* (see Figure 1).

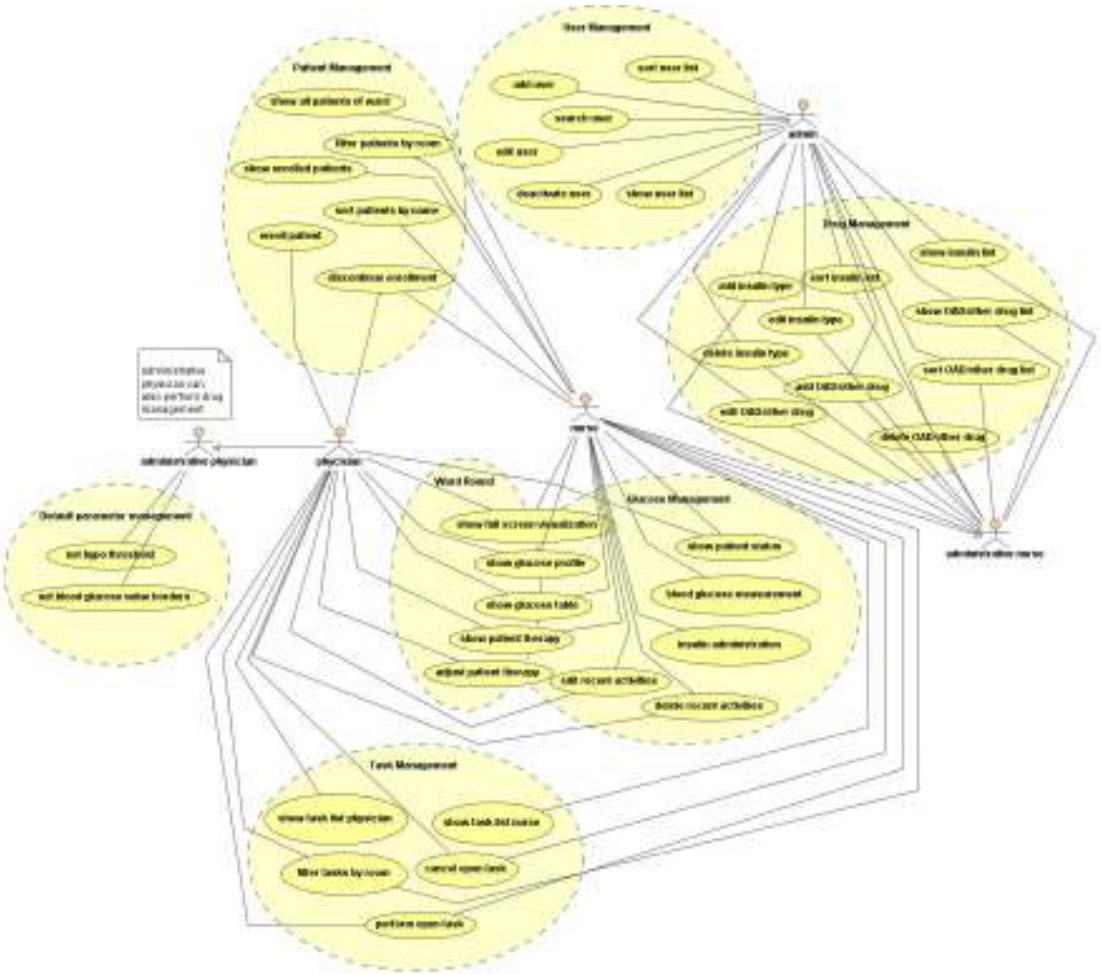


Figure 1: Use cases from ID2-6-2

5.1.2 Verification phase

The verification phase consists of three different sessions the unit tests, the integration tests and the system tests. The unit tests are based on test cases and have been implemented and performed by each developing partner (for the components of its competence) in the development machine.

Components have been delivered to the partner responsible for the integration (FORTH-ICS) together with their unit tests report describing the successful results of the unit tests performed on the components. The unit tests have been included in this document.

Then, the integration tests have been performed in the internal test site assembling the components together and testing their interoperability. Finally the system tests, more oriented to the verification of the requirements that the first-year prototype had to match, have been performed again in the internal test site and reported in this document.

5.1.3 Unit testing

Unit testing is a process of evaluating units of source code in order to determine if they are fit to use. A unit is a small part of an application. In different programming techniques a unit may represent a function, procedure (procedural programming) or method (in object oriented). Unit testing is created and executed by programmers during the development process in the development machine. The unit test framework selected for the back end is the TestNG framework[5]. The diagram in Figure 2 shows the order in which the initialisation/tear down methods are invoked with regard to TestNG tests:

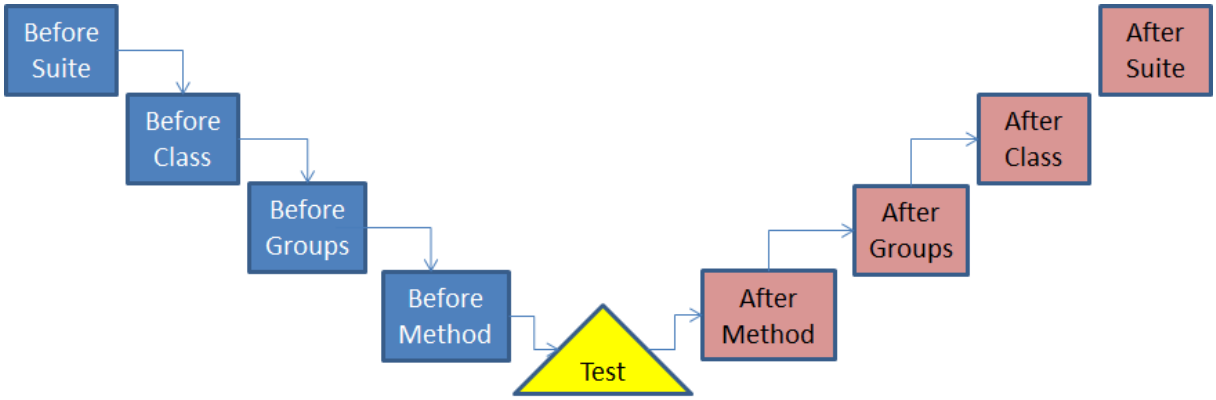


Figure 2: Order of the initialisation/tear down methods in testing framework

TestNG is a testing framework inspired from JUnit and NUnit but introducing some new functionalities which make it more powerful and easier to use, such as:

- Annotations
- Run your tests in arbitrarily big thread pools with various policies available (all methods in their own thread, one thread per test class, etc.)
- Test that your code is multithread safe
- Flexible test configuration
- Support for data-driven testing (with @DataProvider)
- Support for parameters
- Powerful execution model (no more TestSuite)
- Supported by a variety of tools and plug-ins (Eclipse, IDEA, Maven, etc.)
- Embeds BeanShell for further flexibility
- Default JDK functions for runtime and logging (no dependencies)
- Dependent methods for application server testing.

[5] http://testng.org/doc/index.html

The TestNG provides a way for reporting the test results using the log4J framework. These reporting capabilities allow us to examine what test that has been run what are the results for specific test cases.

For the front end application the smallest identifiable units are on the one hand Android Dialogs and Android Activities at the surface and on the other hand Android Services, Android Broadcasts and some Java-based components, as for example the handler for managing web service calls, in the background. Figure 3 illustrates the splitting from the system to the smallest identifiable units:

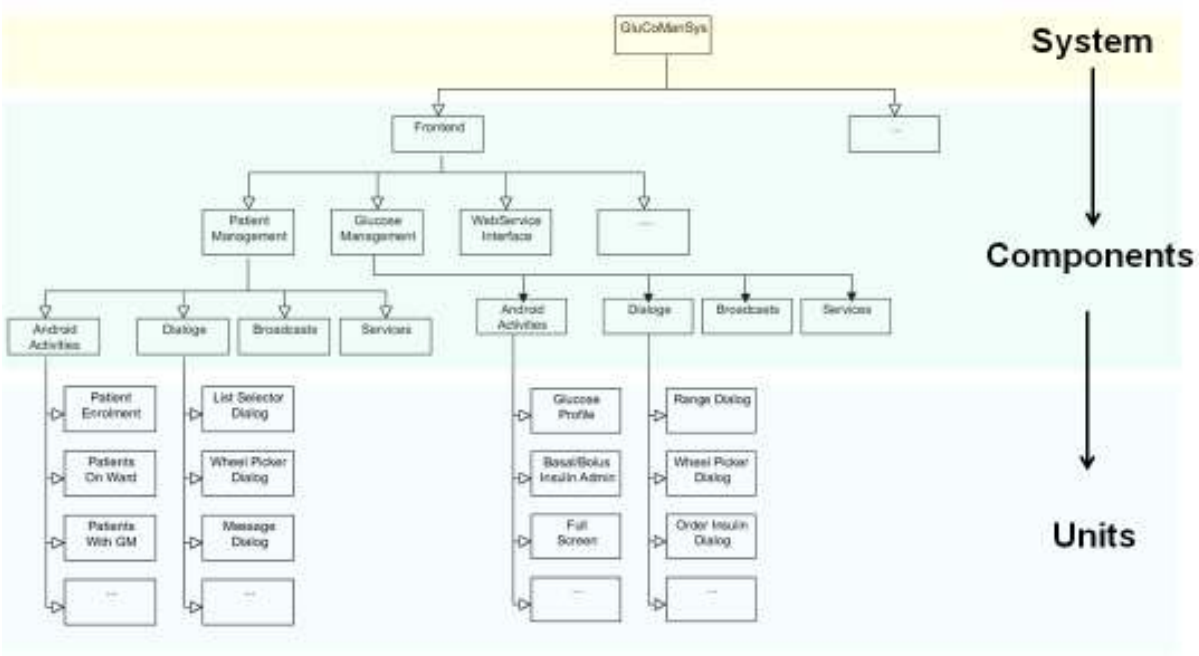


Figure 3: Splitting the system into units

The testing has to be performed in the opposite direction, as indicated in Figure 4.

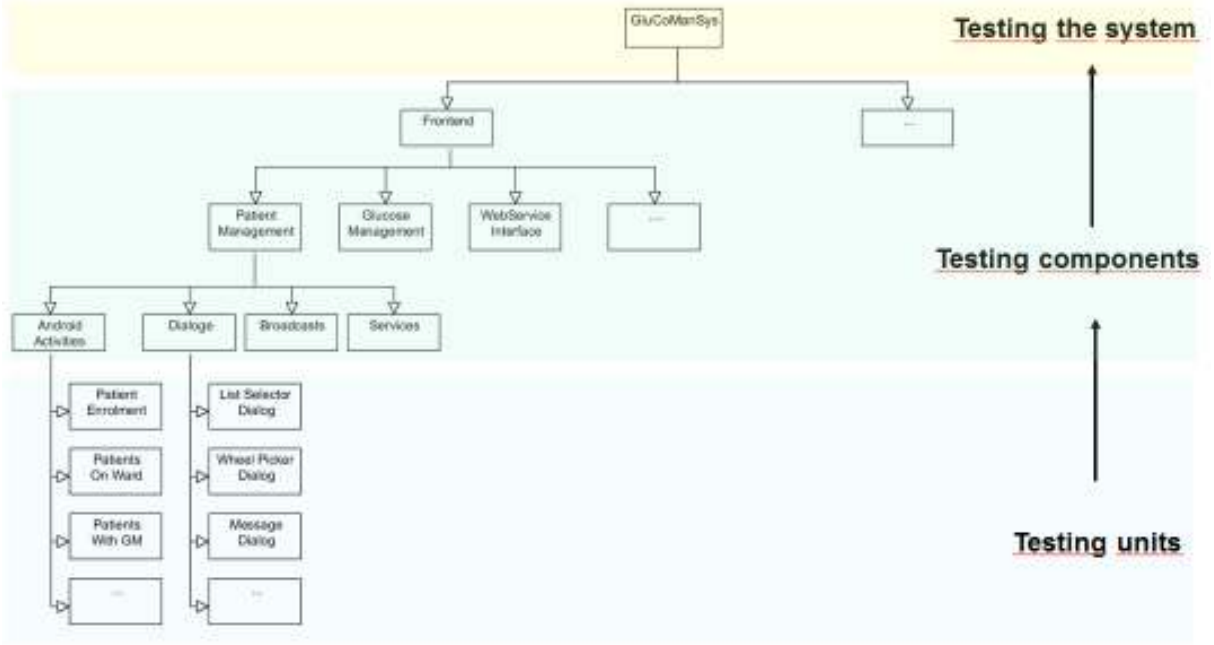


Figure 4: Testing the system

Testing of identified units should test the encapsulated functionality of one unit independently of any other units. In contrast Testing components mean to test the functionality of in several components encapsulated units. Finally system test should test all of the system's components.

At the moment only unit testing is automatically performed at the front end, using Instrumentation tests. Instrumentation tests simulate user interactions and ensure that the application prints out the expected output for every input. For testing encapsulate units, each test case initializes the target application with data before the target surface (Android Activity or Android Dialog) is started by the instrumentation test runner. Afterwards a robot simulates user interactions to verify that the surface behaves as expected.

For testing the front end application an own Android test application was created, which consists of test classes, where each test class represents a group of tests. Each test class can be activated or deactivated in a TestConfig.properties file in the asset directory of the application. A GluCoManSysTestSuite loads the TestConfig.properties, parses its content and fill the test suite with the test classes that should be executed by the InstrumentationTestRunner, if the proper test class is set to true in the properties file. Each test class extends the ActivityInstrumentationTestCase2 class, which provides testing of Android Activities and offers a method setUp(), which initialize the environment before each test runs as well as a method tearDown(), which is called after each test has finished and makes sure that the environment is cleaned up before moving to the next test. The simulation of an application's surface is done by an external library called Robotium[6]. With the support of Robotium, a free testing tool, which simulates touching, scrolling, clicking and other actions, test case developers can write function, system and acceptance test scenarios spanning multiple Android Activities.

5.1.3.1 Back end unit tests

The unit tests that evaluate the back end development can be divided in two major categories, Domain tests and Service tests.

Domain tests are the unit tests that evaluate data persistence functionality (DAO tests). These, more into details, have to do with Entity functions that implement data persistence transactions (i.e., Create, Read, Update and Delete) and the required transformation procedures regarding the value objects (i.e., the objects that are being exposed as input/output of the business logic). These entity functions have to be tested in specific test cases in order to verify their proper operation and transformation, providing the expected results. This stage is essential in order to proceed with the next development level which is the service implementation.

The domain test report has been summarised in Appendix A.

Service tests are the next level of the development phase and represent the functions that would be exposed and used from the front end application due to various use cases. Moreover the service tests provide an assessment mechanism for the proper execution of the business logic of the back end.

The service test has been divided in the following categories:

- Patient tests
- Patient Data Management tests
- User tests
- Enrolment tests
- Therapy tests
- Measurement tests
- Medication tests
- Proposed Medication tests
- Facility tests
- Task tests
- DSS tests
- Recent Activities tests

[6] http://code.google.com/p/robotium/

Each one of these categories contains test cases for the methods which have to succeed for a successful verification of their operation.

The results of the test cases for each of the above categories are shown in Appendix B.

5.1.3.1.1 Summary of the unit tests

This is a summary of all unit tests performed in the development machine:

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running TestSuite
Running TestSuite
DEBUG
eu.reaction.prototype.glucosemanagement.service.task.TaskManagementCoreServ
iceImpl - * description:
DEBUG
eu.reaction.prototype.glucosemanagement.service.task.TaskManagementCoreServ
iceImpl - * scheduledStartDateTime: Thu Apr 05 09:00:00 EEST 2012
DEBUG
eu.reaction.prototype.glucosemanagement.service.task.TaskManagementCoreServ
iceImpl - * scheduledEndDateTime: Thu Apr 05 11:00:00 EEST 2012
DEBUG
eu.reaction.prototype.glucosemanagement.service.task.TaskManagementCoreServ
iceImpl - * gracePeriodBefore: 1800000
DEBUG
eu.reaction.prototype.glucosemanagement.service.task.TaskManagementCoreServ
iceImpl - * gracePeriodAfter: 1800000
DEBUG
eu.reaction.prototype.glucosemanagement.service.task.TaskManagementCoreServ
iceImpl - * createdBySystem: true
INFO
eu.reaction.prototype.glucosemanagement.service.task.TaskManagementCoreServ
iceImpl - Successfully created new task with task id <1333605600000-
1800000-1333612800000-1800000-THERAPY_ADJUSTMENT-e_id>.
INFO
eu.reaction.prototype.glucosemanagement.service.task.TaskSchedulerServiceIm
pl - Scheduled [4] new tasks for enrolment id <e_id>.

Tests run: 243, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 53.75 sec
```

From the unit test results we can summarise that 243 unit tests have been performed, without any failure or skip.

5.1.3.2 Front end unit tests

According to the medical device directives for software it is required to provide a detailed documentation of what has been tested and what are the results of these tests.

For the front end application a special system was considered to generate the documentation of the test cases. On the one hand the specification of each test group (test class) and for each test case (test method) is written directly in the source code using Java Doc comments. Each test case specification consists of a general description of the test case itself and all the steps which are performed by the test case. The following example shows a test case called testTaskListOnClickListener, which should check the correct behaviour of the task list in case a click or long click event is performed.

```
/**
* The test case checks the correct behavior of the task list in case a click or
long click event is received. Therefore the test case checks if the expected
window opens or the correct patient (patient id) and task (task id) is in use
* 1) Click on group list item with text '<string/id_room>: R-100'
```

```
* 2) Check if text 'Foreman, Eric' is visible
* 3) Click on group list item with text '<string/id_room>: R-100'
* 4) Check if text 'Foreman, Eric' is not visible
* 5) Click on child list item with text 'Cameron, Allison'
* 6) Wait for dialog with title 'string/tl_error' and check if patientID is
'pAllisonCameron'
* 7) Dismiss dialog
* 8) Click long on list item with text 'Adams, Jessica'
* 9) Check if dialog with title 'string/tl_select_operation' is presented
* 10) Check if dialog contains 2 list items:
*      - 'string/le_start_gm'
*      - 'string/le_add_task'
* 11) Click on list item with text 'le_start_gm'
* 12) Wait for dialog with title 'string/tl_error' and check if patientID is
'pJessicaAdams'
* 13) Dismiss dialog
* 14) Click on bg measurement symbol of patient Jessica Adams
*         15)         Check         if         dialog         with         text
'<string/bt_bg_measurement>"\n"<string/id_middlestring_task_activity>      Jessica
Adams' is presented
* 16) Click on button to perform task
* 17) Wait for dialog with title 'string/tl_error' and check if patientID is
'pJessicaAdams'
* 18) Dismiss dialog
* 19) Click on medication symbol of patient 'Allison Cameron'
*         20)         Check         if         dialog         with         text
'<string/id_activity_bolus_admin>"\n"<string/id_middlestring_task_activity>
Allison Cameron' is presented
* 21) Click on button to perform task
* 22) Wait for dialog with title 'string/tl_error' and check if patientID is
'pAllisonCameron'
*/
public void testTaskListOnClickListener() {
   //1)
   mySolo.clickOnText(this.getActivity().getString(R.string.id_room) + ": R-100");

   //2)
   Assert.assertTrue(mySolo.searchText("Foreman, Eric", true));

   //3)
   …
}
```

For the test results a test listener was implemented, which extends the InstrumentationTestRunner and generates a test report, including the name of the test case, the test group (test class), the test result, the needed time, failures (if appeared) and a screenshot of the current surface (if there was a failure, the screenshot contains the surface at the time the failure appeared). An example is shown in Figure 5.

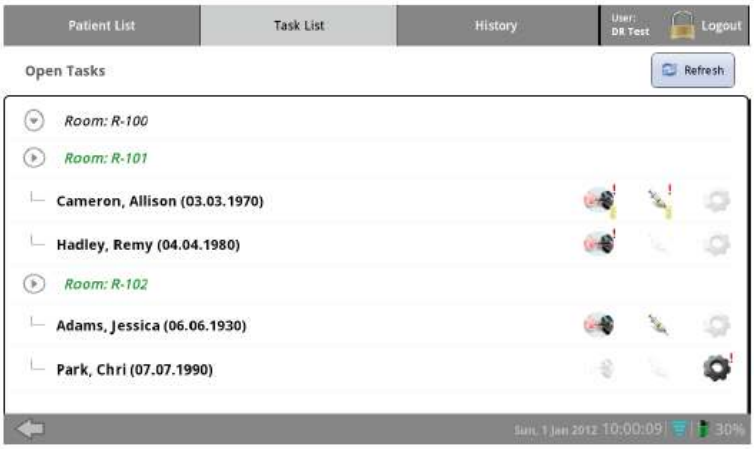
Test ID	**testCheckCorrectPresentationOfTasks**
Test Group	**TaskManagementTests**
Test Result	PASSED
Time needed	976.009 seconds
Screenshot of result	

Figure 5: Test result of test case testCheckCorrectPresentationOfTasks

The complete test report of a test run is presented in Appendix C.

5.1.4 Integration testing

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It is the next after unit testing and before system testing. Integration testing takes, as its input, modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

For the in-hospital scenario the integration test has to be performed with the front end and back end components. The integration process has to test all the available back end services that are being activated by the Front end application. The procedure regarding the integration testing and the findings is provided bellow:

Step 1: Deployment of the binary back end component from the FTP (i.e., glucomansys-webservice-1.2.1.war) to Tomcat server.

Status: FAILED

Comments: Although the deployment was ok, when I tried to perform a test request through SoapUI I got an "**HTTP 404**" error response.

Solution: The component had to be rebuilt it from the SVN (tagged version 1.2.1) and re-deployed to Tomcat server. The deployment was successful. The component has been tested with SoapUI and responses are now received without errors.

Step 2: Deployment of the binary back end integration test component from the FTP (i.e., glucomansys-integrationtest-webservice-1.2.1.war) to Tomcat server.

Status: FAILED

Comments: Although the deployment was ok, when I tried to perform a test request through SoapUI I got an "**HTTP 404**" error response.

Solution: I have re-built the component from the SVN (tagged version 1.2.1) and re-deployed to Tomcat server. The deployment was successful. The component has been tested with SoapUI and responses are now received without any errors.

Step 3: Installation of the binary front end component to device from the FTP (according to the guidelines).

Status: SUCCESS

Comments: The application has been installed and configured according to the guidelines.

Solution: -

Step 4: Initialization of the database and back end integration testing.

Status: SUCCESS

Comments: The initialization of the database was performed and the WS has been tested with SoapUI without any problem. The requests/responses were properly performed with secure and insecure endpoint connections.

Solution: -

Step 5: Front end integration testing with the back end

Status: FAILED

Comments: The front end was unable to retrieve any data from the back end. The list of tasks is empty and messages that were no tasks have shown (not the "error on loading tasks"). Additionally no patients were shown or activities. Using the SoapUI instead of the front end I was able to retrieve the task list which was not empty but filled with the test data from the database.

Solution: It seems evident that the interaction between the front end and back end is not working properly. Since the back end works fine with SoapUI tests, possibly a new version of the front end has to be built and to be uploaded to the FTP.

All the developer partners have been informed about the results of the integration tests. The involved partners were worked together towards the implementation of the required bug fix and new releases have been uploaded to SVN and FTP repositories.

The current stable release of the In-hospital prototype is the release-1.2.1 tagged in the SVN and FTP repositories. The integration process has to test all the available back end services that are being activated by the Front end application. Thus there is a need to define which web services in the back end have been:

- Fully integrated
- Partially integrated
- Not integrated

Table 2 represents all the services in the back end and the level of integration of the Front end for the **release-1.2.1**:

Back end Services	Integration status with the Front end
PatientService	Fully integrated
EnrolmentService	Fully integrated
TherapyAdjustmentService	Fully integrated
MedicationService	Partially integrated
MeasurementService	Partially integrated
DrugService	Fully integrated
FacilityService	Fully integrated
ProposedMedicationService	Fully Integrated
UserService	Not integrated
PatientDataManagementService	Not Integrated
TaskManagementService	Fully Integrated
BasalBolusTherapyRegimenHandlerService	Fully Integrated
RecentActivitiesService	Fully Integrated

Table 2: Level of integration between the back end services and the front end

For the fully integrated services, the integration tests have been successfully performed. The partially integrated services are going to be extended with more functionalities in the 2nd year prototype (i.e., edit/delete measurement, medication), thus they are reported as “Partially Integrated”.

5.1.5 System testing

System testing is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. All the functional requirements that were determined in *ID2-6-2 Prototype Application Specification 2* were filtered including only the ones impacting on the 2nd year in-hospital prototype.

Specific tests have been performed in order to specify which of the functional requirements have been satisfied and the level of the satisfaction. Four levels of satisfaction have been defined and assigned in each of the functional requirements involved:

- Fully satisfied
- Partially satisfied
- Improperly satisfied
- Not satisfied

The distribution is shown in Figure 6.

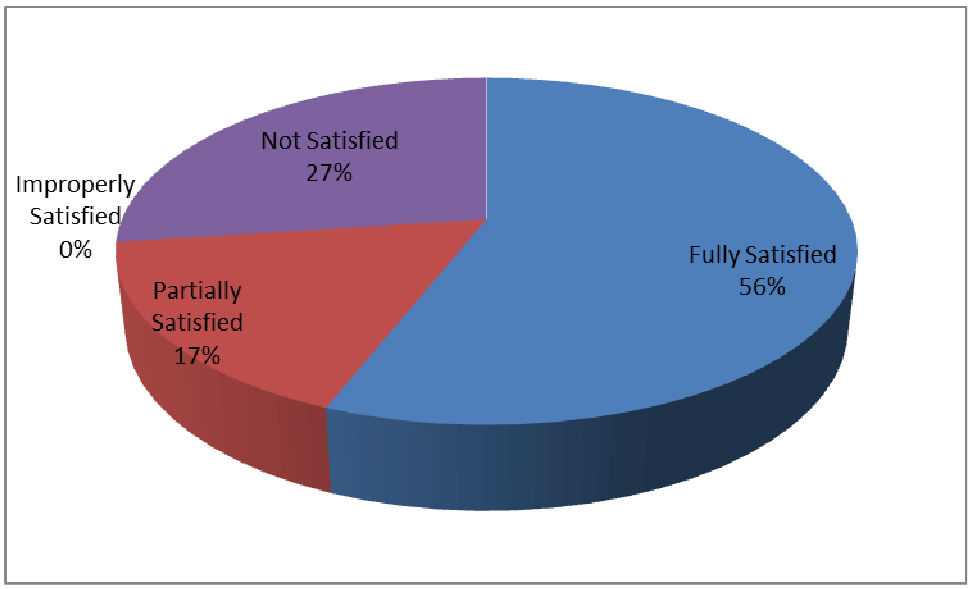


Figure 6: Satisfaction of requirements for second-year in-hospital prototype

In some cases additional tools have been used in order to populate the database and to be able to run some specific tests referring to specific requirements.

It has to be noted that for several requirements the complete satisfaction of the requirement itself has been gradually scheduled during the entire course of the project, thus the sentence “partially satisfied” has to be seen as “satisfied for the implementation foreseen in the second year (even if implementation has to continue in the next years)”.

Details of the satisfaction of requirements are listed in Appendix D

5.2 Summary of validation results

Not available yet (clinical study for insulin dosing protocol will start in mid-2012.

5.3 Summary of results from usability testing

5.3.1 Requirements

The development of mobile applications in a medical context provides engineers with a complex task. In addition to the aim of supporting the daily medical routine, usability is an additional important issue, according to clinical safety, to consider. Non-intuitive usability often leaves the user frustrated and

unable to complete simple tasks. The lack of usability in medical devices is dangerous, and can lead to unforeseeable risks to patients.

Therefore EN 62366 defines requirements for a process to analyze, specify, develop, verify and validate usability aspects of medical devices. According to EN 62366 the manufacturer has to implement, document and follow a usability orientated development process to ensure safety for patients, users and other persons in terms of usability. EN 62366 should guarantee that the final user interface is intuitive and easy to learn or rather to use.

5.3.2 Methods

From the beginning the development of the in-hospital glucose management system followed a user-centred design to avoid later preventable use errors and to meet the requirements of EN 62366. Already after the completion of the first prototype usability tests were performed to collect deeper information, according to workflow support, functionalities and how to display patient and therapy information. Afterwards the information was used to draw up the collected requirements in mock-ups, which were evaluated and verified by clinicians once again. Finally, the verified design was implemented in mobile Android user interface, which communicates via WLAN to a server application that is responsible for business logic and data storage. During the all development process continuous feedback of selected clinicians was gathered.

In December and January extensive second usability tests were performed with the aim to verify the already implemented functionality, design and workflow of the in-hospital glucose management prototype.

The usability tests were divided into two parts. At the first part a Heuristic Evaluation took part with the aim to identify usability problems of the user interface based on 10 predefined Heuristics:

GMS01 Feedback of current system status
The system should give appropriate feedback within an appropriate time.
Examples: User measures blood glucose; user is waiting for data from lab system

GMS02 Speak the Users 'Language
The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.
Examples: the medical (clinical) language should be supported and conventions from the medical environment should be considered. Information should be presented and requested in logical order.

GMS03 Reversible Actions
The user should be free to explore the system without penalty.
Examples: The user aimed to do a blood glucose measurement and by mistake she loaded the insulin administration. It should be easily possible to go back to the blood glucose measurement.

GMS04 Consistency
The same word, phrase, action, or situation should always mean the same thing.
Example: The action "blood glucose measurement" should not be substituted for example by "BG-measurement" or "blood sugar measurement" in a submenu.

GMS05 Error Prevention
Prevention is better than cure. Careful design can prevent a problem from occurring in the first place.
Examples: different (meaningful) names for different buttons; safety questions for critical action, e.g., changing of suggestions by decision support); visible alarm borders

GMS06 Recognition rather than Recall
Make objects, actions, and options visible
Examples: relevant information should be visible over different dialog screens; useful default values should be available; mandatory fields should be visible clearly

GMS07 Aesthetics and Minimalist Design
"Less is more".
Examples: Dialogues should not contain information which is irrelevant or rarely needed. Every piece of unnecessary decoration in a dialogue competes with the relevant units of information for the attention of the user.

GMS08 Good Error Messages
A good error message helps users recognize, diagnose, and recover from errors.
Good error messages should be: in plain language (no codes), precise (precisely indicate the problem), defensive (never blame the user), constructive (suggest a solution), and multi-level (include a link to further information or the help system).

GMS09 Hardware/Software
Information on the display should be easily readable. Smooth working has to be possible.
Examples: font size of letters; illumination and contrast are sufficient even in light rooms; needed buttons are well positioned and can be reached easily; immediate feedback of system

GMS10 Help and Documentation
Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Help functions/messages should guide user through problems. Information on the display should be easily readable. Smooth working has to be possible.
Example: User should know/find help how the algorithm calculates the insulin dosage; font size of letters; illumination and contrast are sufficient even in light rooms; needed buttons are well positioned and can be reached easily; immediate feedback of system.

At the Heuristic Evaluation 10 evaluators walked through the user interface and evaluated the design using these 10 heuristics.

At the second part of the usability trials Thinking Aloud Tests were conducted in the Medical University Hospital Graz, using 9 nurses and 6 physicians. Aims of the thinking aloud tests were to:

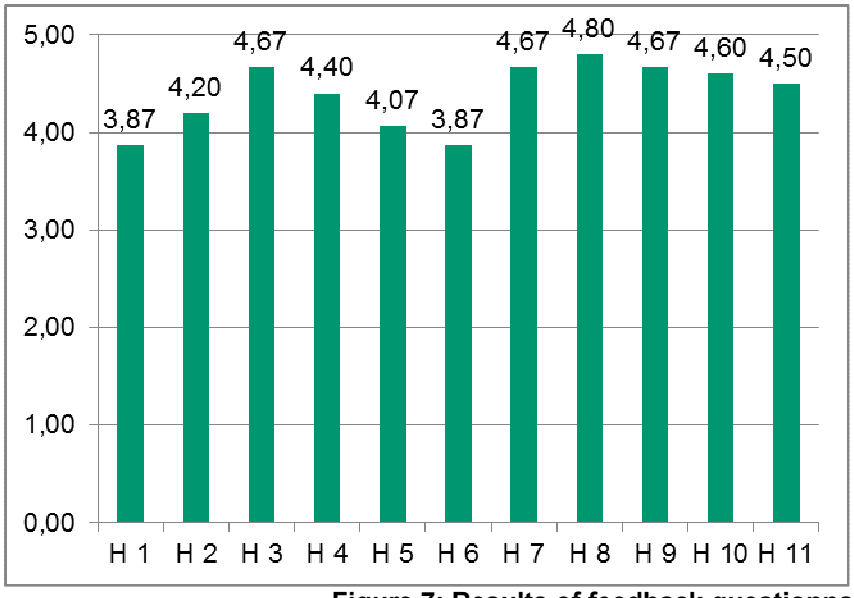
- Test the usability of the user interface
- Test the learnability of the user interface
- Test the workflow support of the user interface related to the general ward

At the Thinking Aloud tests the participants performed test tasks to simulate usual use cases. All participants were asked to verbalize their thoughts while using the system. After the test tasks, there was a conversation between the facilitator and the participant where they talked about points of interests, which had not been achieved during the tasks. At the end of each test the participant rated the system using the heuristics of the heuristic Evaluation using a 5 point scale.

All user operations were video documented and afterwards analysed.

5.3.3 Results

Generally the mobile in-hospital glucose management system earned positive feedback. 14 of 15 participants mentioned that they would prefer the mobile system against the current paper based solution. The result of the feedback questionnaire is presented in Figure 7. Therefore the general impression of the system was rated with 4.5 out of 5.0 points. A few difficulties only occurred in heuristic H1 (Feedback of current system status) and heuristic H6 (Recognition rather than Recall). Therefore you have to say that the participants received no schooling into the system.



H1… Feedback of current system status
H2… Speak the Users 'Language
H3… Reversible Actions
H4… Consistency
H5… Error Prevention
H6… Recognition rather than Recall
H7… Aesthetics and Minimalist Design
H8… Good Error Messages
H9… Hardware/Software
H10… Help and Documentation
H11… General impression of the system

Figure 7: Results of feedback questionnaire

Touch screen navigation did not cause many problems, neither for young nor elder participants, although some participants had no experiences with that kind of navigation. However, some design problems could be detected which repeatedly occurred during the tests. The main findings, including design problems, positive and negative impressions as well as suggestions for improvements are listed in the following sections.

5.3.3.1 Login/Logout

Login: Generally the login to the system seemed to be intuitive and uncomplicated. However a problem appeared to find some special character (e.g., ß). Furthermore some participants touched the "enter" button of the soft keyboard, which only led to dismiss the dialog. One participant also mentioned that he would prefer to enter the credentials on one screen.

Logout: The button to logout was easily found by all participants. Some participants unintentionally logged out of the system because they touched the wrong button (e.g., user wanted to touch therapy tab in main screen, but accidentally touched logout button).

Automatic logout: Most of the participants mentioned that automatic logout is necessary and desired. According to the participants the automatic logout should be executed between 10 – 30 minutes without any user operation.

5.3.3.2 Open task management

General: Without previous introduction about the purpose of the open task management, the functionality seemed to be confusing for most of the participants. Some participants did not even know what is meant by the term "open tasks". After a short description about the purpose of the task management the participants described the tasks management as "useful".

Task list: The task list was generally described as clearly structured. However it was not immediately understood that rooms, which contain open tasks are expanded and coloured in green.

Task symbols: The symbols related to blood glucose (BG) measurement and insulin administration were correctly interpreted by all participants and the symbol for therapy adjustment was recognised unequivocally by everyone. Some participants mentioned that some tasks (e.g., critical tasks, manually added tasks) should be marked for example by a red border.

Task details: Many participants searched for task details, especially the execution period of the proper task was often sought.

Adding new tasks: Adding a new task seemed to be one of the most difficult tasks for the participants. Almost all participants tried to first click a patient for whom the task should be created. The button "Add task" was overlooked by many participants. Also the screen for adding a task seemed to be not intuitive and many participants were not sure if the task was already saved after setting the required parameters. The button "Add task" in this screen caused even more confusion. Also the parameter 'Criticality' was not understood by the participants.

Execution period: Many participants did not understand what is meant by the term execution period. Furthermore the dialog to set the execution period seemed to be too complicated and overloaded.

Future tasks: Some participants would also prefer an additional view where current tasks as well as tasks in the near future are available.

Countdown: 1 participant suggested using a countdown near the blood glucose measurement task to guarantee control measurement.

Lifetime of tasks: One participant mentioned that tasks should be available until performed. If a task is not performed (e.g., 1 hour after execution time) the system should remind user with a dialog.

5.3.3.3 Input methods

Wheel picker: According to the input method using the wheel picker, the participants' opinions are divided. Where 2 participants prefer an input of numeric values with a soft keyboard, 8 participants felt satisfied with the wheel picker. 1 participant prefers both input options. A striking feature of the usability study was that at the first tasks almost each participant had problems in handling with the wheel picker, but after a few attempts the participants became much quicker and more skilful. The

biggest problem seemed to be the setting of very high or very low blood glucose values, because of the wide range of this value. 1 participant also mentioned that the direction of the wheel should be reversed.

Context Menu (long click on list item): Almost all participants needed a hint to find the context menu in the patient list by long-clicking the proper list item. However, most of the participants mentioned that long click events are acceptable if they are known. It was striking that the participants had big problems closing an accidentally opened context menu because there was no abort button available.

Back button: Sometimes participants were searching for a back button, or were clicking the back button of the device, to go back to previous operations or screens. However in most cases, the back button of the device was disabled for safety reasons (Android back button forces the current Android Activity to finish).

Slide event: At the patient details in the main screen of a patient's glucose management the system offers some supplement (not so important) information about the patient (weight, pre-therapy). No participant was able to find that hidden functionality.

Tab navigation: Especially at the main screen some participants had problems to navigate through tabs, which resulted in starting unintentionally operations (e.g., 1 participant wanted to touch "Therapy" tab, but logged out of the system)

Order insulin: Some participants did not recognize that the list, which contains available types of insulin, is scrollable.

Close Dialog: Some participants repeatedly tried to close alert dialogs by touching the symbol to the left of the dialog title.

5.3.3.4 Edit/delete recent activities

History: In most cases the purpose of the history was misunderstood by the participants. The participants expected a history of the selected patient, not a documentation of all activities. However the history of a patient can already be displayed in the therapy profile and therapy table in an appropriate manner. The "History" button in the menu bar tends rather to confusion than to a helpful feature.

Show Activity details: Almost all participants intuitively touched a chart point at the therapy profile to get information (e.g., performer) about an executed activity. However at the current state only comments can be displayed this way.

Edit/Delete activities: Almost all participants prefer to edit recent activities directly in the therapy profile (not in the history). Important parameters to edit are: time, nutrition, value, comment. Some participants also mentioned that editing an activity should be accompanied by a mandatory comment. One participant also mentioned that entering a comment to a BG measurement retrospectively would be enough (BG value must not be editable).

5.3.3.5 Decision support

Insulin Administration: The presentation of the calculated suggestions related to current insulin doses was generally evaluated as clearly structured. Some participants mentioned that the bolus division should be more traceable. Some participants also mentioned that the labels for Basal and Bolus Insulin ("Bolus Insulin", "Basal Insulin") are legible and should be coloured black. The participants were also asked if some additional information about the dose algorithm should be available (e.g., calculation table) in order to provide a better understanding about the decision support functionality. However, only 1 participant called this feature "advantageous".

Daily Dose Initialization: Only 1 participant had problems to find out how to use the DSS functionality to calculate the initial daily insulin dose. Another participant mentioned that the default creatinine value should be less than 2. 1 participant, who was not familiar with the REACTION algorithm, was confused because the DSS suggested the Basal Insulin on Midday.

Therapy Adjustment: The purpose of the therapy adjustment seemed to be not clear for all participants (maybe wrong caption). 1 participant would prefer to show therapy values of last 48 hours, including night areas, during therapy adjustment and that important blood glucose values (morning

and evening values for calculation) should be marked to make dose suggestion more traceable. Another participant would like to view the actually administered basal and bolus insulin of the last 24 hours within the dose suggestion. Furthermore 1 participant mentioned that therapy adjustment should be available all time, but if today there already have been a therapy adjustment, a notification should warn the user.

Daily Dose: 1 participant mentioned that term daily consumption in therapy list should be renamed to daily dose. Furthermore 1 participant suggested providing additional information about the date and time that the daily dose was initialised or adjusted.

Therapy Tab: 1 participant mentioned that tab caption "Therapy" should be renamed to caption "Current Therapy"

Calculation Details: 1 participant mentioned that it would be advantageous to have the possibility to get more information about the calculation details on the DSS.

5.3.3.6 Special situations

Patient not available: One of the most usual special situations is that an activity cannot be performed because the patient is not available (e.g., examination). This means that the activity must be rescheduled or omitted. Therefore, according to the participants, a comment is necessary to inform other users about the absence of the activity.

5.3.3.7 Device

Dimensions: All participants, except of 1, evaluated the shape and size of the device as likeable (not smaller, not bigger). Only 1 participant would prefer a smaller device.

Amount: Two thirds of all participants would prefer 3 devices (1 per unit). One third thought that 2 devices would be sufficient.

Storage: Almost every participant agreed that the device must be storage in the head of nurses near the laptops and POCT devices, where only authorized persons have access. 1 participant mentioned that the device should be locked in the medication locker to avoid theft.

Hygiene: A conversation with the hygiene commissioner disclosed the fact that surface disinfection is required. Therefore the touch screen must be unaffected by the current moist disinfection mixture (Incidin Liquid; Incidin Plus[7]).

5.3.3.8 Visualisation and display

Therapy profile: The opinions of the participants related to the therapy profile are divided. Some participants preferred to view the therapy values in the therapy table, some participants preferred to vie the therapy values in the therapy profile. A problem which occurred by some participants was that they were not sure if the last 24 hours or the last 48 hours was presented in the therapy profile. 1 participant also mentioned that green blood glucose values (inside target range) are not easy to read. Another participant suggested marking the current day in the therapy profile.

Full Screen: At the full screen the participants criticized that it is difficult to know which date is currently presented. Furthermore the night periods are not marked, as in the therapy profile.

Selected tabs: Some participants mentioned that they are not sure which tab is selected.

Menu Bar: 1 patient mentioned that the patient list and the task list could be combined to one menu item.

5.3.3.9 Blood glucose measurement

Aborting blood glucose measurements: The system is designed so that after starting a blood glucose measurement, the user has to abort or save the measurement to start another operation. However, almost every participant had problems aborting an unintentionally started blood glucose measurement.

[7] http://www.ecolabhealthcare.at/Incidin-Liquid.html; http://www.ecolabhealthcare.at/Incidin-Plus.html

Adapt measurement time: The time between the measurement and the input of the measurement to the system can vary (e.g., emergency). Therefore the time of the input of the measurement must be adaptable.

5.3.3.10 Additional requirements

Printing functionality: 1 participant mentioned that a printing functionality would be very helpful. The print should contain the patient's therapy values in an appropriate manner.

Alarm and error history: 1 participant suggested to also provide an alarm and error history.

Sober information: 1 participant suggested providing an additional flag to indicate that a patient is to stay sober.

Set Hypo/Hyperglycaemia limits: A physician should have the possibility to set the hypo/hyperglycaemia limits of a patient.

5.3.3.11 Further findings

- Some participants had problems distinguishing between "Medication" and "Therapy"
- 1 participant was not sure if list item "Daily Insulin Dose" in therapy list was initial or current daily dose
- 1 participant mentioned that notification during changing to non-supported therapy should not contain "… deleted"
- 1 participant suggested improving of the notification of Hypoglycaemia warning after blood glucose measurement
- 1 participant mentioned that a "sort-by-name" functionality in the patient list is not necessary.

5.4 Summary of outcomes of field trials

The results from the second series of usability trials are currently being implemented, and the third series will be performed with the improved user interface in the summer of 2012 in the form of clinical trials at the Medical University Hospital Graz.

6 Validation results – Primary care prototype

The first primary care prototype is composed of 3 main logical components which share a central DB schema supported by a Microsoft SQL server DB engine.

These logical components are the medical devices + application hosting device (AHD) + observation manager (the “acquisition chain”), the clinician portal and the patient portal.

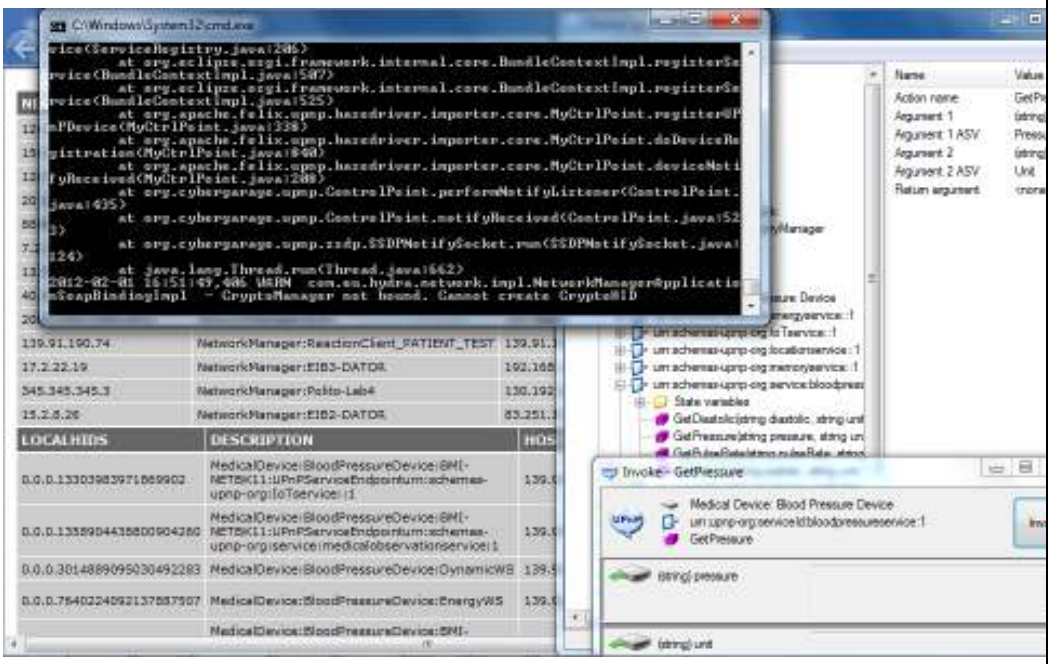
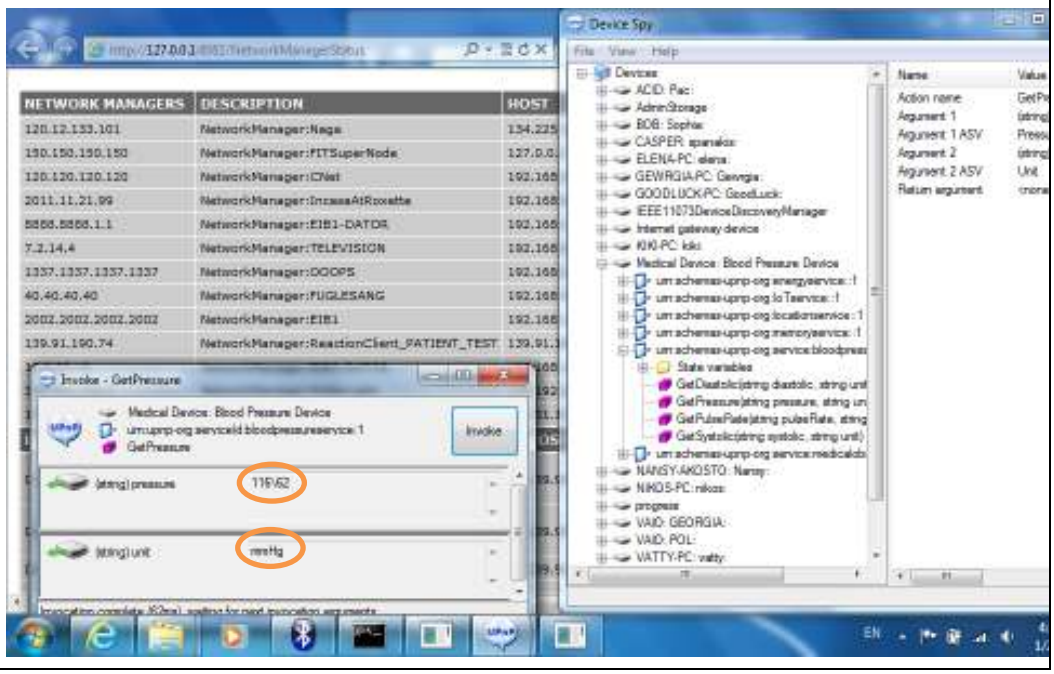
The verification phase consists of three different sessions, the unit tests, the integration tests and the system tests. The unit tests are based on test cases and have been implemented for each of these logical components.

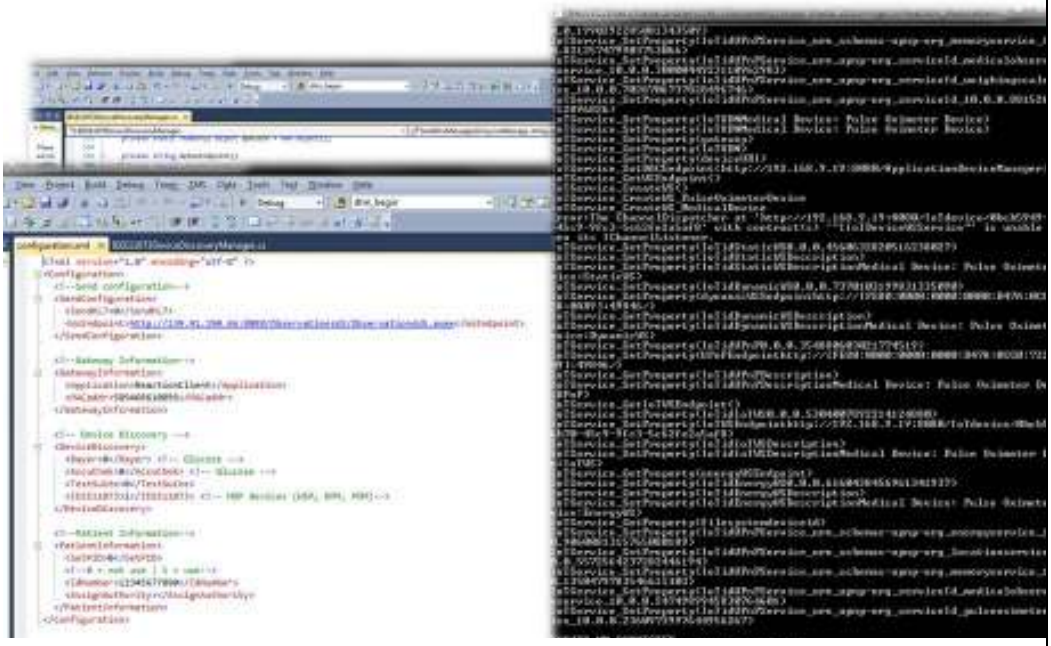
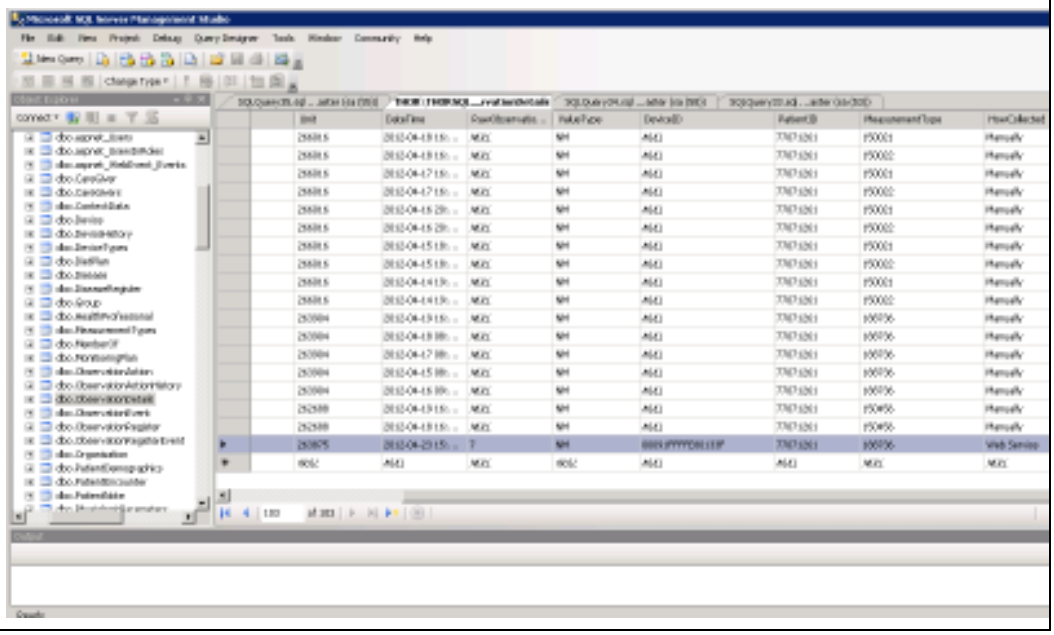
Then, the integration tests have been performed in the internal test site assembling the components together and testing their interoperability. Finally the system tests, more oriented to the verification of the requirements that the first-year prototype had to match, have been performed again in the internal test site and reported in this document.

6.1 Acquisition chain unit tests

The acquisition chain unit tests have been performed for the main functionalities available in the first primary care prototype.

The results are reported below and are related to the transfer of a measurement from the medical device to the back end server and from the back end server to the REACTION primary care database.

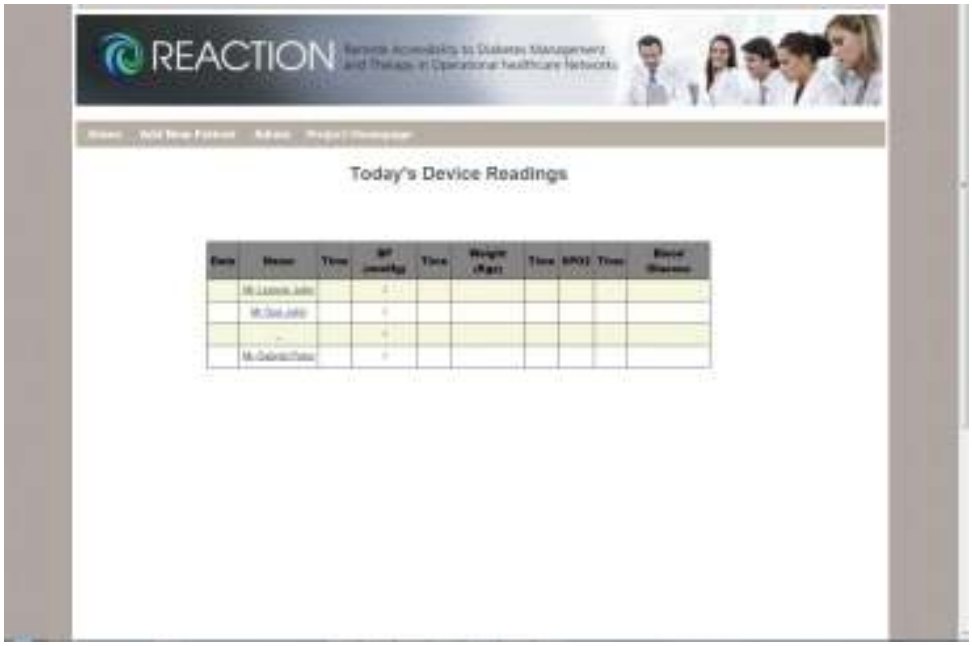
Test ID	**testTakeAMeasurementFromPatientToReactionServer**
Test Result	PASSED
Description	The user uses the medical devices to take a measurement and the REACTION middleware (LinkSmart) using a separate application the measurement can be accessed through LinkSmart. Then these measurements are ready to be sent to the database using appropriate web services. The following pictures show the results (in orange circles).
Initial Screenshot	
Screenshot of result	

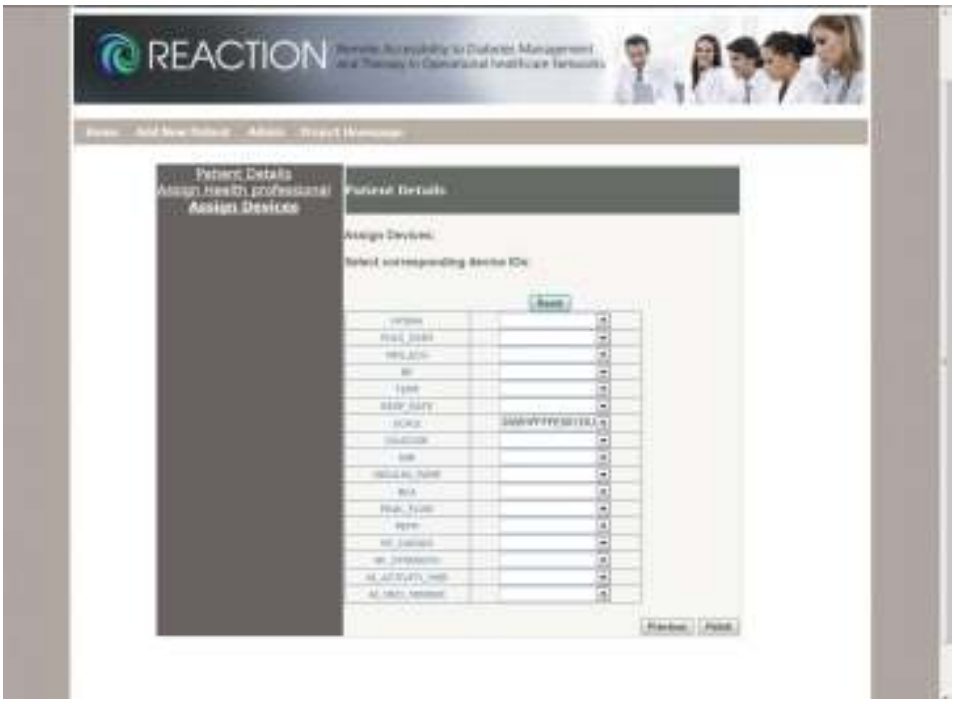
Test ID	**testInsertATakenMeasurememtToReactionDatabase**
Test Result	PASSED
Description	The users, takes a measurement with the medical device. The REACTION Hosting Client (at the patient site using the LinkSmart middleware) acquired the measurement using the Continua protocol through the device and constructs and HL7 message. This HL7 message is been used to run the ObservationWS (or MeasurementWS) in order to insert correctly the measurement data to the database.
Initial Screenshot	
Screenshot of result	

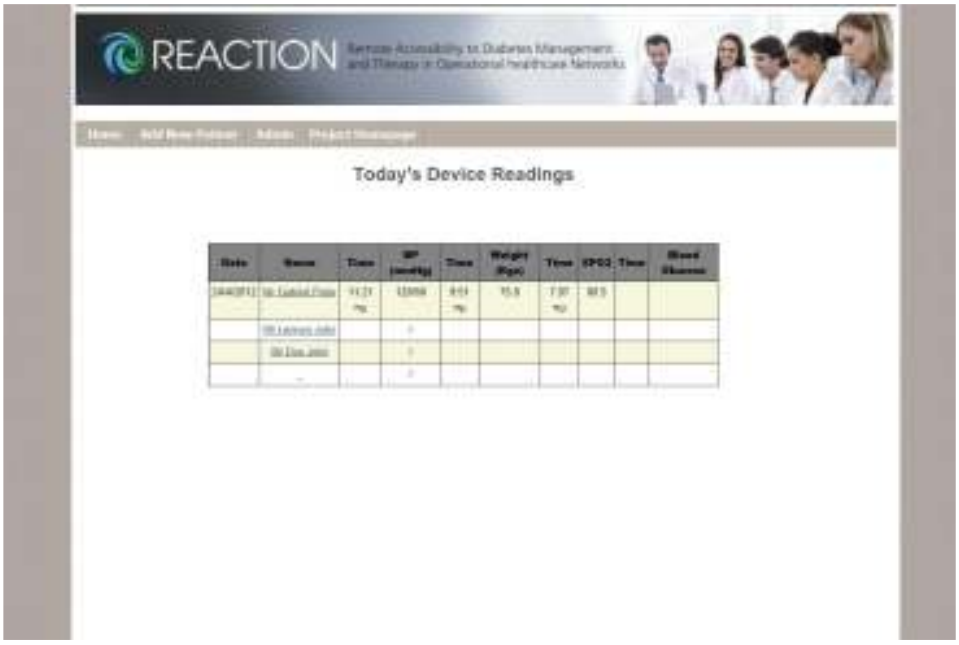
6.2 REACTION (clinician) portal unit tests

The REACTION (clinician) portal unit tests have been performed for the main functionalities available in the first primary care prototype.

The results are reported below.

Test ID	**testAddNewPatientToClinicalPortal**
Test Result	PASSED
Description	The user attempts to add a new patient to the REACTION Clinical Portal web application, by providing detailed personal information and assigning devices to the patient. Upon success the user is transferred to the Today's readings Home (main) page of the REACTION Clinical Portal.
Initial Screenshot	
Screenshot of result	

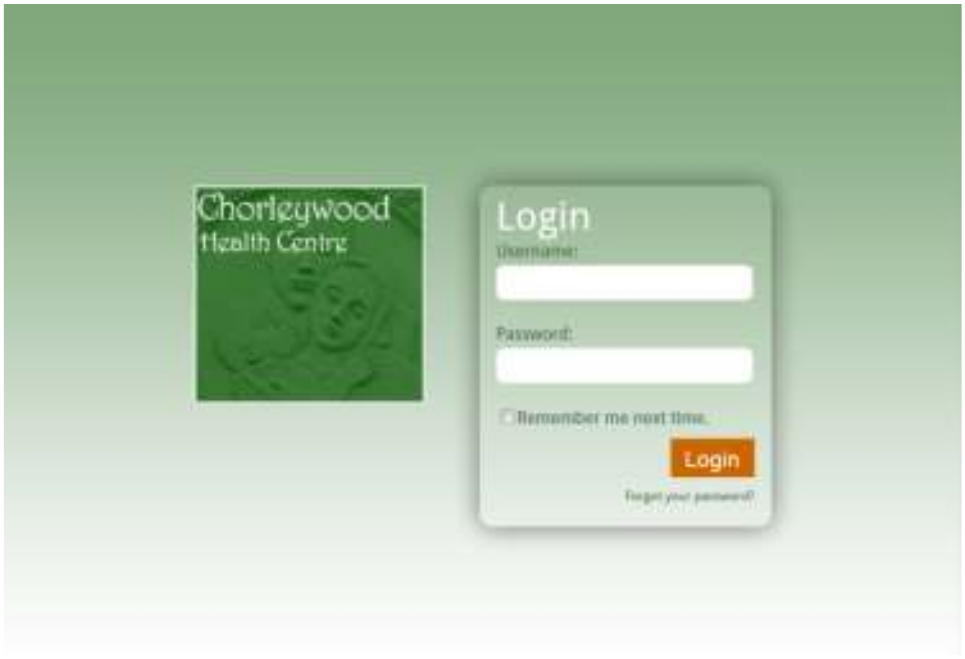
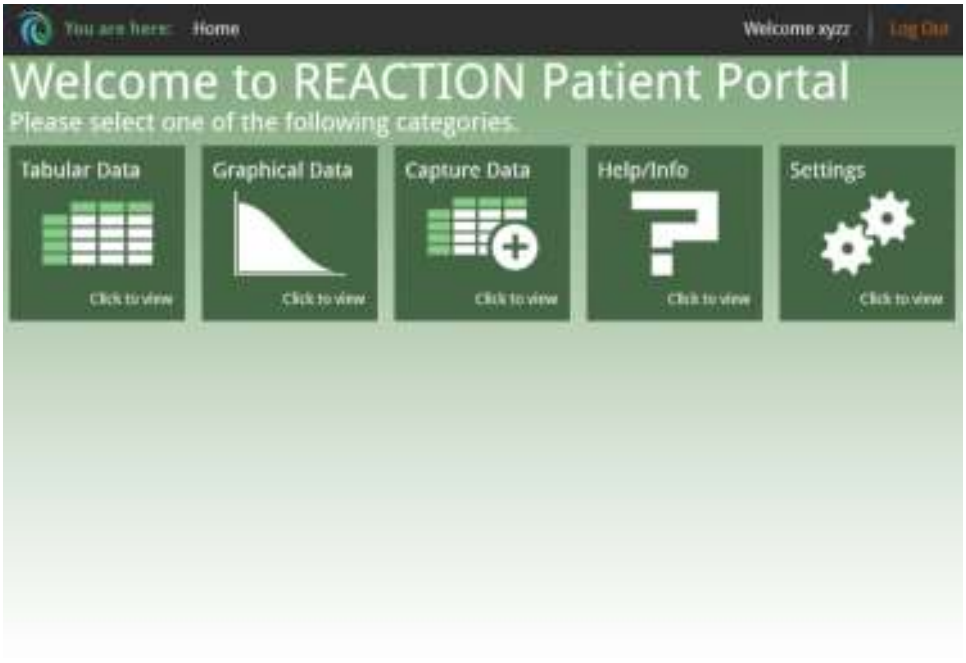
Test ID	**testAddNewDeviceToClinicalPortal**
Test Result	PASSED
Description	The admin user attempts to add a new device to the REACTION Clinical Portal web application, by providing device detailed information. Upon success the device is in list for assigning it to the patient of the REACTION Clinical Portal.
Initial Screenshot	
Screenshot of result	

Test ID	**testViewMeasurementsToClinicalPortal**
Test Result	PASSED
Description	Using the “Home” button you can see the measurements for the patients with the assigned devices. It is entitled as “Today’s Device Readings.
Screenshot of result	

6.3 Patient portal unit tests

The patient portal unit tests have been performed for the main functionalities that had to be included in the first primary care prototype.

The results are reported below.

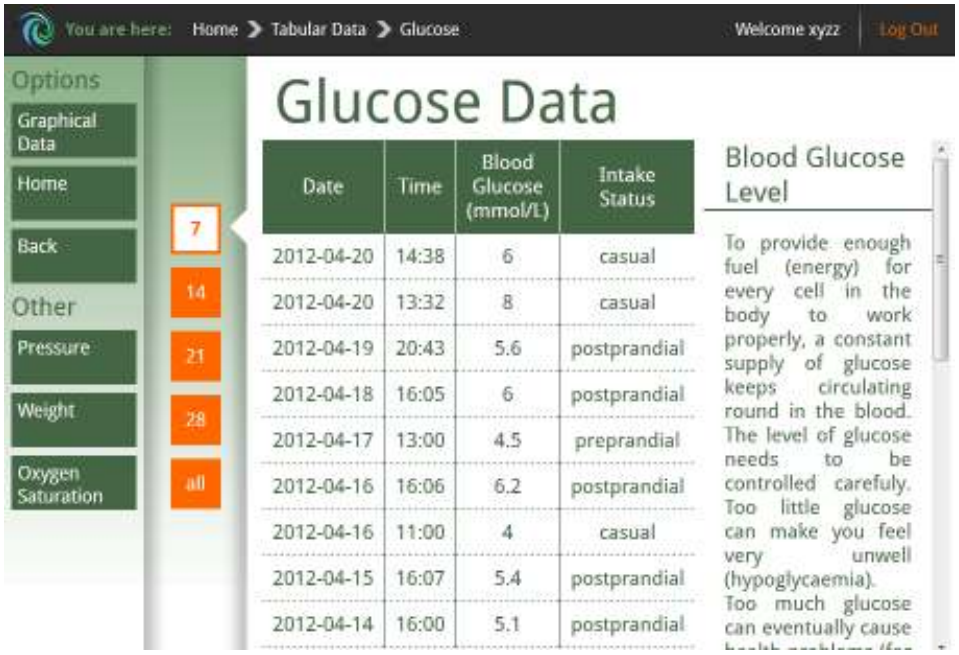
Test ID	**testLoginToPatientPortal**
Test Result	PASSED
Description	The user attempts to login to the REACTION Patient Portal web application, by providing his/hers credentials. Upon success the user is transferred to the Home (main) page of the REACTION Patient Portal.
Initial Screenshot	
Screenshot of result	

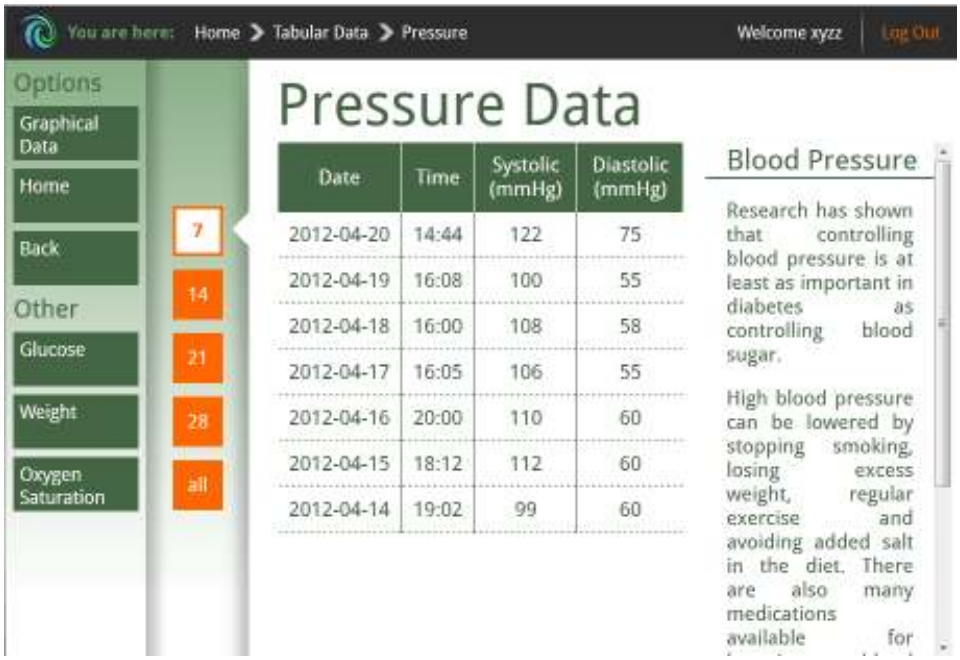
Test ID	**testCaptureManualMeasurementGlucose**
Test Result	PASSED
Description	The user inserts manually a glucose measurement. Upon success, the REACTION Patient Portal redirects the user to the main page of the manual measurements (which is the previous page) in order to simplify the procedure of adding another measurement.
Initial Screenshot	
Screenshot of result	

Test ID	**testCaptureManualMeasurementPressure**
Test Result	PASSED
Description	The user inserts manually a blood pressure measurement. Upon success, the REACTION Patient Portal redirects the user to the main page of the manual measurements (which is the previous page) in order to simplify the procedure of adding another measurement.
Initial Screenshot	
Screenshot of result	The same as in the test case with ID **testCaptureManualMeasurementGlucose**

Test ID	**testCaptureManualMeasurementWeight**
Test Result	PASSED
Description	The user inserts manually a weight measurement. Upon success, the REACTION Patient Portal redirects the user to the main page of the manual measurements (which is the previous page) in order to simplify the procedure of adding another measurement.
Initial Screenshot	
Screenshot of result	The same as in the test case with ID **testCaptureManualMeasurementGlucose**

Test ID	**testCaptureManualMeasurementOxygenSaturation**
Test Result	PASSED
Description	The user inserts manually an oxygen saturation measurement. Upon success, the REACTION Patient Portal redirects the user to the main page of the manual measurements (which is the previous page) in order to simplify the procedure of adding another measurement.
Initial Screenshot	
Screenshot of result	The same as in the test case with ID **testCaptureManualMeasurementGlucose**

Test ID	**testViewTabularDataGlucose**
Test Result	PASSED
Description	The user wants to check the blood glucose measurements that have been submitted to the platform. Thus he/she navigates to the corresponding section of the REACTION Patient Portal (Home->Tabular Data ->Glucose) where the data are presented in tabular format.
Initial Screenshot	None
Screenshot of result	

Test ID	**testViewTabularDataPressure**
Test Result	PASSED
Description	The user wants to check the blood pressure measurements that have been submitted to the platform. Thus he/she navigates to the corresponding section of the REACTION Patient Portal (Home->Tabular Data ->Pressure) where the data are presented in tabular format.
Initial Screenshot	None
Screenshot of result	

Test ID	**testViewTabularDataWeight**
Test Result	PASSED
Description	The user wants to check the weight measurements that have been submitted to the platform. Thus he/she navigates to the corresponding section of the REACTION Patient Portal (Home->Tabular Data ->Weight) where the data are presented in tabular format.
Initial Screenshot	None
Screenshot of result	

Test ID	**testViewTabularDataOxygenSaturation**
Test Result	PASSED
Description	The user wants to check the oxygen saturation measurements that have been submitted to the platform. Thus he/she navigates to the corresponding section of the REACTION Patient Portal (Home->Tabular Data ->Oxygen Sat.) where the data are presented in tabular format.
Initial Screenshot	None
Screenshot of result	

Test ID	**testViewGraphicalDataGlucose**
Test Result	PASSED
Description	The user wants to check the blood glucose measurements that have been submitted to the platform. Thus he/she navigates to the corresponding section of the REACTION Patient Portal (Home->Graphical Data ->Glucose) where the data are presented in graphical format.
Initial Screenshot	None
Screenshot of result	

Test ID	**testViewGraphicalDataPressure**
Test Result	PASSED
Description	The user wants to check the blood pressure measurements that have been submitted to the platform. Thus he/she navigates to the corresponding section of the REACTION Patient Portal (Home->Graphical Data ->Pressure) where the data are presented in graphical format.
Initial Screenshot	None
Screenshot of result	

Test ID	**testViewGraphicalDataOxygenSaturation**
Test Result	PASSED
Description	The user wants to check the oxygen saturation measurements that have been submitted to the platform. Thus he/she navigates to the corresponding section of the REACTION Patient Portal (Home->Graphical Data ->Oxygen Sat.) where the data are presented in graphical format.
Initial Screenshot	None
Screenshot of result	

Test ID	**testViewGraphicalDataWeight**
Test Result	PASSED
Description	The user wants to check the weight measurements that have been submitted to the platform. Thus he/she navigates to the corresponding section of the REACTION Patient Portal (Home->Graphical Data ->Weight) where the data are presented in graphical format.
Initial Screenshot	None
Screenshot of result	

6.4 Integration tests

The main purpose of the integration tests is to test the aggregation of the main components and to verify if they work properly together.

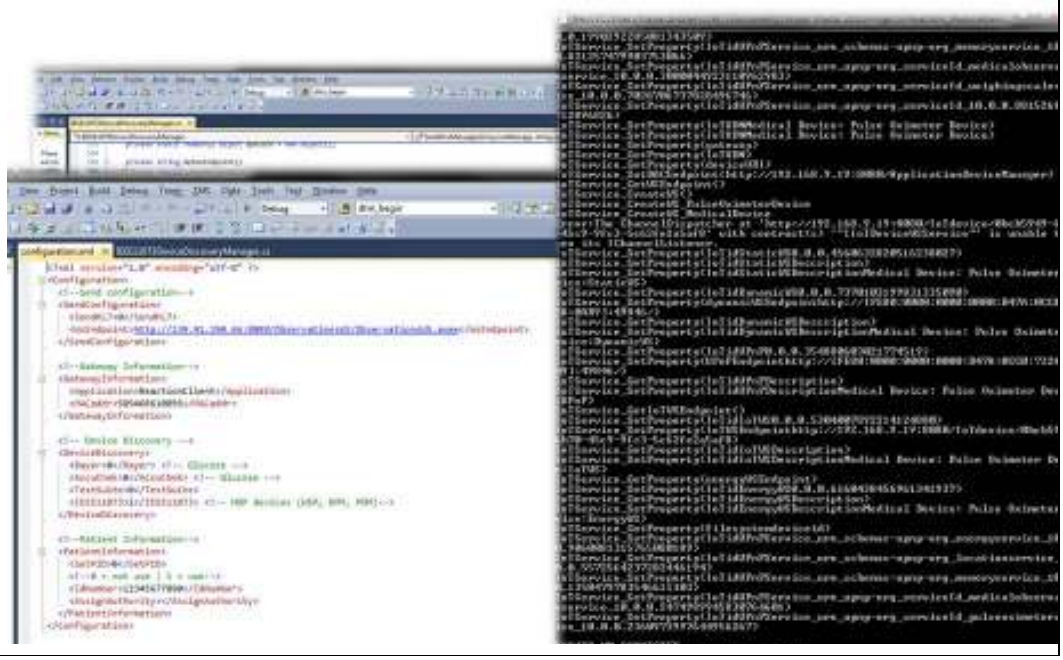
Integration testing takes as its input the components that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

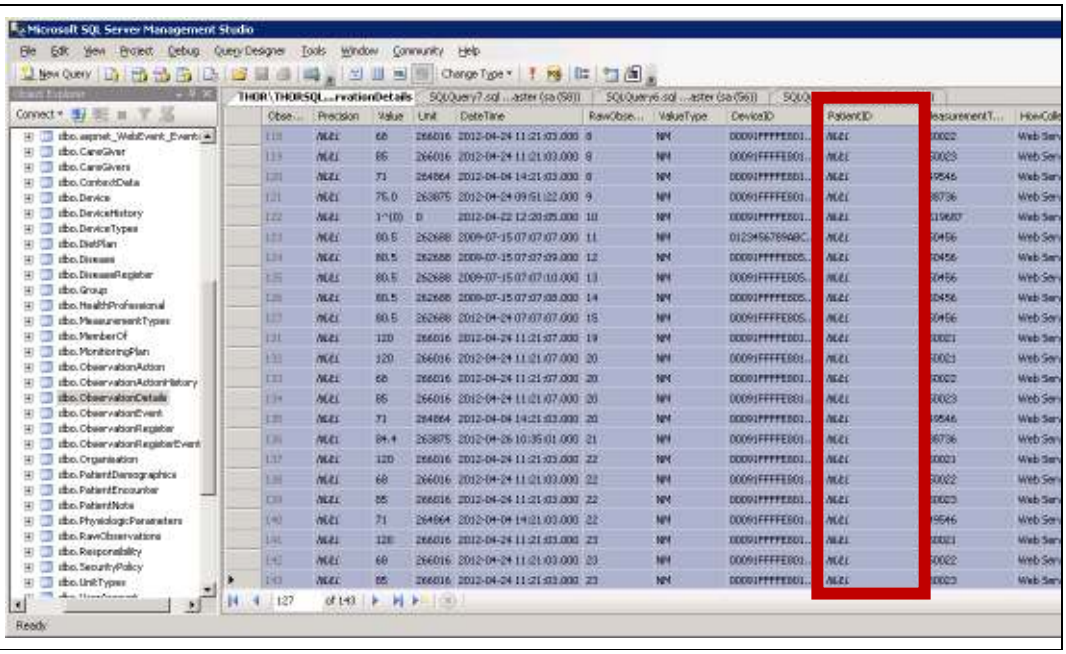
For the primary care environment the integration tests have to be performed with the acquisition chain, the REACTION portal and the patient portal. The main integration tests performed for the first primary care prototype are focused on the visualization of the measurements acquired through the acquisition chain using the REACTION portal and the patient portal. Further tests have not been foreseen in this phase since the user management has not been implemented yet in the REACTION portal.

There are still problems in these first integration tests, since the patient ID has not been properly filled in when the measurements have been inserted in the observation table and thus it is not possible to understand which patient is the owner of the acquired measurements.

The device ID cannot be used since, according to the detailed specifications, the same device is not allocated “forever” to a patient but only for a certain period after that it is de-allocated and assigned to another patient. The ObservationWS has (using the device ID) to retrieve the patient ID from the database and then to store in the observation table also the patient ID.

UBRUN, the partner responsible for the ObservationWS, has been informed in order to solve this issue.

Test ID	**testInsertAutomaticAMeasurememtFromDeviceUsingHydraToObservationWStoReactionDatabase**
Test Result	FAILED
Description	The user takes a measurement with the medical device. The REACTION Hosting Client (at the patient site using the LinkSmart middleware) constructs an HL7 message and, using the ***ObservationWS***, inserts the measurement data in the database. The data are parsed and inserted in the ObservationDetails table. The observationWS does not insert the correct PatientID and thus the REACTION Patient Portal is not able to represent the data. Retrofit and feedback for this has been given in order to fix this issue.
Initial Screenshot	

Screenshots of result	

6.4.1 System tests

System testing is testing has to be conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box

testing, and as such, should require no knowledge of the inner design of the code or logic. All the functional requirements that were determined in *ID2-6-2 Prototype Application Specification 2* were filtered including only the ones impacting on the first primary care prototype.

Even if a complete system testing could not be performed, because of the failure in the integration phase, some preliminary tests were performed in order to check which of the functional requirements has been satisfied (or expected to be satisfied once the issues raised in the integration phase would be solved) and the level of the satisfaction. The following four levels of satisfaction have been defined and assigned in each of the functional requirements involved.

- Fully satisfied
- Partially satisfied
- Improperly satisfied
- Not satisfied

The functional requirements related to the primary care prototype were selected from the JIRA requirements project with the exception of the resolved ones with resolution “duplicate”, out of scope” or “cannot be implemented” and inspected one by one.

The total number of requirements inspected was 64 and each one was classified according to the current level of satisfaction. The distribution is shown in Figure 8.

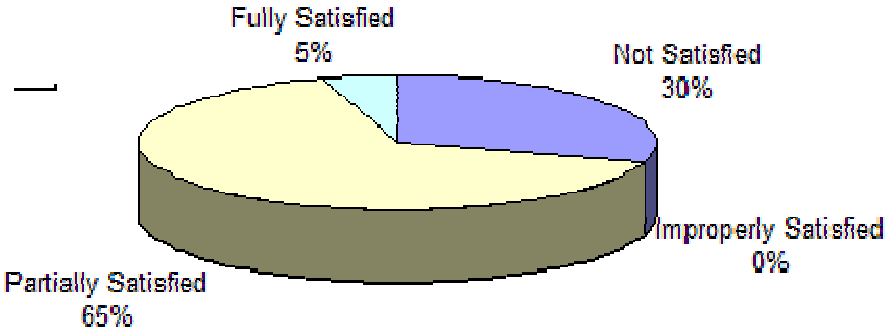


Figure 8: Satisfaction of requirements for first primary care prototype

6.5 Summary of validation results

Not yet available.

7 Impact assessment

7.1 Impact on architecture

The impact of this iteration on the REACTION architecture is minor, meaning that the SOA approach taken is maintained and supports the functionalities wanted for the REACTION system. The platform itself is under continuous development and feature refinement and can therefore not be considered finalised yet. On the other hand, through the experience gained in this iteration, the consortium now understands that the REACTION platform needs to be made more independent of the database structure. Some platform components must share a common view of some central objects, e.g., patient, measurement, device and context in order to ease, for example, the aspects of seamless integration of components and the consumption of measurements for the different sphere applications. These objects should be made available as services within the platform, allowing their fast integration and providing the consequent dynamic and goal-oriented application development. During iteration three, emphasis will be on a more distinct service layer in the REACTION architecture model which will contain all of these types of services.

7.2 Compliance with Medical Device Directive

7.2.1 Medical Device Directive

The revised Medical Device Directive came into effect in March 2010. Medical software now may have to comply with the same rules as medical devices. The question that must be asked is: If we are going to apply a medical device to a human being, does the device comply with the medical device directive? If yes, how can we prove it? Which standards do we use? Auto-certification should be possible but if the regulatory authorities do not trust us, we need to provide documentation. Partners should have a quality management system for medical device development. But while this is quite normal for some companies and for academic or research centres, for industrial partners not working in the medical domain it is very likely that they do not have it. However, the REACTION consortium should demonstrate that the REACTION platform and the developed sensors do not harm patients (e.g., automatic transfer of glucose measurements in REACTION is correct and not wrong). Risk analysis and software life cycle management have to be performed. It is necessary to identify the risks and the actions taken against the risks (for each crucial step).

7.2.2 European Union legal framework and definition

Rules relating to the safety and performance of medical devices are harmonised in the EU and consists of 3 directives:

- Directive 90/385/EEC regarding active implantable medical devices
- Directive 93/42/EEC regarding medical devices
- Directive 98/79/EC regarding in vitro diagnostic medical devices

They aim at ensuring a high level of protection of human health and safety. These 3 main directives have been supplemented over time by several modifying and implementing directives. For REACTION the most relevant directive is the Directive 93/42/EEC. It was reviewed and amended by the 2007/47/EC and a number of changes were made (e.g., software for medical applications has become a medical device).

Directive 2007/47/EC defines a medical device as: "*any instrument, apparatus, appliance, software, material or other article, whether used alone or in combination, including the software intended by its manufacturer to be used specifically for diagnostic and/or therapeutic purposes and necessary for its proper application, intended by the manufacturer to be used for human beings*". Devices are to be used for the purpose of:

- Diagnosis, prevention, monitoring, treatment or alleviation of disease
- Diagnosis, monitoring, treatment, alleviation of or compensation for an injury or handicap

- Investigation, replacement or modification of the anatomy or of a physiological process
- Control of conception
- The government of each Member State has been required to transpose the Medical Device Directive 2007/47/EC into National Law by March 21, 2010.

This means that medical software is now classified as a medical product by the Medical Device Directive. A new regime is in force governing all medical device software development for all classes of device.

Previous software safety standards were best suited to medical devices with low levels of risk, as opposed to products where software failure could be extremely serious and result in death. As more electronic products have become dependent on embedded software, the focus has shifted to the reliability of software systems within the devices and the associated risks at all levels of usage. As a result, the new EN/IEC 62304 standard has emerged as a global benchmark for management of the software development lifecycle.

IEC 62304 is a harmonised standard for software design in medical products adopted by the European Union and the United States. Because the standard is “harmonised”, medical device manufacturers adopting it will satisfy the essential requirements contained in Medical Devices Directive 93/42/EEC (MDD) with amendment 2007/47/EC as related to software development. This is the least onerous route to ensuring compliance with the MDD.

Designing to IEC 62304 ensures that quality software is produced by means of a defined and controlled process of software development (see Figure 9). This standard provides a framework of life cycle processes with activities and tasks necessary for the safe design and maintenance of medical device software.

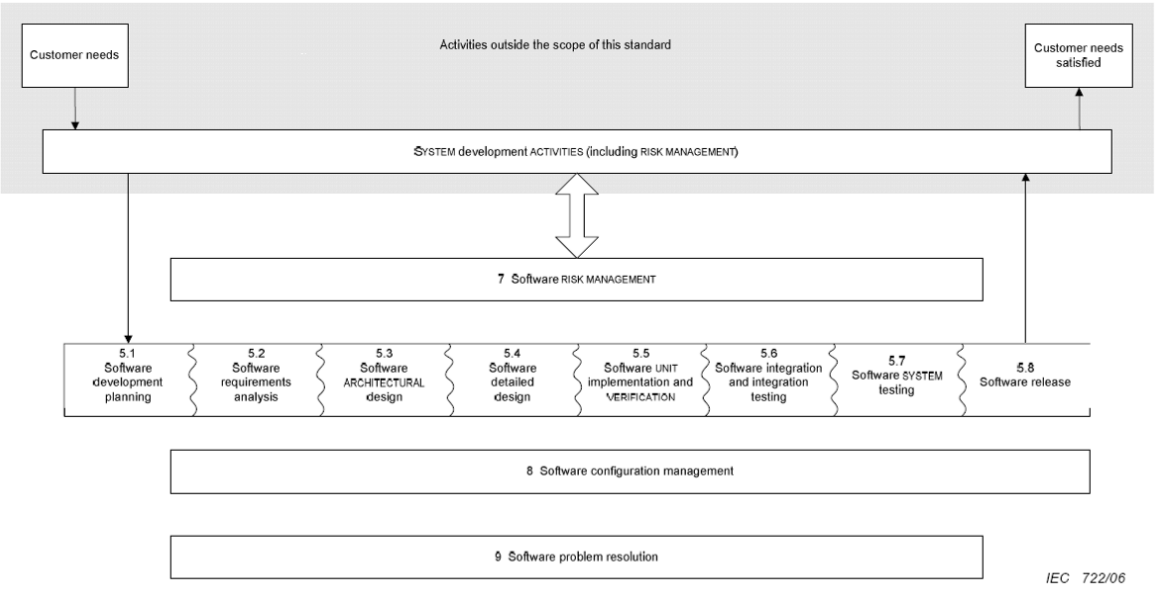


Figure 9: Overview of software development *processes* and *activities* as per IEC 62304

IEC 62304 is a well considered, logical standard for developing safety critical and high reliability software for medical devices. Now that this standard has been adopted it would be very difficult for a medical device software developer to justify any equivalent approach that meets the requirements of the MDD, without effectively complying with this standard.

The REACTION in-hospital Glucose Management System, which will assist professionals (physicians and nurses) in the glucose management of patients at general wards in the hospital must be considered as a medical device. Therefore the system, which consists of software as well as hardware, must fulfil the essential requirements set out in the Medical Device Directive (2007/47/EC). In order to prove its compliance with the MDD (for the Ethics committee and the legal authorities) the development process will be based on IEC 62304.

The entire compliance with the essential requirements can more easily be proved through the adoption of existing standards as shown in Figure 10.

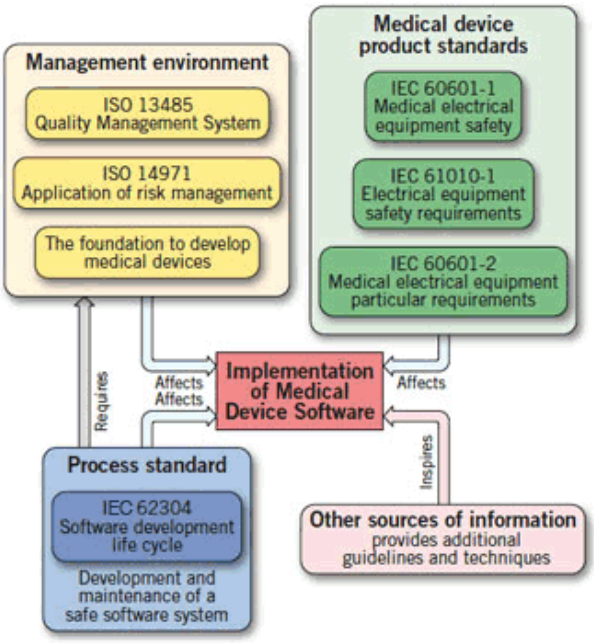


Figure 10: Compliance process for Medical Device Software and its relationship to standards

7.2.3 Risk analysis and assessment

Risk analysis will be performed with key users and experts and main results (identified risks/measures) reported. The first session (risk identification) has already been conducted.

7.2.3.1 Risk analysis for hardware and software design

Medical product designers have used risk management techniques to help reduce the risks associated with device hardware. BS/EN/ISO 14971 has traditionally been adopted as the base standard for risk management for medical devices. The 2007 version of this standard[8] is considerably extended from its previous version, and the techniques described are now intended to be applied to both software and hardware systems.

The approach that should be taken is to consider the risks posed by the medical device as a whole, before the software/hardware split has been decided. Hardware risk analysis can then run alongside software risk analysis to define the required safety systems for the device.

7.2.3.2 Risk management process

The manufacturer shall establish, document and maintain throughout the life cycle an ongoing process for identifying hazards associated with a medical device, estimating and evaluating the associated risks, controlling these risks, and monitoring the effectiveness of the controls (see Figure 11).

[8] http://www.isosert.ru/isosert_iso_14971.pdf

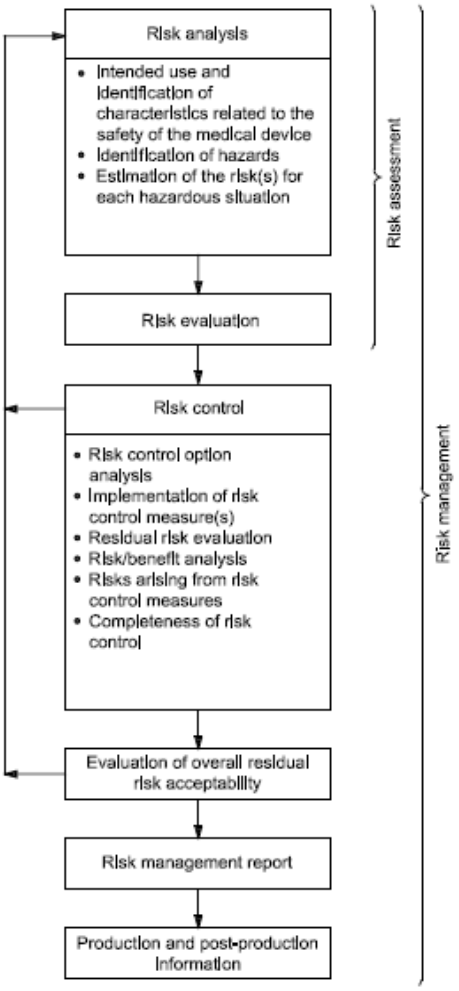


Figure 11: A schematic representation of the risk management process

The ISO 14971 standard defines the following elements in the process:

- Risk analysis
- Risk evaluation
- Risk control
- Production and post-production information.

The risk analysis process is started with a description of the intended use and characteristics related to the safety of the medical device. In the next step potential hazards are identified and its risks are estimated for hazardous situations.

Both components of a risk, probability and consequence, are analysed separately for the estimation of a hazard (see Figure 12). For risk control there will be a stepwise approach to reduce risk:

- Inherent safety by design
- Protective measures in the medical device itself or in the manufacturing process
- Information for safety.

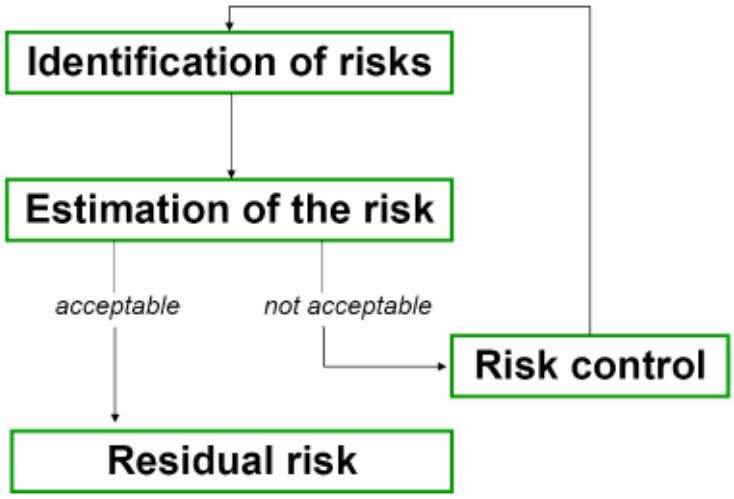


Figure 12: Process of risk analysis, evaluation and control

This means that if practicable, the medical device should be designed to be inherently safe. If this is not practicable, then protective measures such as barriers or alarms are appropriate. The least preferred protective measure is a written warning or contra-indication. It is recognised that one possible results of the risk control option analysis could be that there is no practicable way of reducing the risk to acceptable levels. In this case, a risk/benefit analysis can be carried out to determine whether the benefit of the medical device outweighs the residual risk.

The block diagram in Figure 13 was used to identify possible risks of the system. It is important to mention that the system will only give an advice to the professional user which means that in any case the user will do a plausibility check before insulin is going to be injected. The risks identified so far are summarised in Table 3.

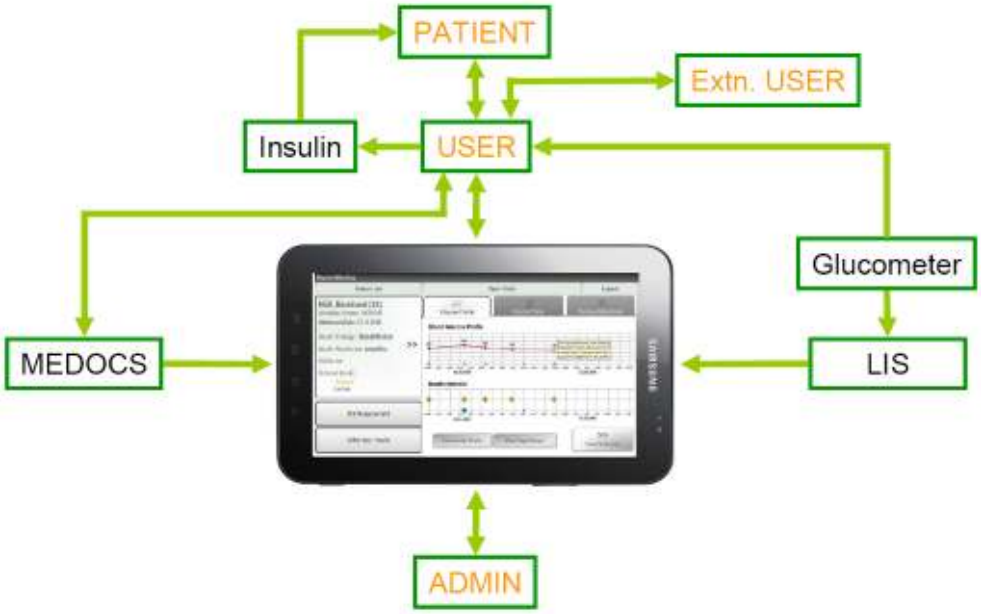


Figure 13: Block diagram of the REACTION in-hospital Glucose Management System

Component, function, process	Error	Cause of error, consequence	Hazard	Risk control
Data acquisition	Input data error	Login is not possible (e.g., password forgotten); no possibility to calculate insulin dosing	Hyper-Hypoglycaemia	Login for emergency cases; (has the system a life-sustaining function?)
Data acquisition	Input data error	Entry of data (e.g., name) is not possible because of limited space; to open an account for the patient is not possible or wrong account will be created	Hyper-Hypoglycaemia	Software will be adjusted to the hospital information system
Data acquisition	Input data error	Wrong glucose data (input error), wrong calculation of insulin dose	Hyper-Hypoglycaemia	Check of plausibility (define check criteria),training
Data acquisition	Input data error	Wrong/missing data of nutrition, wrong calculation of insulin dose	Hyper-Hypoglycaemia	Training
Data acquisition	Input data error	Wrong/missing data of insulin, wrong calculation of insulin dose	Hyper-Hypoglycaemia	Reminder/alarm system, training
Data acquisition	Input data error	Wrong/missing data, wrong calculation of insulin dose	Hyper-Hypoglycaemia	Check of plausibility (define check criteria),training, Log-function for traceability
Automatic data acquisition	erroneous functioning	Wrong glucose data (transfer error), wrong calculation of insulin dose	Hyper-Hypoglycaemia	Check of plausibility (define check criteria),test of the system, checksum
Data acquisition	erroneous functioning	Simultaneous entry of data, wrong display of data, wrong dosing of insulin	Hyper-Hypoglycaemia	Software is able to detect simultaneous entry of data, check of plausibility,
Data acquisition	erroneous functioning	Offline, loss of data; wrong calculation of insulin dose (no actual data can be entered into the system)	Hyper-Hypoglycaemia	Display the status of the system (i.e., offline, restricted use, etc.)
Data acquisition	erroneous functioning	Offline, loss of data; wrong calculation of insulin dose at a later point in time – no data cab be transferred to the server	Hyper-Hypoglycaemia	Display the status of the system (i.e., offline, restricted use, etc.)
Data acquisition; readout of data	erroneous functioning	The system is not available – data cannot be retrieved; dosing of insulin is not possible	Hyper-Hypoglycaemia	Backup of the data must be available
Data acquisition; readout of data	erroneous functioning	Decision Support is not available; wrong dosing	Hyper-Hypoglycaemia	Standard operating procedures (SOPs) available

Component, function, process	Error	Cause of error, consequence	Hazard	Risk control
Data acquisition; readout of data	Input data error	User interchanges two patients; wrong dosing of insulin for two patients	Hyper-Hypoglycaemia	
Data acquisition; readout of data	Input data error	Wrong login; Traceability of entry of data is not correct	Wrong traceability	
Data acquisition; readout of data		Someone hacks into the system and changes data; wrong dosing of insulin	Hyper-Hypoglycaemia, Infringement of data protection	Encrypted data transfer; non-public WLAN
Display of information	erroneous functioning	Wrong display of data (e.g., units on different devices are not the same); wrong calculation of insulin dose	Hyper-Hypoglycaemia	Validation of the system and subsystems, check conformity of the units, note in the instruction manual
Display of information	erroneous functioning	Wrong reminder/alarm; user does not trust the system – system will not be used or there is no attention to the alarms; additional measurements will not be performed	Hyper-Hypoglycaemia	
Display of information	erroneous functioning	Used insulin is not available in the system; wrong dosing of insulin	Hyper-Hypoglycaemia	
Display of information	erroneous functioning	The implemented algorithm does not work; wrong dosing of insulin	Hyper-Hypoglycaemia	Clinical validation of the system; notice: ”For clinical investigations only”
Display of information	System does not work properly (erroneous functioning)	System is too slow; wrong calculation of insulin dosing	Hyper-Hypoglycaemia	
Display of information	Limited usability	Display of information is not clear (e.g., foreign language, new phrases, …); wrong calculation of insulin dosing	Hyper-Hypoglycaemia	Usability Tests
Display of information	Limited usability	Display of information for the particular user not clear – special needs (colour-blindness, limited acoustic perception, …); wrong entry of data; wrong calculation of	Hyper-Hypoglycaemia	Usability Tests, Instruction manual

Component, function, process	Error	Cause of error, consequence	Hazard	Risk control
		insulin dosing		
Input of data	Limited usability	Entry of data of the user not easily possible (size of the finger is too big for keypad), wrong entry of data; wrong calculation of insulin dosing	Hyper-Hypoglycaemia	Usability Tests, Instruction manual
tablet PC	Misuse	Theft of the system. Patient related data can be read by not authorised user	Infringement of data protection	Automatic logout of the system after a predefined period
tablet PC	Misuse	Theft of the system. Patient related data can be changed by not authorised user; wrong calculation of insulin dosing	Hyper-Hypoglycaemia	Automatic logout of the system after a predefined period; log function of the system
tablet PC	erroneous functioning	Using the system for other software applications which make the system instable – wrong calculation of insulin dose	Hyper-Hypoglycaemia	Block of software which is not necessary for the use
tablet PC		Disinfection of the system destroys display – error in reading, wrong calculation of insulin dose	Hyper-Hypoglycaemia	Device is certified for clinical use
tablet PC		Glossy display - error in reading, wrong calculation of insulin dose	Hyper-Hypoglycaemia	Adequate lighting, notice in the instruction manual
tablet PC	erroneous functioning	Rechargeable batteries are empty; the system is not available; data cannot be retrieved; dosing of insulin is not possible	Hyper-Hypoglycaemia	Display of power source; regular charging of the system
tablet PC	erroneous functioning	System actively interferes with devices in the hospital; wrong therapy	Wrong treatment	Test of electromagnetic compatibility, notice in the instruction manual
tablet PC	erroneous functioning	System interferes with devices in the hospital; wrong therapy wrong calculation of insulin dose	Hyper-Hypoglycaemia	Test of electromagnetic compatibility, notice in the instruction manual
tablet PC	Mechanical damage	Mechanical damage of the system; does not work properly/at	No treatment is possible	Robust system

Component, function, process	Error	Cause of error, consequence	Hazard	Risk control
		all; wrong/no calculation of insulin dose		
tablet PC, accessories kit	erroneous functioning	The system is not available (absent or not working); wrong/no calculation of insulin dose	Hyper-Hypoglycaemia	More than one device (incl. auxiliary equipment) is available
server	erroneous functioning	Breakdown of the server; no data for insulin dosing is available; wrong/no calculation of insulin dose	Hyper-Hypoglycaemia	Backup system
maintenance	erroneous functioning	Software update is not available on all devices; wrong calculation of insulin dose	Hyper-Hypoglycaemia	
maintenance	erroneous functioning	No contact person; wrong use of the system	Hyper-Hypoglycaemia	Training, instruction manual,
maintenance	erroneous functioning	Erroneous functions are not detected – no warning; wrong calculation of insulin dose	Hyper-Hypoglycaemia	Alarms, self checks, plausibility checks

Table 3: Risk table

8 Appendix A

In-hospital domain tests (back end) – Summary of domain tests

Group name	Methods
AuditTrailDao	AuditTrailDao_createAuditTrailTest.testCreateAuditTrail_deactivateAuditTrail _user_exists() AuditTrailDaoTransformTest.testToAuditTrailVO_GlucoManUser_linked() AuditTrailDao_createAuditTrailTest.testCreateAuditTrail_createAuditTrail_ user_exists() AuditTrailDaoTransformTest.testAuditTrailVOToEntity() AuditTrailDaoTransformTest.testToAuditTrailVO_GlucoManUser_not_linked()
DrugDao	DrugDaoTransformTest.testDrugVOToEntity_entity_exists() DrugDaoTransformTest.testDrugVOToEntity() DrugDaoTransformTest.testToDrugVO() DrugDao_findByCriteriaTest.testSuccessPath()
EnrolmentRecordDao	EnrolmentRecordDaoTransformTest.testToEnrolmentRecordVO() EnrolmentRecordDaoTransformTest.testEnrolmentRecordVOToEntity() EnrolmentRecordDaoTransformTest.testToEnrolmentRecordDetailVO() EnrolmentRecordDaoTransformTest.testEnrolmentRecordDetailVOToEntity()
EnrolmentDao	EnrolmentDao_findCurrentEnrolmentForPatientTest.testFindCurrentEnrolment _no_enrolments() EnrolmentDao_findCurrentEnrolmentForPatientTest.testFindCurrentEnrolment_ one_enrolment()
GlucoManUserDao	GlucoManUserDao_findByPhraseTest.testSuccessPath_no_parameter() GlucoManUserDao_findByNameTest.testSuccessPath() GlucoManUserDaoTransformTest.testUserVOToEntity_user_with_userName_exists() GlucoManUserDaoTransformTest.testUserVOToEntity_no_user_with_userName_exists() GlucoManUserDao_findByPhraseTest.testSuccessPath() GlucoManUserDaoTransformTest.testUserListVOToEntity() GlucoManUserDao_findByPhraseTest.testSuccessPath_only_phrase() GlucoManUserDaoTransformTest.testToUserListVO() GlucoManUserDaoTransformTest.testToUserVO() GlucoManUserDao_findByPhraseTest.testSuccessPath_only_deactivated()
MeasurementRecordDao	MeasurementRecordDaoTransformTest.testToMeasurementDetailVO() MeasurementRecordDaoTransformTest.testMeasurementRecordVOToEntity() MeasurementRecordDao_findByCriteriaTest.testAscendingCriteriaOrderDirection() MeasurementRecordDaoTransformTest.testToMeasurementRecordVO() MeasurementRecordDao_findByCriteriaTest.testSuccessPath()

	MeasurementRecordDaoTransformTest.testMeasurementDetailVOToEntity()
MeasurementTypeDao	MeasurementTypeDaoTransformTest.testMeasurementTypeVOToEntity() MeasurementTypeDaoTransformTest.testToMeasurementTypeVO()
MedicationRecordDao	MedicationRecordDao_findByCriteriaTest.testSuccessPath() MedicationRecordDaoTransformTest.testToMedicationRecordDetailVO() MedicationRecordDaoTransformTest.testToMedicationRecordVO() MedicationRecordDaoTransformTest.testMedicationRecordVOToEntity() MedicationRecordDaoTransformTest.testMedicationRecordDetailVOToEntity()
Patient	PatientTest.testGetCurrentVisit_oneVisit() PatientTest.testGetCurrentVisit_noVisit() PatientTest.testGetCurrentVisit_twoVisits()
PatientDao	PatientDaoTransformTest.testToPatientsListVO_no_currentVisit_no_currentPatientLocation() PatientDaoTransformTest.testPatientsListVOToEntity() PatientDao_findPatientsTest.testFindPatients_all_parameters() PatientDaoTransformTest.testToPatientVO_all_populated() PatientDaoTransformTest.testPatientVOToEntity() PatientDao_findPatientsTest.testFindPatients_currentvisit_no_enrolment() PatientDaoTransformTest.testToPatientVO_no_currentPatientlocation_ no_currentVisit() PatientDaoTransformTest.testToPatientsListVO_all_populated()
PatientLocationDao	PatientLocationDaoTransformTest.testToPatientLocationVO() PatientLocationDaoTransformTest.testPatientLocationVOToEntity()
ProposedMedicationRecordDao	ProposedMedicationRecordDaoTransformTest.testToProposedMedication RecordDetailVO() ProposedMedicationRecordDaoTransformTest.testProposedMedication RecordVOToEntity() ProposedMedicationRecordDaoTransformTest.testToProposedMedication RecordVO() ProposedMedicationRecordDaoTransformTest.testProposedMedication RecordDetailVOToEntity()
RecordDao	RecordDao_getAllVersionsByRecordIDTest.testSuccessPath() RecordDao_findRecordsForPatientTest.testFindRecordsForPatient_ok() RecordDao_findRecordsForPatientTest.testfindRecordsForPatient_no_patientID() RecordDao_findByPatientIDTest.testFindRecordsForPatient_ok() RecordDao_findRecordsForPatientTest.testfindRecordsForPatient_ok2() RecordDao_findByPatientIDTest.testfindRecordsForPatient_no_patientID() RecordDao_findByEnrolmentIDTest.testFindByEnrolmentID_sort_order() RecordDao_findByPatientIDTest.testfindRecordsForPatient_ok2() RecordDao_findByEnrolmentIDTest.testFindByEnrolmentID_ok() RecordDao_findRecordsForEnrolmentTest.testFindRecordsForEn

	rolment_ok() RecordDaoTransformTest.testRecentActivitiesVOToEntity() RecordDao_findRecordsForEnrolmentTest.testfindRecordsForEnrolment_no_enrolmentId() RecordDao_findByEnrolmentIDTest.findByEnrolmentID_no_enrolmentId() RecordDao_getActiveByRecordIDTest.testSuccessPath() RecordDaoTransformTest.testToRecentActivitiesVO() RecordDao_findByRecordTypeTest.testSuccessPath()
RoomDao	RoomDaoTransformTest.testRoomVOToEntity_entity_exists() RoomDaoTransformTest.testToRoomVO() RoomDaoTransformTest.testRoomVOToEntity()
StopEnrolmentDao	StopEnrolmentRecordDao_findByCriteriaTest.testSuccessPath()
TaskDao	TaskDao_findByCriteriaTest.testSuccessPath() TaskDaoTransformTest.testTaskDetailVOToEntity() TaskDaoTransformTest.testToTaskDetailVO()
TherapyAdjustmentRecordDao	TherapyAdjustmentRecordDao_findByCriteriaTest.testSuccessPath() TherapyAdjustmentRecordDaoTransformTest.testToTherapyAdjustmentVO() TherapyAdjustmentRecordDaoTransformTest.testTherapyAdjustmentVOToEntity() TherapyAdjustmentRecordDaoTransformTest.testToTherapyAdjustmentDetailVO() TherapyAdjustmentRecordDaoTransformTest.testToTherapyRegimenChangesVO() TherapyAdjustmentRecordDaoTransformTest.testTherapyAdjustmentDetailVOToEntity()
TherapyMedicationDao	TherapyMedicationDaoTransformTest.testToTherapyMedicationVO() TherapyMedicationDaoTransformTest.testTherapyMedicationVOToEntity()
Visit	VisitTest.testGetEnrolmentStatus_stoped_oneEnrolment() VisitTest.testGetEnrolmentStatus_started_stopStartEnrolment() VisitTest.testGetEnrolmentStatus_unknown() VisitTest.testGetEnrolmentStatus_stoped_stopStopEnrolment() VisitTest.testGetEnrolmentStatus_started_oneEnrolment()
WardDao	WardDaoTransformTest.testWardVOToEntity() WardDaoTransformTest.testWardVOToEntity_entity_exists() WardDaoTransformTest.testToWardVO()

```
Default test
Tests run: 100, Failures: 0, Skips: 0
```

9 Appendix B

In-hospital unit tests (back end)

Service tests

Enrolment tests

Service name	Methods
RecentActivitiesService	RecentActivitiesService_loadRecentActivitiesTest.testLoadRecentActivities_per_patient() RecentActivitiesService_loadRecentActivitiesTest.testLoadRecentActivities_per_user() RecentActivitiesService_scheduleActivitiesTest.testSuccessPath() RecentActivitiesService_loadRecentActivitiesTest.testLoadRecentActivities_all()

```
Default test
Tests run: 4, Failures: 0, Skips: 0
```

Enrolment tests

Service name	Methods
EnrolmentService	EnrolmentService_updateEnrolmentTest.testStartEnrolment_NoEnrolmentExists() EnrolmentService_updateEnrolmentTest.testStartEnrolment_DataintegrityCheck() EnrolmentService_stopEnrolmentTest.testStopEnrolment_SuccessPath() EnrolmentService_updateEnrolmentTest.testUpdateEnrolment_SuccessPath() EnrolmentService_stopEnrolmentTest.testStopEnrolment_NoEnrolmentExists() EnrolmentService_startEnrolmentTest.testStartEnrolment_NoPatientFound() EnrolmentService_startEnrolmentTest.testStartEnrolment_SuccessPath() EnrolmentService_startEnrolmentTest.testStartEnrolment_No_CurrentVisit() EnrolmentService_startEnrolmentTest.testStartEnrolment_EnrolmentExists()

```
Default test
Tests run: 9, Failures: 0, Skips: 0
```

Facility tests

Service name	Methods
FacilityService	FacilityService_loadRoomsForWardTest.testCase_ward_not_found() FacilityService_loadRoomsForWardTest.testSuccessPath_one_ward_two_rooms()
LocalFacilityService	LocalFacilityService_loadRoomsForWardTest.testCase_ward_not_found() LocalFacilityService_loadRoomsForWardTest.testSuccessPath_one_ward_two_rooms()

```
Default test
Tests run: 4, Failures: 0, Skips: 0
```

Measurement tests

Service name	Methods

MeasurementService	MeasurementService_addMeasurementTest.testAddMeasurement_no_enrolment() MeasurementService_findMeasurementsTest.testSuccessPath() MeasurementService_addMeasurementTest.testAddMeasurementSuccessPath() MeasurementService_addMeasurementTest.testAddMeasurement_measurement_type_not_set() MeasurementService_updateMeasurementValueTest.testUpdateMeasurementValue_measurement_deactivated() MeasurementService_updateMeasurementRecordTest.testUpdateMeasurementRecord_one_previous_mr() MeasurementService_deactivateMeasurementTest.testDeactivateMeasurementRecord_measurement_deactivated() MeasurementService_deactivateMeasurementTest.testDeactivateMeasurementRecord_successPath() MeasurementService_loadMeasurementTypesTest.testSuccessPath() MeasurementService_addMeasurementTest.testAddMeasurement_measurement_type_deactivated() MeasurementService_updateMeasurementValueTest.testUpdateMeasurementValueSuccessPath() MeasurementService_updateMeasurementValueTest.testUpdateMeasurementValue_measurement_not_found() MeasurementService_deactivateMeasurementTest.testDeactivateMeasurementRecord_measurement_not_found()

```
Default test
Tests run: 13, Failures: 0, Skips: 0
```

Medication tests

Service name	Methods
DrugService	DrugService_loadDrugsTest.testLoadDrugsTest_SuccessPath()
LocalDrugService	LocalDrugService_loadDrugsTest.testLoadDrugsTest_SuccessPath()
MedicationService	MedicationService_addMedicationRecordTest.testAddMedicationRecord_therapy_adjustment_not_set() MedicationService_addMedicationRecordTest.testAddMedicationRecord_therapy_medication_not_set() MedicationService_addMedicationRecordTest.testAddMedicationRecordSuccessPath() MedicationService_updateMedicationRecordTest.testUpdateMedicationRecord_deactivated_mr() MedicationService_deactivateMedicationRecordTest.testDeactivateMedicationRecord_enrolment_stopped() MedicationService_deactivateMedicationRecordTest.testDeactivateMedicationRecord_mr_deactivated() MedicationService_deactivateMedicationRecordTest.testDeactivateMedicationRecord_successPath() MedicationService_updateMedicationRecordTest.testUpdateMedicationRecord_enrolment_stopped() MedicationService_updateMedicationRecordTest.testUpdateMedicationRecord_one_previous_mr() MedicationService_addMedicationRecordTest.testAddMedicationRecord_non_supported_therapy_without_measurement() MedicationService_addMedicationRecordTest.testAddMedicationRecord_stoped_Enrolment() MedicationService_findMedicationsTest.testSuccessPath() MedicationService_addMedicationRecordTest.testAddMedicationRecord_no_Enrolment()

	MedicationService_addMedicationRecordTest.testAddMedicationRecord_critical() MedicationService_updateMedicationRecordTest.testUpdateMedicationRecord_mr_not_found() MedicationService_deactivateMedicationRecordTest.testDeactivateMedicationRecord_mr_not_found()
ProposedMedicationService	ProposedMedicationService_addProposedMedicationTest.testAddProposedMedicationRecord_therapy_adjustment_not_set() ProposedMedicationService_updateProposedMedicationTest.testUpdateProposedMedicationRecord_enrolment_stopped() ProposedMedicationService_addProposedMedicationTest.testAddProposedMedicationSuccessPath() ProposedMedicationService_addProposedMedicationTest.testAddProposedMedicationRecord_therapy_medication_not_set() ProposedMedicationService_updateProposedMedicationTest.testUpdateProposedMedicationRecord_deactivated_pmr() ProposedMedicationService_deactivateProposedMedicationTest.testDeactivateProposedMedication_enrolment_stopped() ProposedMedicationService_addProposedMedicationTest.testAddProposedMedicationRecord_no_Enrolment() ProposedMedicationService_initProposedMedicationsTest.testSuccessPath() ProposedMedicationService_deactivateProposedMedicationTest.testDeactivateProposedMedication_successPath() ProposedMedicationService_deactivateProposedMedicationTest.testDeactivateProposedMedication_pmr_deactivated() ProposedMedicationService_deactivateProposedMedicationTest.testDeactivateProposedMedication_pmr_not_found() ProposedMedicationService_updateProposedMedicationTest.testUpdateProposedMedicationRecord_one_previous_pmr() ProposedMedicationService_addProposedMedicationTest.testAddProposedMedicationRecord_stoped_Enrolment()

```
Default test
Tests run: 31, Failures: 0, Skips: 0
```

Patient tests

Service name	Methods
PatientDataManagement Service	PatientDataManagementService_ admitPatientTest.testAdmitPatient_update_patient_ admission() PatientDataManagementService_ cancelAdmissionTest.testCancelAdmission_ successPath() PatientDataManagementService_ mergePatientInformationTest. testMergePatientInformation_nonSurvivingPatient() PatientDataManagementService_ updatePatientInformationTest. testUpdatePatientInformation_no_patient() PatientDataManagementService_ transferPatientTest.testTransferPatient_no_patient() PatientDataManagementService_ admitPatientTest.testAdmitPatient_ patient_visit_not_exist() PatientDataManagementService_ dischargePatientTest.testDischargePatient_no_visit() PatientDataManagementService_ dischargePatientTest.testSuccessPath()

	PatientDataManagementService_ dischargePatientTest.testDischargePatient_no_patient() PatientDataManagementService_ transferPatientTest.testTransferPatient_no_visit() PatientDataManagementService_ cancelAdmissionTest.testCancelAdmission_no_patient() PatientDataManagementService_ cancelDischargeTest.testCancelDischarge_ no_prior_locations() PatientDataManagementService_ cancelDischargeTest.testCancelDischarge_no_visit() PatientDataManagementService_ mergePatientInformationTest. testMergePatientInformation_survivingPatient() PatientDataManagementService_ cancelAdmissionTest.testCancelAdmission_no_visit() PatientDataManagementService_ mergePatientInformationTest.testSuccessPath() PatientDataManagementService_ transferPatientTest.testSuccessPath() PatientDataManagementService_ cancelDischargeTest.testSuccessPath() PatientDataManagementService_ admitPatientTest.testAdmitPatient_ new_patient_admission() PatientDataManagementService_ admitPatientTest.testAdmitPatient_ patient_visit_not_active() PatientDataManagementService_ admitPatientTest.testAdmitPatient_ patient_location_not_exist() PatientDataManagementService_ cancelAdmissionTest.testCancelAdmission_ active_enrolment() PatientDataManagementService_ cancelDischargeTest.testCancelDischarge_no_patient() PatientDataManagementService_ updatePatientInformationTest.testSuccessPath()
PatientService	PatientService_ loadPatientEnrolmentTest.testLoadPatientEnrolmentTest_ SuccessPath() PatientService_ loadPatientEnrolmentTest.testLoadPatientEnrolmentTest_ patient_not_found() PatientService_ findPatientsTest.testFindPatientsTest_SuccessPath()

```
    Default test
    Tests run: 27, Failures: 0, Skips: 0
```

Basal Bolus Therapy Regimen Handler (DSS)

Service name	Methods
BasalBolusTherapyRegimen HandlerService	BasalBolusTherapyRegimenHandlerService_getInitialDailyInsulinDose Test.testGetInitialDailyInsulinDose_therapy_adjustment_set() BasalBolusTherapyRegimenHandlerService_getCurrentDailyInsulinDo seTest.testGetCurrentDailyInsulinDose_dailyInsulinDose_not_set() BasalBolusTherapyRegimenHandlerService_getNewDailyInsulinDose

	Test.testGetNewDailyInsulinDose_morningBloodGlucoseValue_greater70_eveningBloodGlucoseValue_greater141_successPath() BasalBolusTherapyRegimenHandlerService_getNewDailyInsulinDoseTest.testGetNewDailyInsulinDose_morningBloodGlucoseValue_greater181_eveningBloodGlucoseValue_greater181_successPath() BasalBolusTherapyRegimenHandlerService_getPartialInsulinDoseRecommendationTest.testGetPartialInsulinDoseRecommendation_not_basal_bolus() BasalBolusTherapyRegimenHandlerService_getCurrentDailyInsulinDoseTest.testGetCurrentDailyInsulinDose_regimenType_not_supported() BasalBolusTherapyRegimenHandlerService_getPartialInsulinDoseRecommendationTest.testGetPartialInsulinDoseRecommendation_dailyInsulin_zero() BasalBolusTherapyRegimenHandlerService_getNewDailyInsulinDoseTest.testGetNewDailyInsulinDose_dssDailyInsulinDose_negative() BasalBolusTherapyRegimenHandlerService_getPartialInsulinDoseRecommendationTest.testGetPartialInsulinDoseRecommendation_no_measurements() BasalBolusTherapyRegimenHandlerService_getPartialInsulinDoseRecommendationTest.testGetPartialInsulinDoseRecommendation_dssDailyInsulin_successPath() BasalBolusTherapyRegimenHandlerService_getPartialInsulinDoseRecommendationTest.testGetPartialInsulinDoseRecommendation_measurementID_successPath() BasalBolusTherapyRegimenHandlerService_getPartialInsulinDoseRecommendationTest.testGetPartialInsulinDoseRecommendation_insulinResistance_null() BasalBolusTherapyRegimenHandlerService_getPartialInsulinDoseRecommendationTest.testGetPartialInsulinDoseRecommendation_insulinResistanse_Sensitive_successPath() BasalBolusTherapyRegimenHandlerService_getCurrentDailyInsulinDoseTest.testGetCurrentDailyInsulinDose_remainderDose_greaterEqual3() BasalBolusTherapyRegimenHandlerService_getPartialInsulinDoseRecommendationTest.testGetPartialInsulinDoseRecommendation_insulinResistanse_Resistant_successPath() BasalBolusTherapyRegimenHandlerService_getCurrentDailyInsulinDoseTest.testGetCurrentDailyInsulinDose_remainderDose_greaterEqual2() BasalBolusTherapyRegimenHandlerService_getNewDailyInsulinDoseTest.testGetNewDailyInsulinDose_morningBloodGlucoseValue_greater141_eveningBloodGlucoseValue_greater141_successPath() BasalBolusTherapyRegimenHandlerService_getCurrentDailyInsulinDoseTest.testGetCurrentDailyInsulinDose_remainderDose_greaterEqual1() BasalBolusTherapyRegimenHandlerService_getInitialDailyInsulinDoseTest.testGetInitialDailyInsulinDose_successPath_creatinine_greater_2() BasalBolusTherapyRegimenHandlerService_getNewDailyInsulinDoseTest.testGetNewDailyInsulinDose_morningBloodGlucoseValue_hypo_eveningBloodGlucoseValue_hypo() BasalBolusTherapyRegimenHandlerService_getNewDailyInsulinDoseTest.testGetNewDailyInsulinDose_measurements_before_therapyAdjustment() BasalBolusTherapyRegimenHandlerService_getInitialDailyInsulinDoseTest.testGetInitialDailyInsulinDose_successPath() BasalBolusTherapyRegimenHandlerService_getInitialDailyInsulinDoseTest.testGetInitialDailyInsulinDose_successPath_age_greater_70() BasalBolusTherapyRegimenHandlerService_getPartialInsulinDoseRecommendationTest.testGetPartialInsulinDoseRecommendation_bg_zer

	o() BasalBolusTherapyRegimenHandlerService_getNewDailyInsulinDose Test.testGetNewDailyInsulinDose_morningMeasurement_missing() BasalBolusTherapyRegimenHandlerService_getCurrentDailyInsulinDo seTest.testGetCurrentDailyInsulinDose_remainderDose_less0() BasalBolusTherapyRegimenHandlerService_getNewDailyInsulinDose Test.testGetNewDailyInsulinDose_eveningMeasurement_missing()

```
Default test
Tests run: 76, Failures: 0, Skips: 0
```

Task Manager

Service name	Methods
TaskManagementCore Service	TaskManagementCoreService_ resolveTaskTest.testResolveTask_ resolve_task_no_exc_if_not_found() TaskManagementCoreService_ resolveTaskTest.testResolveTask_resolve_activity_task() TaskManagementCoreService_ cancelTaskTest.testCancelTask_ cancel_reminder_task_all_patients() TaskManagementCoreService_ cancelTaskTest.testCancelTask_cancel_activity_task() TaskManagementCoreService_ generateTaskIDTest.testSuccessPath() TaskManagementCoreService_ resolveTaskTest.testResolveTask_ resolve_reminder_task_all_patients() TaskManagementCoreService_ findTasksTest.testFindTasks_ scheduledStartDateTimeFrom_To() TaskManagementCoreService_ cancelTaskTest.testCancelTask_ cancel_reminder_task_specific_patient() TaskManagementCoreService_ findTasksTest.testFindTasks_ scheduledEndDateTimeFrom_To() TaskManagementCoreService_ findTasksTest.testSuccessPath() TaskManagementCoreService_ createTaskTest.testCreateTask_createdBySystem() TaskManagementCoreService_ resolveTaskTest.testResolveTask_ resolve_reminder_task_specific_patient()
TaskManagement Service	TaskManagementService_ cancelTaskTest.testSuccessPath() TaskManagementService_ findTasksTest.testSuccessPath() TaskManagementService_ resolveTaskTest.testSuccessPath() TaskManagementService_ createTaskTest.testSuccessPath()
TaskSchedulerService	TaskSchedulerService_ scheduleTasksForEnrolmentTest. testSuccessPath_eveningTasks() TaskSchedulerService_ scheduleTasksForEnrolmentTest.

	testSuccessPath_dailyTasks() TaskSchedulerService_ scheduleTasksForEnrolmentTest.testSuccessPath_ midDayTasks() TaskSchedulerService_ scheduleTasksTest.testScheduleTasks() TaskSchedulerService_ scheduleTasksForEnrolmentTest. testSuccessPath_nightTasks() TaskSchedulerService_ removeScheduledTasksForEnrolmentTest. testRemoveScheduleTasksForEnrolment_one_active_task()

```
Default test
Tests run: 22, Failures: 0, Skips: 0
```

Therapy tests

Service name	Methods
TherapyAdjustment Service	TherapyAdjustmentService_ setTherapyAdjustmentTest.testSetTherapyAdjustment_ one_prior_record_exists() TherapyAdjustmentService_ getRegimenChangesTest.testGetRegimenChanges_ To_Date_Not_Set() TherapyAdjustmentService_ loadTherapyAdjustmentTest.testLoadTherapyAdjustment_ one_medication_one_proposed_medication() TherapyAdjustmentService_ getRegimenChangesTest.testGetRegimenChanges_ From_and_To_Date_Not_Set() TherapyAdjustmentService_ loadTherapyAdjustmentTest.testLoadTherapyAdjustment_ no_medication_no_proposed_medication() TherapyAdjustmentService_ setTherapyAdjustmentTest.testSetTherapyAdjustment_ no_Enrolment() TherapyAdjustmentService_ getRegimenChangesTest.testGetRegimenChanges_ SuccessPath() TherapyAdjustmentService_ getRegimenChangesTest.testGetRegimenChanges_ No_TherapyAdjustment_Exists() TherapyAdjustmentService_ setTherapyAdjustmentTest.testSetTherapyAdjustment_ no_prior_record_exists()

```
Default test
Tests run: 9, Failures: 0, Skips: 0
```

User tests

Service name	Methods
UserService	UserService_ findUsersTest.testFindUsersTest_SuccessPath()

```
Default test
```

```
Tests run: 1, Failures: 0, Skips: 0
```

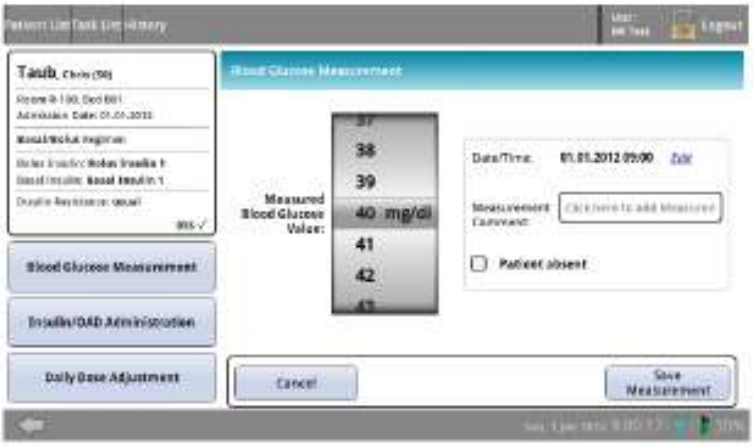
10 Appendix C

In-hospital unit tests (front end)

GluCoManSys Android Application - Test Report

executed at: Jan 1, 2012 9:11:20 AM

Test Case

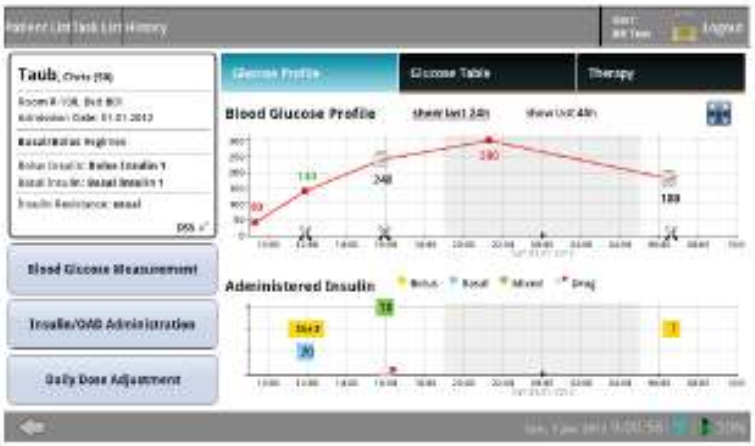
Test ID	testCheckBGMeasurement
Test Group	GMMainScreenTests
Test Result	PASSED
Time needed	-335.813 seconds
Screenshot of result	

Test Case

Test ID	testCheckBasalBolusTherapySettings
Test Group	GMMainScreenTests
Test Result	PASSED
Time needed	8.553 seconds
Screenshot of result	

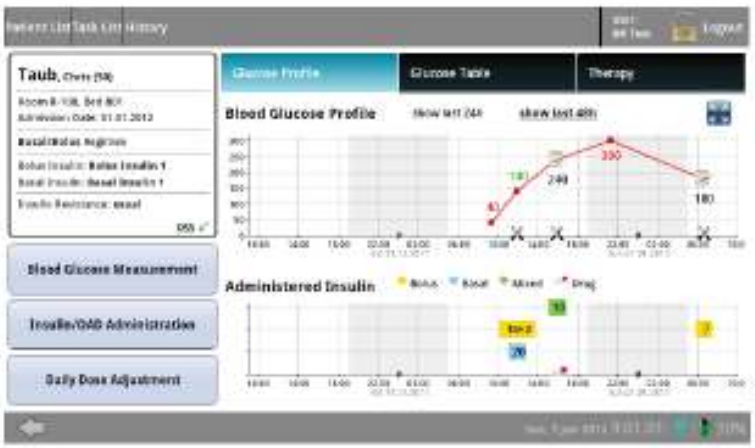
Test ID	testCheckDailyDoseAdjustment
Test Group	GMMainScreenTests
Test Result	PASSED
Time needed	26.867 seconds
Screenshot of result	

Test Case

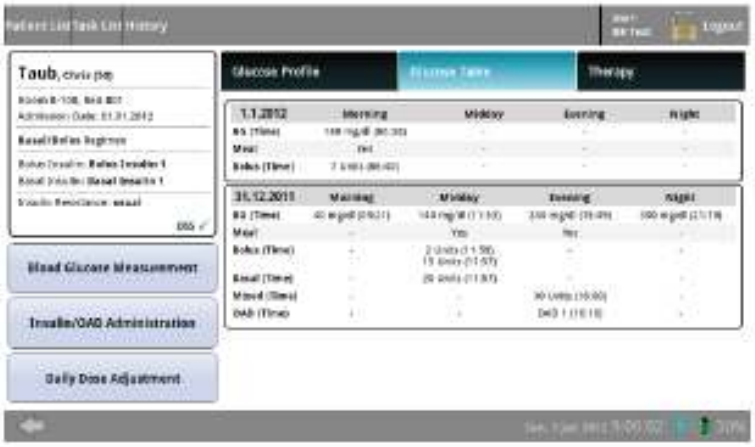
Test ID	testCheckGMMainScreentActivityActionBar
Test Group	GMMainScreenTests
Test Result	PASSED
Time needed	1.944 seconds
Screenshot of result	

Test Case

Test ID	testCheckGlucoseProfile
Test Group	GMMainScreenTests
Test Result	PASSED
Time needed	5.810 seconds

Screenshot of result	

Test Case

Test ID	**testCheckGlucoseTable**
Test Group	**GMMainScreenTests**
Test Result	PASSED
Time needed	-60.104 seconds
Screenshot of result	

Test Case

Test ID	**testCheckInsulinAdministrationInBasalBolus**
Test Group	**GMMainScreenTests**
Test Result	PASSED
Time needed	55.225 seconds

Screenshot of result	

Test Case

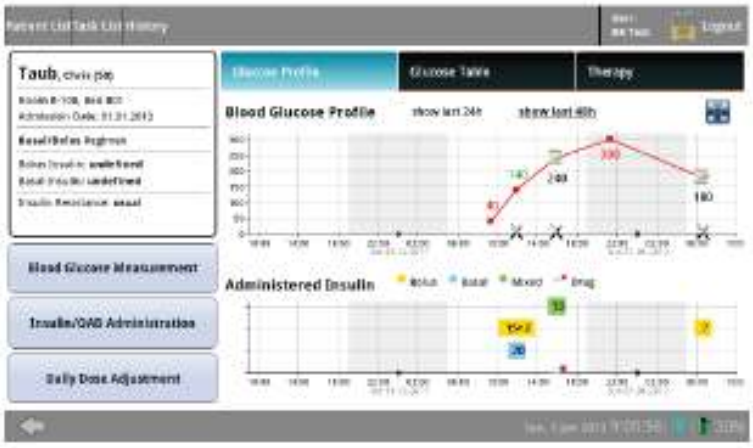
Test ID	testCheckInsulinAdministrationInNonSupported
Test Group	GMMainScreenTests
Test Result	PASSED
Time needed	-15.425 seconds
Screenshot of result	

Test Case

Test ID	testCheckNonSupportedTherapySettings
Test Group	GMMainScreenTests
Test Result	PASSED
Time needed	8.399 seconds

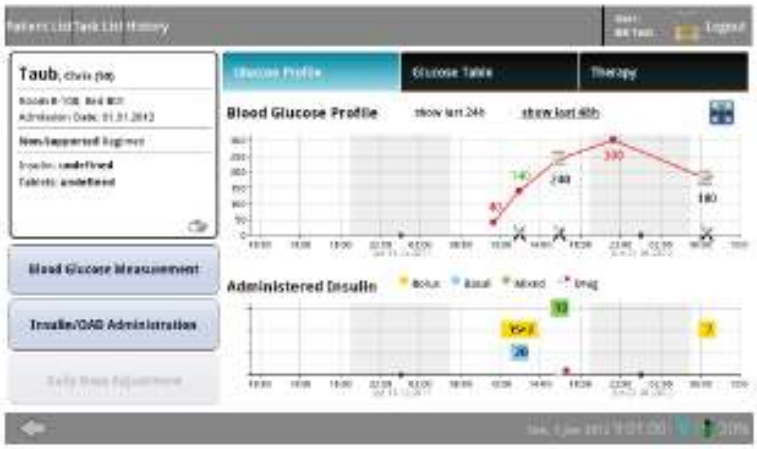
Screenshot of result	

Test Case

Test ID	**testCheckPatientDetailsInBasalBolusRegimen**
Test Group	**GMMainScreenTests**
Test Result	PASSED
Time needed	4.607 seconds
Screenshot of result	

Test Case

Test ID	**testCheckPatientDetailsInNonSupportedRegimen**
Test Group	**GMMainScreenTests**
Test Result	PASSED
Time needed	3.530 seconds

Screenshot of result	

Test Case

Test ID	**testCheckPermissionsInBasalBolusRegimen**
Test Group	**GMMainScreenTests**
Test Result	PASSED
Time needed	-9.685 seconds
Screenshot of result	

Test Case

Test ID	**testCheckPermissionsInNonSupportedRegimen**
Test Group	**GMMainScreenTests**
Test Result	PASSED
Time needed	48.538 seconds

Screenshot of result	

Test Case

Test ID	**testCheckActivityActionBar**
Test Group	**PatientManagementTests**
Test Result	PASSED
Time needed	2.157 seconds
Screenshot of result	

Test Case

Test ID	**testCheckPatientListDefaultPresentation**
Test Group	**PatientManagementTests**
Test Result	PASSED
Time needed	3.658 seconds

Screenshot of result	

Test Case

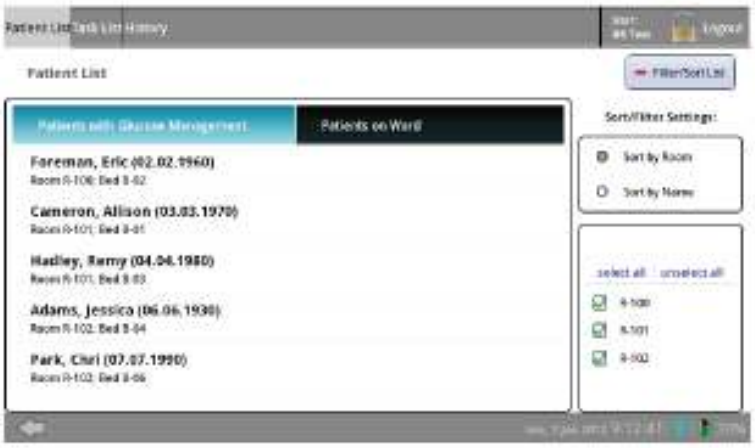
Test ID	**testCheckPatientListWithoutPatients**
Test Group	**PatientManagementTests**
Test Result	PASSED
Time needed	1.617 seconds
Screenshot of result	

Test Case

Test ID	**testPatientListOnClickListener**
Test Group	**PatientManagementTests**
Test Result	PASSED
Time needed	56.443 seconds

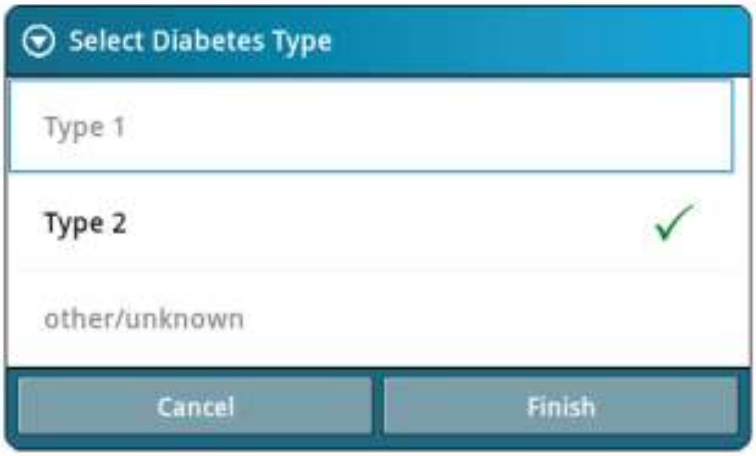
Screenshot of result	

Test Case

Test ID	**testSortAndFilteringPatientLists**
Test Group	**PatientManagementTests**
Test Result	PASSED
Time needed	17.290 seconds
Screenshot of result	

Test Case

Test ID	**testCheckPatientEnrolmentActivityActionBar**
Test Group	**PatientEnrolmentTests**
Test Result	PASSED
Time needed	1.456 seconds

Screenshot of result	

Test Case

Test ID	testCheckPatientEnrolmentActivityInUpdateMode
Test Group	PatientEnrolmentTests
Test Result	PASSED
Time needed	12.531 seconds
Screenshot of result	

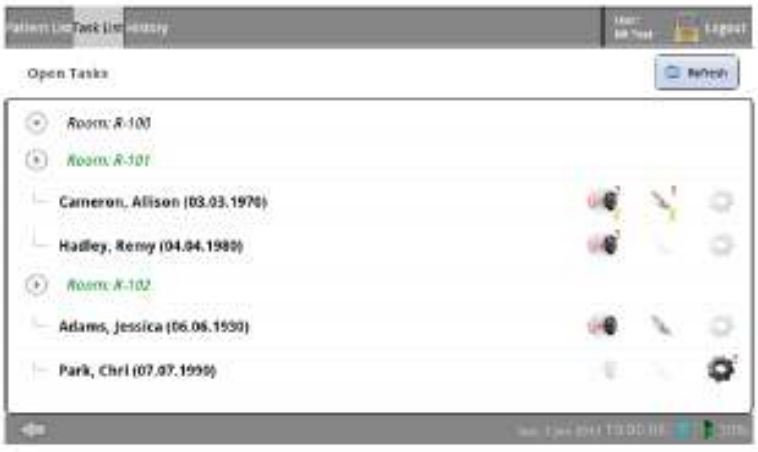
Test Case

Test ID	testCheckPatientEnrolmentActivityWithAlreadyEnrolledPatient
Test Group	PatientEnrolmentTests
Test Result	PASSED
Time needed	16.464 seconds
Screenshot of result	

Test Case

Test ID	**testCheckPatientEnrolmentActivityWithNeverEnrolledPatient**
Test Group	**PatientEnrolmentTests**
Test Result	PASSED
Time needed	12.327 seconds
Screenshot of result	

Test Case

Test ID	**testCheckCorrectPresentationOfTasks**
Test Group	**TaskManagementTests**
Test Result	PASSED
Time needed	2802.475 seconds
Screenshot of result	

Test Case

Test ID	**testCheckTaskListWithoutPatients**
Test Group	**TaskManagementTests**
Test Result	PASSED

Time needed	1.498 seconds
Screenshot of result	

Test Case

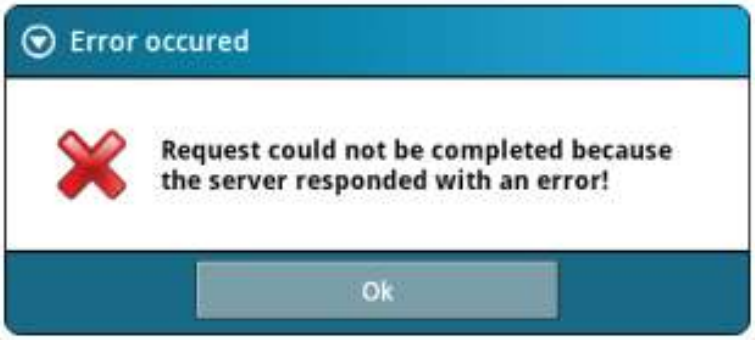
Test ID	**testCheckTaskListWithoutTasks**
Test Group	**TaskManagementTests**
Test Result	PASSED
Time needed	17.642 seconds
Screenshot of result	

Test Case

Test ID	**testCheckTaskManagementActivityActionBar**
Test Group	**TaskManagementTests**
Test Result	PASSED
Time needed	1.835 seconds

Screenshot of result	

Test Case

Test ID	**testTaskListOnClickListener**
Test Group	**TaskManagementTests**
Test Result	PASSED
Time needed	31.720 seconds
Screenshot of result	

Test Case

Test ID	**testCheckScreenOrientation**
Test Group	**StartScreenTests**
Test Result	PASSED
Time needed	4.974 seconds

Screenshot of result	

Test Case

Test ID	testCheckStartScreenActivityActionBar
Test Group	StartScreenTests
Test Result	PASSED
Time needed	1.947 seconds

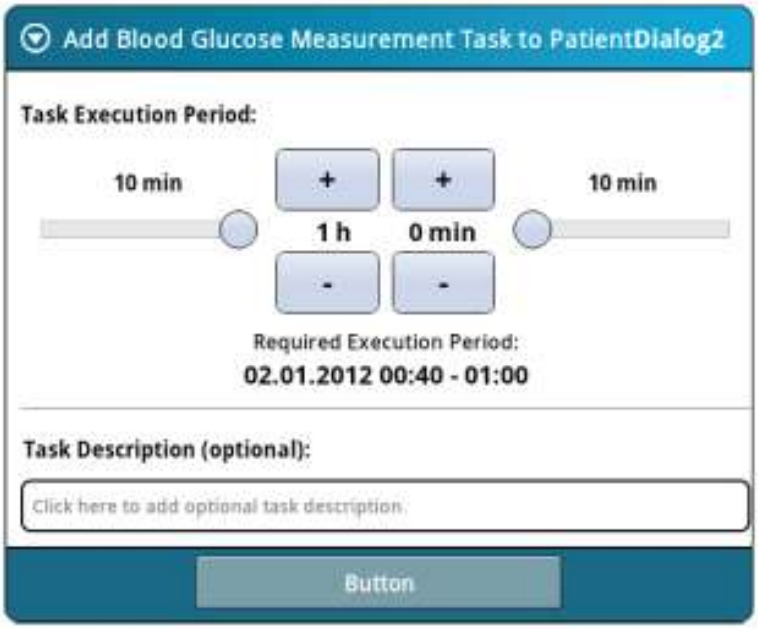
Screenshot of result	

Test Case

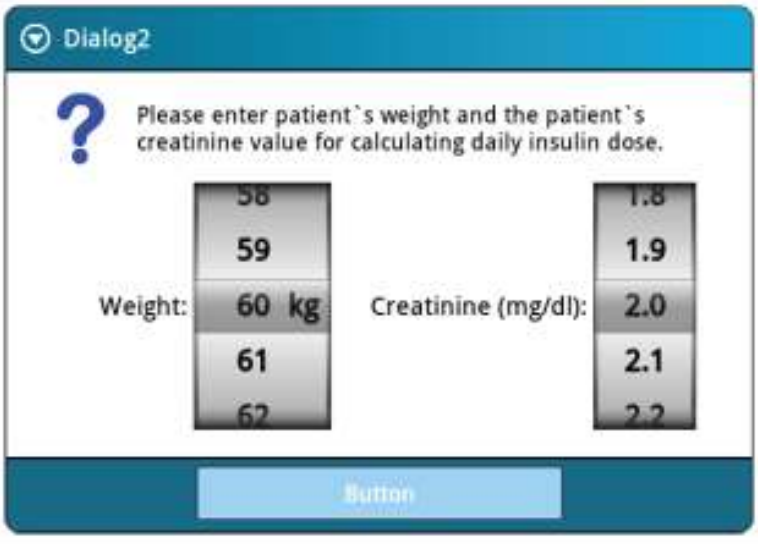
Test ID	testYesNoDialogFunctionality
Test Group	YesNoDialogTests
Test Result	PASSED
Time needed	10.732 seconds
Screenshot of result	

Test Case

Test ID	testCheckAddTaskDialogFunctionality
Test Group	AddTaskDialogTests
Test Result	PASSED
Time needed	49723.222 seconds

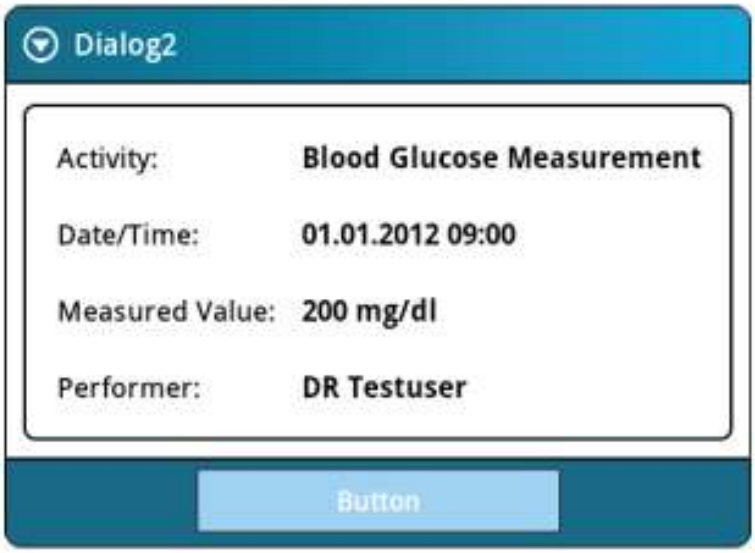
Screenshot of result	

Test Case

Test ID	testCalcDailyInsulinDoseDialogFunctionality
Test Group	CalcDailyInsulinDoseDialogTests
Test Result	PASSED
Time needed	8.092 seconds
Screenshot of result	

Test Case

Test ID	testChartPointInfoDialogFunctionalityWithBG
Test Group	ChartPointInfoDialogTests
Test Result	PASSED

Time needed	8.618 seconds
Screenshot of result	

Test Case

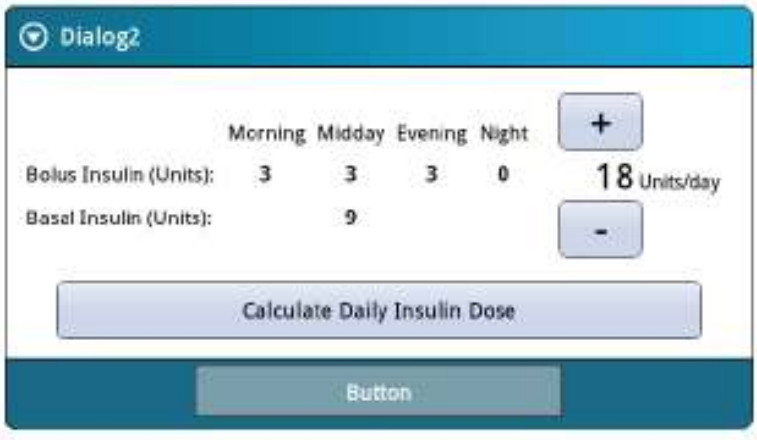
Test ID	testChartPointInfoDialogFunctionalityWithBolusInsulin
Test Group	ChartPointInfoDialogTests
Test Result	PASSED
Time needed	16.060 seconds
Screenshot of result	

Test Case

Test ID	testChartPointInfoDialogFunctionalityWithNutrition
Test Group	ChartPointInfoDialogTests

Test Result	PASSED
Time needed	4.863 seconds
Screenshot of result	

Test Case

Test ID	testDailyInsulinDoseDialogFunctionality
Test Group	DailyInsulinDoseDialogTests
Test Result	PASSED
Time needed	12.488 seconds
Screenshot of result	

Test Case

Test ID	testDateTimeDialogFunctionality

Test Group	**DateTimeDialogTests**
Test Result	PASSED
Time needed	-85825.809 seconds
Screenshot of result	

Test Case

Test ID	**testListOperatorDialogFunctionality**
Test Group	**ListOperatorDialogTests**
Test Result	PASSED
Time needed	6.495 seconds

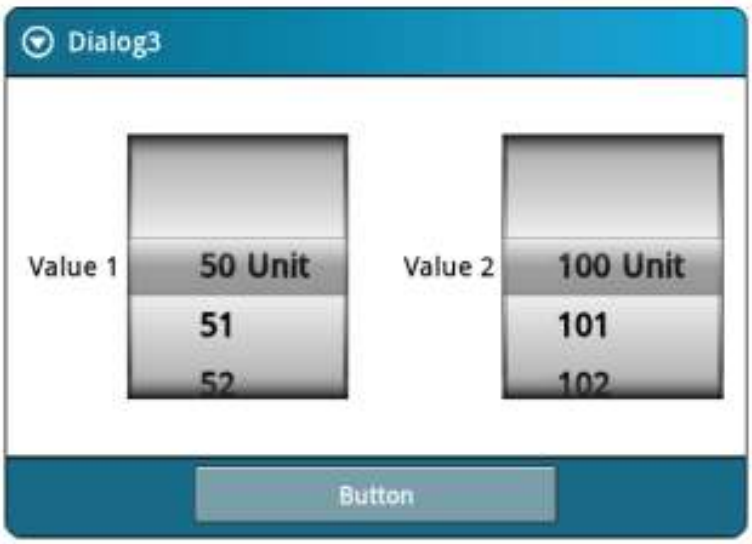
Screenshot of result	

Test Case

Test ID	testListOperatorDialogFunctionality
Test Group	ListSelectorDialogTests
Test Result	PASSED
Time needed	10.338 seconds
Screenshot of result	

Test Case

Test ID	testLowerUpperBorderDialogFunctionality
Test Group	LowerUpperBorderDialogTests
Test Result	PASSED
Time needed	9.805 seconds

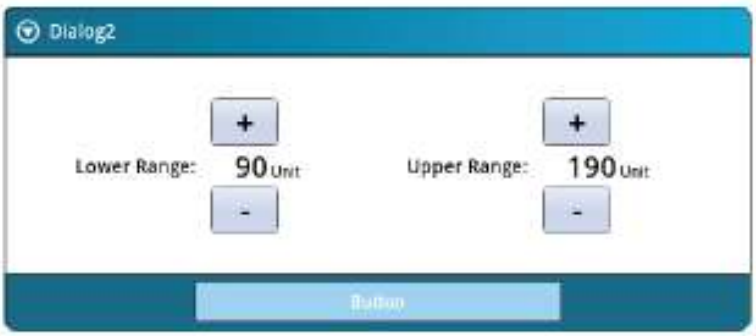
Screenshot of result	

Test Case

Test ID	testMessageDialogFunctionality
Test Group	MessageDialogTests
Test Result	PASSED
Time needed	6.726 seconds
Screenshot of result	

Test Case

Test ID	testRangeDialogFunctionality
Test Group	RangeDialogTests
Test Result	PASSED
Time needed	16.884 seconds

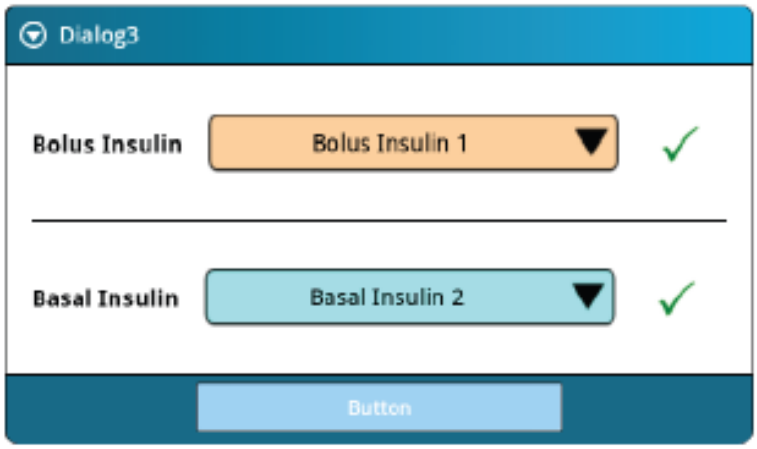
Screenshot of result	

Test Case

Test ID	testRootDialogWithFunctionality
Test Group	RootDialogTests
Test Result	PASSED
Time needed	7.401 seconds
Screenshot of result	

Test Case

Test ID	testSelectBasalBolusInsulinDialogFunctionality
Test Group	SelectBasalBolusInsulinDialogTests
Test Result	PASSED
Time needed	24.974 seconds

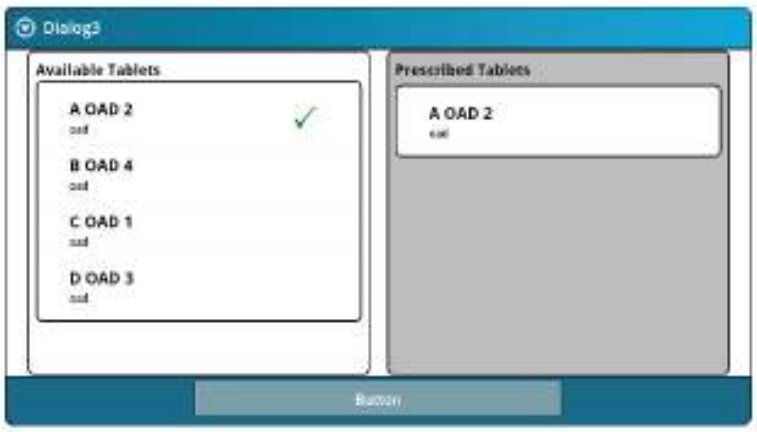
Screenshot of result	

Test Case

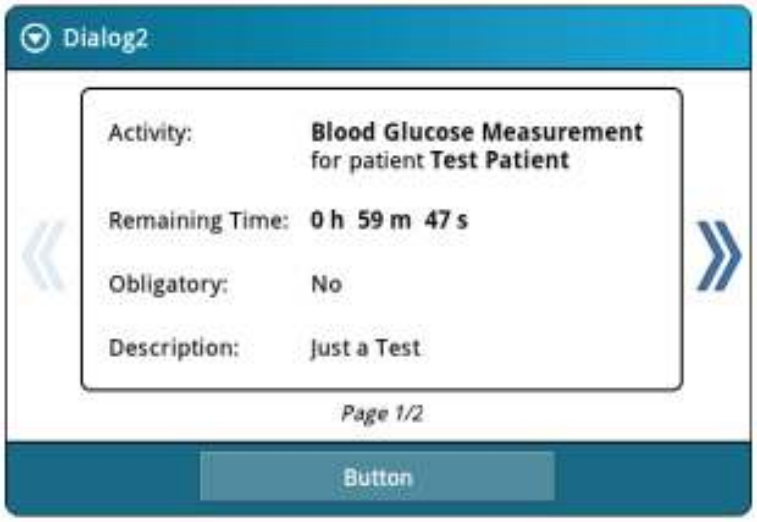
Test ID	**testSelectFreeInsulinDialogFunctionality**
Test Group	**SelectFreeInsulinDialogTests**
Test Result	PASSED
Time needed	18.620 seconds
Screenshot of result	

Test Case

Test ID	**testSelectTabletsDialogFunctionality**
Test Group	**SelectTabletsDialogTests**
Test Result	PASSED
Time needed	16.215 seconds

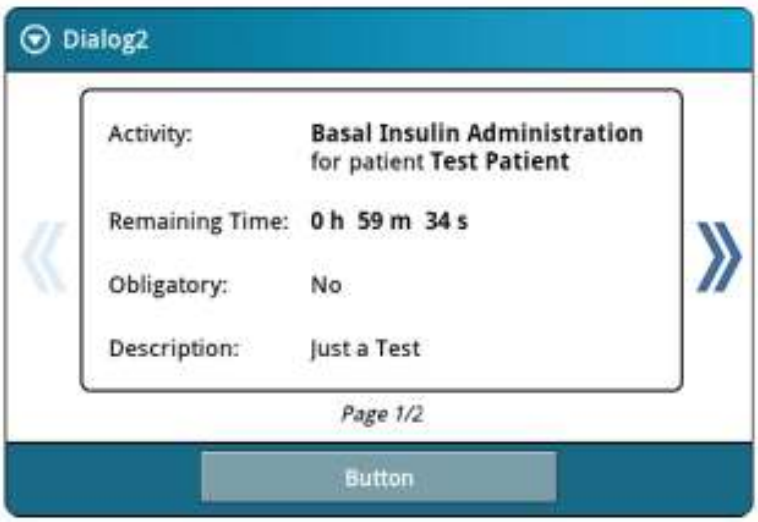
Screenshot of result	

Test Case

Test ID	**testTaskDetailsDialogFunctionalityWithBG**
Test Group	**TaskDetailsDialogTests**
Test Result	PASSED
Time needed	35862.331 seconds
Screenshot of result	

Test Case

Test ID	**testTaskDetailsDialogFunctionalityWithMedication**
Test Group	**TaskDetailsDialogTests**
Test Result	PASSED
Time needed	13.216 seconds

Screenshot of result	

Test Case

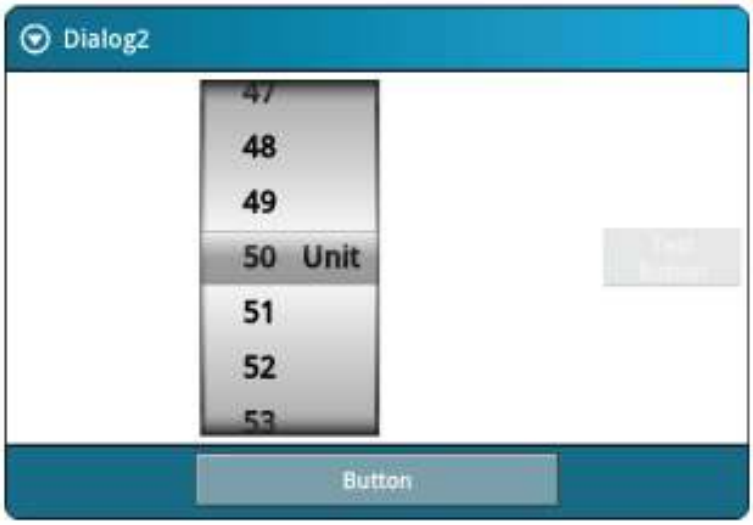
Test ID	testTaskDetailsDialogFunctionalityWithTA
Test Group	TaskDetailsDialogTests
Test Result	PASSED
Time needed	4.703 seconds
Screenshot of result	

Test Case

Test ID	testTextInputDialogFunctionality
Test Group	TextInputDialogTests
Test Result	PASSED

Time needed	14.744 seconds
Screenshot of result	

Test Case

Test ID	**testWheelPickerDialogFunctionality**
Test Group	**WheelPickerDialogTests**
Test Result	PASSED
Time needed	6.765 seconds
Screenshot of result	

11 Appendix D

In-hospital prototype – details of system tests

The table below represents the level of the satisfaction for the 2nd year in-hospital prototype after performing system testing.

11.1 Functional requirements for the In-Hospital application

Requirement Key	Summary	Release 1	Release 2	Status
		1st year in-hospital prototype (release-1.0)	2nd year in-hospital prototype (release-1.2.1)	
REACTION-466	(Web) Service to present decision support for glucose control to clinicians	No	Yes	Fully Satisfied
REACTION-465	Clinical evaluation report	No	Yes	Fully Satisfied
REACTION-463	Context management for clinical (lab) values.	No	Yes	Fully Satisfied
REACTION-462	Interface for user inputs from portable computer in order to store data in In-hospital data storage	No	Yes	Fully Satisfied
REACTION-459	Ontologies and data management designed for the storage and multi-user availability of all relevant information, actions, treatments, events	No	No	Not Satisfied
REACTION-456	Nutrition information has to be stored in the data management	No	Yes	Partially Satisfied
REACTION-446	Clinical data to be stored in the Inpatient environment	Yes	Yes	Fully Satisfied
REACTION-445	Registration of specific interfering drugs (including their dosage)	No	No	Not Satisfied
REACTION-441	Basic workflow in In-hospital environment	Yes	Yes	Fully Satisfied
REACTION-434	Interface to Lab Information System (LIS) for glucose data import	No	No	Partially Satisfied
REACTION-432	Special examinations/treatments to be registered in fever chart	No	Yes	Partially Satisfied
REACTION-428	Drug administration data (OAD and/or insulin)	No	Yes	Fully Satisfied
REACTION-402	Measurements of blood glucose and insulin injections in In-hospital environment	Yes	Yes	Fully Satisfied
REACTION-379	Interface for transmission of glucose values from POCT system to In-hospital prototype	No	No	Not Satisfied
REACTION-377	Electronic fever/sugar chart	No	No	Not Satisfied

REACTION-375	Therapy scheme in In-hospital environment	No	Yes	Fully Satisfied
REACTION-369	Storage of hyperglycaemic or hypoglycaemic episodes	No	No	Not Satisfied
REACTION-363	Interface to Hospital Information System for clinical data import/export	No	Yes	Partially Satisfied
REACTION-362	Interface to patient demographic register	No	No	Partially Satisfied
REACTION-285	User interface for the clinical data stored in the In-hospital environment	Yes	Yes	Fully Satisfied
REACTION-284	Clinical data to be stored in the In-hospital environment	Yes	Yes	Fully Satisfied
REACTION-263	Improve documentation quality and streamlined access to information	No	No	Not Satisfied
REACTION-260	Archive system: data from former admissions of the same patient can be easily retrieved and used for decision making	Yes	Yes	Fully Satisfied
REACTION-259	Automated patient identification	No	No	Not Satisfied
REACTION-258	Automated transfer of patient related data from the hospital information system	No	Yes	Partially Satisfied
REACTION-257	Automated transfer of measured and relevant data to the patient's record	No	No	Partially Satisfied
REACTION-255	Management of missing data	Yes	Yes	Fully Satisfied
REACTION-254	Data to be stored in the Inpatient environment	Yes	Yes	Fully Satisfied
REACTION-252	When some measurements are missing the system shall remind it through an active alarm reminder	No	Yes	Fully Satisfied
REACTION-251	Creation of electronic decision support rules shall be supported	No	Yes	Fully Satisfied
REACTION-250	Different contextualization of the patient clinical information	Yes	Yes	Fully Satisfied
REACTION-248	Ontologies and data management designed for the storage and multi-user availability of all relevant information, actions, treatments, events	Yes	Yes	Fully Satisfied
REACTION-247	Mobile access point in wards of In-hospital environment	Yes	Yes	Fully Satisfied
REACTION-246	Multi-user availability and display of the fever chart	Yes	Yes	Partially Satisfied
REACTION-245	Fever and infections shall be registered in the fever chart and have an impact in the insulin dosage calculation	No	No	Not Satisfied

REACTION-244	The data management and the user interface shall allow the insertion of specific interfering drugs (including their dosage). The dosage of insulin shall vary with these drugs.	No	No	Not Satisfied
REACTION-243	Nutrition has to be taken into account in the calculation of the drug dosage	No	Yes	Partially Satisfied
REACTION-241	Management of hypoglycaemic episodes in In-hospital environment	No	No	Not Satisfied
REACTION-240	Intravenous insulin	No	No	Not Satisfied
REACTION-238	Update and entering of drug administration (OAD and/or insulin) data	No	Yes	Fully Satisfied
REACTION-237	Annotation of blood glucose values, especially in In-hospital environment	Yes	Yes	Fully Satisfied
REACTION-236	Blood glucose measurements in In-hospital environment	No	No	Not Satisfied
REACTION-235	Therapy scheme in In-hospital environment registered immediately after the patient enrolment	Yes	Yes	Fully Satisfied
REACTION-234	Determination of health status in In-hospital environment	Yes	Yes	Fully Satisfied
REACTION-233	Insulin sensitivity and insulin resistance	No	Yes	Fully Satisfied
REACTION-231	End of process for the diabetic patient in the In-hospital environment	Yes	Yes	Fully Satisfied
REACTION-230	Therapy adjustment in In-hospital environment	No	Yes	Fully Satisfied
REACTION-229	Decision on therapy in In-hospital environment	No	Yes	Fully Satisfied
REACTION-228	Blood glucose measurements have to be contextualized (e.g. before/after meal)	Yes	Yes	Fully Satisfied
REACTION-227	Initialization of the fever/sugar chart	No	No	Not Satisfied
REACTION-226	Electronic fever/sugar chart should be modelled in the data management system	No	No	Not Satisfied
REACTION-225	PoC device for blood glucose measurement will be used in the first-year prototype	No	No	Not Satisfied
REACTION-224	Basic workflow is repeated 4 times a day in In-hospital environment	No	Yes	Fully Satisfied
REACTION-223	Basic workflow for insulin treatment in In-hospital environment	No	Yes	Fully Satisfied
REACTION-222	Insulin evaluation in Inpatient environment	No	Yes	Fully Satisfied

REACTION-221	Parameters monitored in Inpatient environment: blood glucose, glycated hemoglobine, nutritional intake and insulin sensitivity (evaluated but not measurable)	Yes	Yes	Fully Satisfied
REACTION-220	Healthcare professionals perform the safe glycaemic control in In-hospital environment (not self-management)	No	Yes	Fully Satisfied
REACTION-219	Safe Glycaemic Control (SGC)	No	Yes	Fully Satisfied
REACTION-175	Automated identification of users (caregivers) working with REACTION front-end in the hospital	No	No	Partially Satisfied
REACTION-174	The system must provide interfaces to HIS and implement data management and data structures for In-hospital scenario	No	No	Partially Satisfied
REACTION-173	Platform should allow ubiquitous access to end-users and sharing of information among caregivers (multiuser access to relevant data)	No	No	Not Satisfied
REACTION-172	The system should automatically transfer measurements from the POCT devices into the platform within a few seconds	No	No	Not Satisfied
REACTION-171	Data input application for In-hospital glucose control	Yes	Yes	Fully Satisfied
REACTION-170	Selection of a mobile device for In-hospital glucose control based on given requirements	Yes	Yes	Fully Satisfied
REACTION-169	Display and input of data should be possible at different locations simultaneously (centrally managed data repositories)	Yes	Yes	Fully Satisfied
REACTION-166	Archive system: data from former admissions of the same patient can be used for decision making	Yes	Yes	Fully Satisfied
REACTION-163	Archive system: data from former admissions of the same patient can be easily retrieved and used for decision making	Yes	Yes	Fully Satisfied
REACTION-161	The system should remind caregivers to perform measurements.	Yes	Yes	Fully Satisfied
REACTION-156	The system should provide a regular backup of data	No	No	Not Satisfied

REACTION-155	The System should keep an electronic paperless data record of the data relevant for Glucose Management	Yes	Yes	Fully Satisfied
REACTION-97	Quality analysis for ward personnel	No	No	Not Satisfied
REACTION-96	Visualization individual patient data to support glucose control (decision support)	Yes	Yes	Fully Satisfied
REACTION-83	Interface to clinical data from "near" real-time observations for decision support	Yes	Yes	Fully Satisfied
REACTION-72	Provide decision support for insulin dosing for clinicians (in-hospital)	No	Yes	Fully Satisfied

11.2 Non-functional requirements relating to the In-hospital application

Requirement Key	Summary	Release 1	Release 2	Status
		1st year in-hospital prototype (release-1.0)	2nd year in-hospital prototype (release1.2.1)	
REACTION-475	Log and log-in system	No	Yes	Partially Satisfied
REACTION-437	Each role MUST be assigned to a set of permissible actions.	No	Yes	Partially Satisfied
REACTION-429	Before transmitting any personal data, the patient's consent MUST be given. If no consent was given yet, the data MUST NOT be sent.	No	No	Not Satisfied
REACTION-415	Each person MAY only perform actions permitted by her role.	No	Yes	Partially Satisfied
REACTION-414	Communication between the Reaction Hosting Client and the Reaction Device Hosting Server MUST be authentic (entity authentication), with integrity, and confidential.	No	Yes	Partially Satisfied
REACTION-412	It MUST be possible to revoke a consent - data already stored MUST NOT be processed any further.	No	No	Not Satisfied
REACTION-407	If data was not transmitted for a lack of consent, the patient or her doctor (in case of a client without display and input capabilities) MUST be notified, e.g., through some pop-up or a notice in some message field.	No	No	Not Satisfied
REACTION-403	Each entity in the Reaction platform MUST be representable by a digital identity.	No	Yes	Partially Satisfied
REACTION-398	If a consent was given, the patient's involvement in the decision MUST be verifiable by the Reaction Hosting Client, especially if the consent was given remotely, e.g., at the doctor's surgery.	No	No	Not Satisfied
REACTION-385	Digital identities for the Reaction platform MUST only be issued or revoked by trusted (third) parties, e.g., a certification authority (CA).	No	No	Not Satisfied

REACTION-373	Data MUST NOT be processed at the Reaction Device Hosting Server if no consent is available and verifiable.	No	No	Not Satisfied
REACTION-370	Consent MUST NOT considered valid if the patient was not involved in the decision.	No	No	Not Satisfied
REACTION-359	Maximum delay to transfer blood glucose value from POCT to In-hospital prototype	No	No	Not Satisfied
REACTION-356	Manual data insertion	Yes	Yes	Fully Satisfied
REACTION-343	Every person represented in the Reaction platform MUST be assigned to one or more roles.	No	Yes	Fully Satisfied
REACTION-341	Roles MUST be defined for stakeholders of the Reaction platform, e.g., doctor, nurse, patient, informal carer, administrative personnel etc.	No	Yes	Fully Satisfied
REACTION-321	Risk analysis	No	Yes	Partially Satisfied
REACTION-162	Documentation of user interfaces	No	No	Not Satisfied
REACTION-151	The user must be able to correct, rectify, block or erase personal data that has been disclosed	No	No	Partially Satisfied
REACTION-122	The portable touch device must have a satisfactory operational time. The battery must be able to support the device for at least half a working day. If the device supports exchangeable battery that would be an advantage.	Yes	Yes	Fully Satisfied
REACTION-121	*The portable touch device must have at least the following connectivity options: WiFi (802.11g or 802.11n), Bluetooth, usb *Also must have built in at least the following sensors: GPS, accelerometer *If the device is a mobile phone it must support 3G	Yes	Yes	Fully Satisfied
REACTION-120	If the touch/tablet/phone device is not able to send the data to the platform (lack of connectivity), it should store them locally and then send them when the connectivity is re-established. (The device must have a decent amount of internal storage, or ac	No	No	Not Satisfied
REACTION-115	Transparency: Security configuration should be hidden from the user as far as possible	Yes	Yes	Fully Satisfied

REACTION-113	Only one or max two categories of different mobile operating systems will be considered for the portable devices	Yes	Yes	Fully Satisfied
REACTION-111	The portable touch device must have a display of sufficient screen estate & resolution (more than a 3.5' display, more than 320px*480px resolution). If the device is not a stylus operating device then the display must be of capacitive technology & with su	Yes	Yes	Fully Satisfied
REACTION-106	The touch/tablet/phone device must support notification messages.	Yes	Yes	Fully Satisfied
REACTION-105	The touch/tablet/phone device must allow the execution of processes in the background.	Yes	Yes	Fully Satisfied
REACTION-100	Access control: Access to sensitive information should only by given to authorised personnel	No	Yes	Partially Satisfied
REACTION-99	Authorisation: Stakeholders must be authorised before they are allowed to perform relevant actions	No	Yes	Partially Satisfied
REACTION-80	Only one or max two categories of different mobile operating systems will be considered for the portable devices	Yes	Yes	Fully Satisfied
REACTION-76	Portability	No	Yes	Fully Satisfied
REACTION-67	Component Repository	Yes	Yes	Fully Satisfied
REACTION-56	The portable touch device must have a satisfactory operational time. The battery must be able to support the device for at least half a day. If the device supports exchangeable battery that would be an advantage.	Yes	Yes	Fully Satisfied
REACTION-55	The portable touch device must have a display of sufficient screen size & resolution (more than a 3,5" display, more than 320px*480px). If not a stylus operating device then the display must be of capacitive technology & with support for multitouch.	Yes	Yes	Fully Satisfied
REACTION-53	*The portable touch device must have at least the following connectivity options: WiFi (802.11g or 802.11n), Bluetooth, USB; *Also it must have built in at least the following sensors: GPS,	Yes	Yes	Fully Satisfied

	accelerometer; *If mobile phone it must support 3G networks.			
REACTION-52	If the portable touch device is not capable to connect wirelessly and send the data, then it should be able to connect via USB to a host gateway with connectivity to the Internet & upload the measurement file to the platform.	No	No	Not Satisfied
REACTION-50	The touch/tablet/phone device must support notification messages	No	No	Not Satisfied
REACTION-49	The touch/tablet/phone device must allow the execution of processes in the background	Yes	Yes	Fully Satisfied
REACTION-48	Support for multilingual user interface	Yes	Yes	Fully Satisfied
REACTION-46	Error messages must be understandable and helpful	No	Yes	Fully Satisfied
REACTION-44	Protection against unintended user actions	Yes	Yes	Fully Satisfied
REACTION-43	Protection against data loss System must protect against: *Loss or replication of data transferred between two systems; *Concurrency problems; *Disk crash; *Protection against physical means.	No	No	Not Satisfied
REACTION-41	The tools developed by the consortium must be properly documented in such a way that the end user can understand them and use them for the intended purpose.	Yes	Yes	Fully Satisfied

11.3 Constraint requirements focused on in-hospital application

Requirement Key	Summary	Release 1	Release 2	Status
		1st year in-hospital prototype	2nd year in-hospital prototype	
REACTION-391	Data fields for the In-hospital glucose control prototype (eDSS).	No	Yes	Fully Satisfied