E-Health Service
Final Project Report

Version 1.0

Doc. No.:

Revision History

Date	Version	Description	Author
2013-01-11	0.1	Initial Draft	Stefania Pezzetti
2013-01-18	0.2	Chapters 2,3,4,5,6	Stefania Pezzetti
2013-01-20	1.0	Finalized document	Stefania Pezzetti

Table of Contents

1. Introduction

1.1 Purpose of this document

The purpose of this document is to display an overview of the results and metrics of the E-Health Service project as performed during the Distributed Software Development course 2012/2013. This course is joint course between Politecnico di Milano (POLIMI) in Italy, University of Zagreb (FER) in Croatia and Mälardalen University (MDH) in Sweden. The E-Health team member are from POLIMI and from FER. This document is written at the end of the course, when the project is concluded.

1.2 Intended Audience

This document informs the project stakeholders about the project performance and achievements.

1.3 Scope

This document covers only the result of the project. It will not cover any assumption made in the beginning of the project and will only cover differences between the assumptions and results

1.4 References

This is the link at our project in DSD site:
http://www.fer.unizg.hr/rasip/dsd/projects/e-health_service/documents

Documentation written for the project can be found in the folder “Documentation”.
Technical documentation as User Manual and Installation Manuals can be consulted in the folder “Final Documentation”.
The source code can be found in the folder “Final Product”.

The link to the E-Health site, where the web application can be used and the Android application downloaded is the following:
http://ehealthservicetest5.cloudapp.net

2. Background and Objectives

E-Health system supports doctors in everyday activities, such as the retrieval of mixed media information of their patients that combines text with or without files referring to their personal data, their medical histories and patient-related diseases.

The patients are allowed to retrieve information about the doctor expertise and they update some vital parameters (e.g. heart rate) thus their health is continuously monitored. On the basis of such data, doctors may send an alarm in case of warning conditions.

The system manages alarm notifications sent to first aid both by doctors and directly by patients.

The final product will be delivered as web application and as mobile application. Milestones and deliverables are reported below:

Every Monday (from 2012-10-29 to 2013-01-21)	*Summary Week Report*
2012-10-30	*Project Plan Presentation*
2012-11-02	*Project Plan Document*
2012-11-02	*Requirements Definition Document*
2012-11-06	*Requirements Definition and System Architecture Presentation*
2012-11-09	*Design Description Document*
2012-11-27	*Alpha Prototype Presentation*
2012-12-18	*Beta Prototype Presentation*
2012-12-31	*Acceptance Test Plan*
2013-01-14	*Test Report*
2013-01-15	*Final Project Presentation*
2013-01-20	*Final Project Reports and Documents*
2013-01-20	*Final Product*

3. Organization

The team is organized as presented above:

- ✓ Project leader (POLIMI): Stefania Pezzetti
- ✓ Team leader Italy (POLIMI): Stefania Pezzetti
- ✓ Team leader Croatia (FER): Marko Kelava

Local teams are daily management by the two team leaders: Stefania is responsible for the Italian team and in general for coordinating the entire work, while Marko is responsible for the Croatian team. For facilitating the communication between members, tools like Google Groups, Gmail, Skype, Subversion and Google Drive will be used.

3.1 Project Group

Name	Initials	Responsibility	Roles
Stefania Pezzetti	**SP**	Project Leader Team Leader Requirements Manager	Web Developer
Marko Kelava	***MK***	Team Leader Database Manager	Business Logic Developer
Gregorio Perego	GP	Documentation Manager Web Manager	Web Developer
Bojan Kosanović	BK	System Architect	Business Logic Developer
Tomislav Tolj	TT	SVN Manager Server Manager	Business Logic Developer
Petar Kekez	PK	Test Manager	Mobile Developer
Vedran Šikić	VS	Mobile Manager	Mobile Developer

Each responsibility is performed by one person. This helps in having a clear understanding about the different fields in the project. The responsible person is involved in coordinating the necessary tasks and keeping track of the progress in his own domain.

Project Leader
The project leader keeps control of the overall development process and guides the team to the required targets. His goal is to keep the team on schedule and hold communication channels active.

Team Leader
The team leader is responsible for the daily management of his assigned team. There is a team leader for each geographical located team and his goal is to stay informed about his team's status and progress, and discuss future progress with other team leaders.

Documentation Manager
Documentation manager reviews all the written contents, and makes sure that documents are consistent and conform to the required format. A second duty is that this member coordinates the documentation of the implementation and testing activities.

Requirements Manager
Analyzing, tracing, prioritizing requirements, controlling their change and communicating them to relevant stakeholders. His goal is to broadcast requirements of the project to other members, so that they are aware of the context of the application.

Server Manager
Server Manager is in charge of the installed software on the team's server. He is also in charge of deployment of

any version of the application on the machine.

SVN Manager
SVN manager applies the SVN policy and makes sure that the SVN repository is kept clean and consistent.
This Manager keeps also track of the backup processes, and verifies that these backups are usable. He is also in charge of any recovery processes. His goal is to keep the team able to work on their code, documents and files.

System Architect
System Architect designs the general architecture of the system. His goal is to create the architecture that can fulfill the demands of the requirements.

Database Manager
Database manager designs the supporting database model, and keeps control of the consistency of the database and any changes made to the database.

Test Manager
Test Manager coordinates all test cases, and is in charge of performing the tests, and reports the results of these tests to other team members.

Mobile Manager
Mobile Manager establishes design and interface of the mobile application.

Web Manager
Web Manager establishes design and interface of the web application.

3.2 Customer

The project customer is Raffaela Mirandola, who is also the supervisor. Customers for the final product could be doctors and medical divisions.

3.3 Supervisor

The project supervisor is Raffaela Mirandola, professor at POLIMI.

3.4 Others

Other people interested in the evolution of the project are all professors involved in DSD course, from POLIMI (Italy), MDH (Sweden) and FER (Croatia).
Also Andrea Ciancone from POLIMI, an expert in mobile development, is involved in the project as consultant for the implementation.
Finally Simone Pellegrino and Fabrizio Quadrio are involved in the project as medical science domain experts.

4. Development process

While the course schedule pushes us towards a classic waterfall model, we feel that this approach is outdated and not viable in any non-trivial system development, so we opted for something similar, but with more flexibility concerning unavoidable changes to implementation and design - RUP (Rational Unified Process). RUP is an iterative software development process framework created by a division of IBM. It employs a lifecycle of four phases:

1. *Inception Phase*

   The primary objective is to scope the system adequately as a basis for validating initial costing and budgets. In this phase the business case which includes business context, success factors (expected revenue, market recognition, etc.), and financial forecast is established. To complement the business case, a basic use case model, project plan, initial risk assessment and project description (the core project requirements, constraints and key features) are generated. After these are completed, the project is checked against the many criteria.

   If the project does not pass this milestone, called the Lifecycle Objective Milestone, it either can be cancelled or repeated after being redesigned to better meet the criteria.

2. *Elaboration phase*

   The primary objective is to mitigate the key risk items identified by analysis up to the end of this phase. The elaboration phase is where the project starts to take shape. In this phase the problem domain analysis is made and the architecture of the project gets its basic form.

   The outcome of the elaboration phase is:

- A use-case model in which the use-cases and the actors have been identified and most of the use-case descriptions are developed. The use-case model should be 80% complete.
- A description of the software architecture in a software system development process.
- An executable architecture that realizes architecturally significant use cases.
- Business case and risk list which are revised.
- A development plan for the overall project.
- Prototypes that demonstrably mitigate each identified technical risk.

  This phase must pass the Lifecycle Architecture Milestone criteria. If the project cannot pass this milestone, there is still time for it to be cancelled or redesigned. However, after leaving this phase, the project transitions into a high-risk operation where changes are much more difficult and detrimental when made.

3. *Construction phase*

   The primary objective is to build the software system. In this phase, the main focus is on the development of components and other features of the system. This is the phase when the bulk of the coding takes place. In larger projects, several construction iterations may be developed in an effort to divide the use cases into manageable segments that produce demonstrable prototypes.

   This phase produces the first external release of the software. Its conclusion is marked by the Initial Operational Capability Milestone.

4. *Transition phase*

   The primary objective is to 'transit' the system from development into production, making it available to and understood by the end user. The activities of this phase include training the end users and

maintainers and beta testing the system to validate it against the end users' expectations. The product is also checked against the quality level set in the Inception phase.

If all objectives are met, the Product Release Milestone is reached and the development cycle is finished.

Six Best Practices

Six Best Practices as described in the Rational Unified Process is a paradigm in software engeneering, that lists six ideas to follow when designing any software project to minimize faults and increase productivity. These practices are:

Develop iteratively
It is best to know all requirements in advance; however, often this is not the case. Several software development processes exist that deal with providing solution on how to minimize cost in terms of development phases.

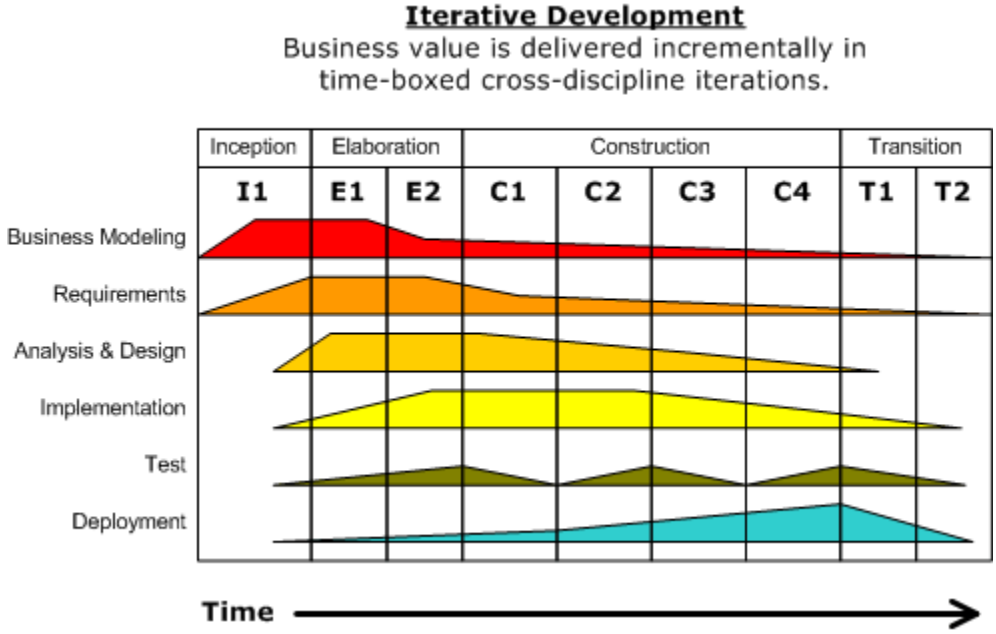
Manage requirements
Always keep in mind the requirements set by users.

Use components
Breaking down an advanced project is not only suggested but in fact unavoidable. This promotes ability to test individual components before they are integrated into a larger system. Also, code reuse is a big plus and can be accomplished more easily through the use of object-oriented programming.

Model visually
Use diagrams to represent all major components, users, and their interaction. "UML", short for Unified Modeling Language, is one tool that can be used to make this task more feasible.

Verify quality
Always make testing a major part of the project at any point of time. Testing becomes heavier as the project progresses but should be a constant factor in any software product creation.

Control changes
Many projects are created by many teams, sometimes in various locations, different platforms may be used, etc. As a result it is essential to make sure that changes made to a system are synchronized and verified constantly.



While we may not follow RUP to the point and we may adapt it to our needs, it is the general development model we will employ.

5. Milestones

Id	Milestone Description	Responsible Dept./Initials	Finished week				Metr.	Rem.
			Plan	Forecast		Actual		
				Week	+/-			
M001	Project Plan	SP - GP	44			43		
	Requirement Engineering	MK - GP	44			44		
M002	Project Design	SP - MK	45			45		
M003	Policies	VS	46			46		
M004	Alpha prototype	SP – MK	48			48		
M005	Beta prototype	SP – MK	51			51		
M006	Acceptance Test Plan	GP	1			1		
M007	Test Report	MK	3			3		
M008	Installation Manual	SP – MK	3			3		
	User Manual	SP – MK	3			3		
	Final Product	SP – MK	3			3		
	Final Project Report	SP – MK	3			3		

6. Project Results

6.1 Requirements

Id	Requirement Description	completed	Rem
GEN-1	Four user roles: administrator, doctor, first aid, patient.	YES	
GEN-2	Registration of first aid users not necessary: only a single first aid user is enough.	YES	
GEN-3	The system implements access control by mandatory logins with a user name and password. The interface and the functionality depend on the role of the user that has logged into the system.	YES	
GEN-4	Outside users have limited access to the system.	YES	
GEN-5	Access to the system using a web browser or a client application for smartphones.	YES	
GEN-6	Mobile application will be developed for Android.	YES	
GEN-7	Mobile application might be developed for Windows Phone.	PARTIALLY	8
GEN-8	Minimum supported browsers: Google Chrome.	YES	
GEN-9	A new measurement of vital parameters input cannot be accepted by the system unless all mandatory vital parameters values have been entered.	YES	
GEN-10	The administrator can only use the web application.	YES	
GEN-11	Doctors can use the web application or the mobile application.	YES	
GEN-12	Patients can use the web application or the mobile application.	YES	
GEN-13	First aid can only use the web application.	YES	
GEN-14	System should use http basic for authorization	DROPPED	1
GEN-15	System should use https for encryption	DROPPED	2
GEN-16	System should take no more than 2 minutes using GPS to get location	YES	
	Administrator		
ADM-1	Adding and deleting doctors.	YES	
ADM-2	Deleting a doctor deletes all his patients.	YES	
ADM-3	Change the login password for the first aid.	YES	
ADM-4	Change his personal login password.	YES	
	Outside users		
OU-1	Check doctor’s public information.	YES	
OU-2	Downloading doctor’s public information as pdf.	YES	
	Doctors		
DR-1	Edit his public information displayed to outside users.	YES	
DR-2	Register a new patient to the system.	YES	
DR-3	Delete a patient from the system. Deleting a patient from the system removes all the data concerning the patient.	YES	
DR-4	Add, edit and delete notes from patient’s medical history.	YES	
DR-5	Adding a note to a patient’s medical history needs to support adding images.	YES	
DR-6	Define patient’s mandatory vital parameters during or after registration.	YES	
DR-7	For each vital parameter of a patient, set a normal interval, which doesn’t trigger an alarm. This option should be available during or after registration.	YES	
DR-8	Input and delete measurements of vital parameters of the patient.	YES	
DR-9	Send alarm with patient’s data to the first aid.	YES	
DR-10	Change his login password.	YES	
DR-11	Edit patient’s personal data	YES	
	Patients		
PA-1	Each patient has exactly one assigned doctor.	YES	

PA-2	View and edit his personal data, including location.	DROPPED	3
PA-3	Input a new measurement of vital parameters.	YES	
PA-4	Update location from web by inputting the address.	YES	
PA-5	Update location using GPS (mobile application).	YES	
PA-6	Send an alarm from a mobile device without specifying any information.	YES	
PA-7	After registration receives an email with his login password. Then he can change his login password from his personal home page both with web application and mobile applications.	YES	
	System		
SER-1	Alert doctor about patient's abnormal vital parameters via e-mail.	YES	
SER-2	Notify unusual alarm activities to the first aid. An unusual alarm activity is triggered when the system notices that 15 alarms have been raised within 15 minutes inside a 100 meter radius circle.	DROPPED	4
	First Aid		
FA-1	Display list of alarms sent by patients or doctors. The alarm contains patient's data.	YES	
FA-2	Get alerted when a patient or a doctor sends an alarm.	YES	
FA-3	Delete received alarm from patients or doctors.	YES	
FA-4	Get alerted when the system recognizes an unusual alarm activity.	DROPPED	5
FA-5	Display unusual alarm activities. A raised unusual alarm activity has to contain the GPS location of the unusual activity and the number of raised alarms.	DROPPED	6
FA-6	Delete received alarm for unusual activities.	DROPPED	7
FA-7	Display a map that explains where the patient is when he needs help.	YES	

Completed: Yes (completely implemented)
No (not implemented at all)
Partially (partially implemented, more description under Remarks subsection)
Unknown (completion status not known)
Dropped (requirement was dropped during the course of the project)

6.1.1 Requirements Compliance Summary

Summarize the requirements compliance data.

Total number of requirements	49
Number of requirements implemented	0
Requirements partially fulfilled	0
Requirements not fulfilled	0
Requirements dropped	42

6.1.2 Remarks

Remarks id	Comments
3	Dropped because the doctor will be in charge to change patient's personal data
2	Dropped because the customer does not require it.
4	Dropped because it seems inconsistent with other requirements.
5	Dropped because the requirement SER-2 was deleted.
6	Dropped because the requirement SER-2 was deleted.
7	Dropped because the requirement SER-2 was deleted.
1	Dropped because the customer does not require it.
8	Windows phone is finished more or less at 80%.

6.2 Deliverables

To	Output	Planned week	Promised week	Late +/-	Delivered week	Rem
Project Stakeholders Project Team members	*Project Plan*	44	44	0	44	
Project Stakeholders Project Team members	*Requirements Definition Document*	44	44	0	44	
Project Stakeholders Project Team members	*Design Description Document*	45	45	0	45	
Project Stakeholders Project Team members	*Acceptance test plan*	1	1	0	1	
Project Stakeholders Project Team members	*Test report*	3	3	0	3	
Project Stakeholders Project Team members	*Final Project Report*	3	3	0	3	
Project Stakeholders Project Team members	*Final versions of all documents*	3	3	-	3	
Project Stakeholders Project Team members	*Technical Documents*	3	3	-	3	
Project Stakeholders Project Team members	*User Manual*	3	3	-	3	
Project Stakeholders Project Team members	*Final Product*	3	3	-	3	

7. Risks

Below is reported the table written in our Project Plan at the beginning of the project:

Possibility	Risk	Preventive action
High	Personal lack of time due to other courses or obligations	Weekly report to check status, arrange time in advance
High	Lack of technical expertise in the chosen implementation technologies	Cooperative learning within the team, and share good tutorials
Medium	Wrong interpretation of given requirements and boundaries	Have detailed use cases. Discuss to other team members if not clear about requirements.
Medium	Different understanding between team members	First discuss, then decide. Record decision in documents, for later reference.
Medium	Team Members are not reachable	Make sure that everyone knows what the other is doing so if someone doesn't complete his work another team member can complete that work.
Low	Member leaves	Have back up for each role. Have good documentation for work.
Low	Database Server Crash	Regular backup of database.
Low	SVN Server Crash	Regular backup at safe location

- ***Personal lack of time due to other courses or obligations***: this risk appeared but we divided work in a way that allowed us to work and to reach on time each milestone.
- ***Lack of technical expertise in the chosen implementation technologies***: this risk appeared but as we wrote in the project plan we shared our knowledge and read good tutorials.
- ***Wrong interpretation of given requirements***: this risk appeared in part but we asked mentor to explain very well requirements and we wrote together a document where everyone could present their doubts, if someone was able to respond he or she replied, otherwise we asked mentor.
- ***Different understanding between team members***: we did exactly what we wrote in the project plan. We wrote documents and Minutes of Meetings to remember what everyone should do. We shared these documents on Google Drive and we didn't have any problems.
- ***Team Members are not reachable***: this risk sometimes happened but we reached him with emails after few days.
- ***Member leaves***: this risk didn't happen.
- ***Database Server Crash***: this risk didn't happen.
- ***SVN Server Crash***: this risk didn't happen.
- ***Problems with switching from Repository to Database***: we didn't foresee this problem and we worked a lot to solve it. We had to change technology in order to solve the problem.
- ***Problems with Deployment***: we didn't foresee this problem. We gave up with VMware (that was the technology that we wanted use at the beginning) and we used Windows Azure for deployment.

8. Project Experiences

8.1 Positive Experiences

Most of the team members have acquired valuable experience about big projects implementation. Some of them had not experienced any big size software project before.
Furthermore, we have acquired an understanding about the effort it takes to proper communicate and cooperate with each other. This distributed environment enforced actually us to have a bigger focus on 'formal' communication. The great effort we paid in the beginning of the project on requirements and design definitely helped us keeping the project in direction. We, also, learnt to communicate each other quickly on Skype and to go straight to the point and not be afraid to ask when something doesn't work. We improved our language and we learnt to relate with who is far from us.
Besides learning communication, we also learned new technologies that can help us in the future, from the basics of SVN, to the more advanced used of the Entity Framework, which supported the data persistency in our project. We all learnt to use JSON for communication between Business Logic and Front End. The Italian Team learnt .NET and C# starting from scratch.
Furthermore, we made of course new friendships, both with local and remote team members.

8.2 Improvement Possibilities

Our main problem was Knowledge. We had different skills and we tried to use the differences to make a good product. For instance we decided to develop also the Windows Phone application, besides the Web application and the Android application, because one team member was able to develop for Windows Phone before the project. However some team members had to learn new technologies and this fact involved "loss" of time.

9. Metrics

9.1 Work per Member

Abbreviation	Member
SP	Stefania Pezzetti
MK	Marko Kelava
GP	Gregorio Perego
BK	Bojan Kosanović
TT	Tomislav Tolj
PK	Petar Kekez
VS	Vedran Šikić

Member	W43	W44	W45	W46	W47	W48	W49	W50	W51	W52	W1	W2	W3	Total
SP	20.5	20	13	21	48	19	8	35	13.5	26	13	12.5	9	258.5
MK	12.5	18	13	13	48	3	5	24	7.5	6	7	9.5	5	171.5
GP	18	19	15	21	57	14	9	35	15	20	12	13	6	254
BK	6	22	12	38	6	2	47	24	11.5	15	2	5.5	1	192
TT	11.5	12	11	14	35	2	21	15	3	5	9	10	5	153.5
PK	5	15	8	8	4	20	19	8	16	11	4	25.5	11	154.5
VS	15	11	24	10	48	5	20	26	6	7	0	29	16	217
Total	88.5	117	96	125	246	65	129	167	72.5	90	47	105	53	1401

9.2 Milestone Metrics

Completed as planned or earlier	Total	Timeliness
8	8	100%

9.3 Effort Metrics

ID	Activity	Actual Effort	Planned Effort	Deviation (%)
1	Documentation (requirements gathering, requirements analysis, architecture design, technical documentation)	213	294	62%
2	Implementation	742	882	82%
3	Testing	15	21	60%
4	Deployment	58	42	73%
5	Presentations	76	-	-
6	Study	262	-	-
8	Official Meetings	26	-	-

We didn't forecast hours for meetings, presentations and study, but they were obviously necessary. Testing accuracy it's difficult to calculate because we tested the system also during the implementation so implementation and testing are not so separate. Probably there can be some testing hours in the implementation activity.

Effort estimation accuracy (%) *(100*(1 - abs(Actual – Planned)/Actual))*	89%