Norwegian University of Science and Technology

National Integration Platform for Citizen Centric eHealth in Norway

Authors:
Anders Olsen Sandvik,
Sebastian Zalewski,
Emanuele Di Santo

Supervisor:
Meng Zhu

TDT4290 Group 17
November 2013

NTNU – Trondheim
Norwegian University of
Science and Technology

Helsedirektoratet
Norwegian Directorate of Health

Abstract

This report describes the design of the National Integration Platform (NIP) used for Citizen Centric eHealth in Norway. The NIP is a platform to collect different user generated eHealth data from third party solutions and services. It also describes the making of the working prototype. The challenges of this prototype is to demonstrate the transfer of personal and/or medical data from different devices and systems. The intention of the platform is to enable citizens the ability to publish information they produce into the government run NIP. It is worth noting that security is not a requirement of the prototype. To demonstrate these different parts the creation of an Android application (App), a web App and a back-end Application Programming Interface (API) was required.

The motivation for working in this project is to innovate and create a new platform that can be used to better understand the health of citizens. The citizens gather various health data from different locations and devices and in return provide a better understanding of the citizens health. Together with educated health professionals and doctors this can be a powerful tool for improving the quality of life of the users. The most interesting part was to figure out how to design a system that can have the high level of security required to transfer personal health information.

The demands were to plan, design and describe the NIP and to develop a prototype. The demonstration of this product is aimed mainly at two groups of people:
1. Educated health professionals
2. Developers

The product should have a demonstration side that is easy for the first group to grasp, understand and form an idea of how it will work while also making it appealing and technical for the second group.

The result of this project is first and foremost a working prototype of a NIP. It is also a web App and an Android app to make the demonstration easier to grasp. This report is also part of the result which has the purpose of documenting the problem, the process, the workflow and the final products.

Preface

This report is for the main project in the course TDT4290 Customer Driven Project at NTNU. The project was executed on behalf of Helsedirektoratet or The Norwegian Directorate of Health. The team consisted of three students from NTNU.

The team would like to thank our supervisor Meng Zhu for his guidence, help and advice for this project.

In addition we would also like to thank our customer Helsedirektoratet and their contact person Helge T. Blindheim for their effort.

Contents

Abstract i

Preface ii

List of Figures vii

List of Tables viii

Abbreviations ix

1 Introduction 1
- 1.1 Project description . . . 1
- 1.2 The client . . . 2
- 1.3 Involved parties . . . 2
- 1.4 Project background . . . 3
- 1.5 Problem domain . . . 3
- 1.6 Project objective . . . 3
- 1.7 Duration . . . 4

2 Project management 5
- 2.1 Planning . . . 5
  - 2.1.1 Work plan . . . 5
  - 2.1.2 Resources . . . 5
  - 2.1.3 Limitations . . . 5
  - 2.1.4 Milestones . . . 5
  - 2.1.5 Tool selection . . . 5
- 2.2 Organization . . . 6
  - 2.2.1 Roles . . . 6
  - 2.2.2 Weekly schedule . . . 6
- 2.3 Quality assurance . . . 6
  - 2.3.1 Templates . . . 6
  - 2.3.2 Customer relations . . . 6
  - 2.3.3 Supervisor relations . . . 6
- 2.4 Risk management . . . 6

3 Preliminary Studies 7
- 3.1 Development Methodology . . . 7

3.1.1 Waterfall Model . . . 7
3.1.2 Scrum Model . . . 8
3.1.3 Conclusion . . . 10
3.2 Earlier Experience . . . 10
3.2.1 Competencies . . . 10
3.2.2 Conclusion . . . 10
3.3 Existing Solutions . . . 11
3.3.1 HealthVault . . . 11
3.3.2 Open eHealth Foundation . . . 11
3.3.3 Heart Rate App . . . 11
3.3.4 human/api . . . 12
3.3.5 Conclusion . . . 12
3.4 Used Technologies . . . 12
3.4.1 Server . . . 12
3.4.2 Database . . . 13
3.4.3 Web Page . . . 13
3.4.4 Mobile Technologies . . . 14
3.4.5 Other Technologies . . . 15
3.4.6 Conclusion . . . 15
3.5 Testing . . . 16
3.5.1 JUnit . . . 16
3.5.2 Conclusion . . . 16
3.6 Summary . . . 16

4 Requirements specification **17**
4.1 Stakeholders . . . 17
4.2 Funcional requirements . . . 17
4.3 Non-funcional requirements . . . 18
4.3.1 Quality requirements . . . 18
4.4 Use cases . . . 18

5 System architecture **19**
5.1 Overview . . . 19
5.2 NIPEN . . . 19
5.3 Front-end . . . 19
5.4 Heart rate . . . 19
5.5 Weight . . . 19

6 Sprint 0 **20**
6.1 Planning . . . 20
6.2 Duration . . . 21
6.3 Goals . . . 21
6.4 Feedback . . . 21
6.5 Problems . . . 21
6.6 Evaluation . . . 21

7 Sprint 1 **22**
7.1 Planning . . . 22

7.1.1 Expected results . . . 22
7.2 Feedback . . . 22
7.3 Evaluation . . . 22

8 Sprint 2 **23**
8.1 Planning . . . 23
8.1.1 Expected results . . . 23
8.2 Feedback . . . 23
8.3 Evaluation . . . 23

9 Sprint 3 **24**
9.1 Planning . . . 24
9.1.1 Expected results . . . 24
9.2 Feedback . . . 24
9.3 Evaluation . . . 24

10 Sprint 4 **25**
10.1 Planning . . . 25
10.1.1 Expected results . . . 25
10.2 Feedback . . . 25
10.3 Evaluation . . . 25

11 Sprint 5 **26**
11.1 Planning . . . 26
11.1.1 Expected results . . . 26
11.2 Feedback . . . 26
11.3 Evaluation . . . 26

12 Testing **27**
12.1 Main Section 1 . . . 27
12.1.1 Subsection 1 . . . 27
12.1.2 Subsection 2 . . . 27
12.2 Main Section 2 . . . 28

13 Conclusion and further work **29**
13.1 Main Section 1 . . . 29
13.1.1 Subsection 1 . . . 29
13.1.2 Subsection 2 . . . 29
13.2 Main Section 2 . . . 29

14 Reflection **30**
14.1 Main Section 1 . . . 30
14.1.1 Subsection 1 . . . 30
14.1.2 Subsection 2 . . . 30
14.2 Main Section 2 . . . 31

A Assignment for Customer driven project **32**

B How to build the project **35**

C Templates **36**
C.1 Weekly status report . . . 36
C.2 Agenda for advisor meeting . . . 37
C.3 Minutes of advisor meeting . . . 37
C.4 Agenda for customer meeting . . . 38
C.5 Minutes of customer meeting . . . 39
C.6 Agenda for internal meeting . . . 39
C.7 Minutes of internal meeting . . . 39

D Advisor meeting documents **40**
D.1 Weekly status report . . . 41
D.1.1 Week 35 . . . 41
D.1.2 Week 36 . . . 41
D.1.3 Week 37 . . . 41
D.1.4 Week 38 . . . 41
D.1.5 Week 39 . . . 41
D.1.6 Week 40 . . . 41
D.1.7 Week 41 . . . 41
D.1.8 Week 42 . . . 41
D.1.9 Week 43 . . . 41
D.1.10 Week 44 . . . 41
D.1.11 Week 45 . . . 41
D.1.12 Week 46 . . . 41
D.1.13 Week 47 . . . 41
D.2 Agenda for advisor meeting . . . 41
D.2.1 1 . . . 41
D.3 Minutes of advisor meeting . . . 41

E Customer meeting documents **42**
E.1 Agenda for customer meeting . . . 42
E.2 Minutes of customer meeting . . . 42

F Internal meeting notes **43**
F.1 Agenda for internal meeting . . . 43
F.2 Minutes of internal meeting . . . 43

List of Figures

3.1 Waterfall model 9
3.2 Scrum workflow 9

List of Tables

1.1 Customer representative 2
1.2 Team members 2
1.3 Student advisor 3

3.1 Competencies 11

C.1 Activity chart 36
C.2 Activity chart 37
C.3 Activity chart 38

Abbreviations

AJAX	**A**synchronous **Ja**vaScript and **X**ML
API	**A**pplication **P**rogramming **I**nterface
App	**A**pplication
CSS	**C**ascading **S**tyle **S**heets
DOM	**D**ocument **O**bject **M**odel
HTML	**H**yper**T**ext **M**arkup **L**anguage
IDI	**I**nstitutt for **D**atateknikk og **I**nformasjonsvitenskap
JSP	**J**ava**S**erver **P**ages
NIP	**N**ational **I**ntegration **P**latform
NTNU	**N**orges **T**eknisk-**N**aturvitenskalige **U**niversitet
POM	**P**roject **O**bject **M**odel
SDK	**S**oftware **D**evelopment **K**it

Chapter 1

Introduction

This chapter is an introduction to the project. It describes the project in 1.1. It describes the client 1.2, the involved parties 1.3 and discusses the project background 1.4. It also considers the problem domain 1.5, the project objective 1.6 and mentions the timeline of the project in 1.7.

1.1 Project description

The purpose of the project is to design, develop and document an integration platform for citizen health in Norway. The intention of such platform is to allow citizens to publish date relating their health produced by devices or third party solutions. It is very common today for people to log health data from mobile phones and tablets in their possession. The project assignment can be read in Appendix A.

For the client it was important that this project produced a prototype that could be demonstrated to:

1. Educated health professionals
2. Developers

Security is of high importance when dealing with citizen health data. This was not made a requirement of the project beacue the assignment had to be scaled down because the group only consisted of three members. We will however discuss how you could add a secure layer to the working solution.

The name we chose for the project is **NIPEN**. It is simply an acronym for **N**ational **I**ntegration **P**latform for **e**Health in **N**orway.

1.2 The client

The customer of this project was the Department of the Health portal, Norwegian Directorate of Health (Helsedirektoratet).

The Directorate has, among other, the task of digitalizing Norway's health care system by providing services for both specialists and citizens. The customer was represented by Mr. Helge T. Blindheim. His contact is shown in table 1.1. Their office is located in the capital of Norway, Oslo. This infered that our weekly meeting had to be heald over teleconferencing.

Name	Phone	E-mail
Helge T. Blindheim	46675321	Helge.T.Blindheim@helsedir.no

Table 1.1: Customer representative

1.3 Involved parties

The people involved in this project were the customer, the team and the supervisor. The customer, introduced in the previous section, was represented by Mr. Helge T. Blindheim. The team consisted of three students from the Department of Computer and Information Science (IDI) at the Norwegian University of Science and Technology (NTNU). Their contact information is shown in table 1.2. The group was supervised by PhD. candidate Zhu Meng. His contact information is shown in table 1.3.

Name	Phone	E-mail
Anders Olsen Sandvik	91824583	andsan@stud.ntnu.no
Emanuele Di Santo	...	lemrey@gmail.com
Sebastian Zalewski	95107928	zalewski@stud.ntnu.no

Table 1.2: Team members

Name	Phone	E-mail
Zhu Meng	73551189	zhumeng@idi.ntnu.no

Table 1.3: Student advisor

1.4 Project background

This project is part of the Customer Drivent Project (TDT4290) at NTNU. Digital healthcare is about using information technologies to provide solutions to problems in healthcare. This definition includes a lot of different domains among which is eHealth. eHealth is

> an emerging field in the intersection of medical informatics, public health and business, referring to health services and information delivered or enhanced through the Internet and related technologies.[**?**]

eHealth projects are therefore long, complex and inherently costly. At the same time the progress in information technology has made available powerful and yet cheap devices which can be used to monitor health. These devices are nowdays widespread and used by a large part of the population.

1.5 Problem domain

1.6 Project objective

The goal of the customer is to investigate the possibilities for national eHealth's projects to leverage the dynamics of the market.

The purpose of the course is to let students acquire practical experience in development of a medium-large software project, including experience in project management, group dynamics and customer relations.

1.7 Duration

The project started on august 21th and the final presentation is on november 21th. That gives us a total of 13 weeks to work on the project. The instructors specify a workload of 24 hours per week according to the course page[*]. That makes a total of 312 hours per student. We are a group of three students which makes the total 936 hours.

Start date: 21.08.2013 End date : 21.11.2013

[ADD TO BIB] Retrived 25.09.2013 http://www.ntnu.edu/studies/courses/TDT4290/2013

Chapter 2

Project management

2.1 Planning

2.1.1 Work plan

2.1.2 Resources

2.1.3 Limitations

2.1.4 Milestones

2.1.5 Tool selection

This section will describe the different tools we used during this course

Git and GitHub "Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency." [bib]http://git-scm.com/

Sublime Text

Intelijei IDE

Google Docs

skype

Apache Maven

Travis CI

Latex

Balsamiq Mockups

Lucidchart

2.2 Organization

2.2.1 Roles

2.2.2 Weekly schedule

2.3 Quality assurance

2.3.1 Templates

2.3.2 Customer relations

2.3.3 Supervisor relations

2.4 Risk management

Chapter 3

Preliminary Studies

This chapter consists of a discussion of the development methodologies in 3.1, and some of our earlier experiences in section 3.2. In 3.3 we go through some of the already exiting solutions, and in 3.4 we go through the technologies used in our project. Section 3.5 consists of the testing methods used and 3.6 is a summary of the chapter.

3.1 Development Methodology

In this section we are going to present and discuss two common development methodologies. We are going to conclude with what type of development methodology we chose and why we chose it. This was one of the first discussions we had before starting the project, since without a good development process a project is often doomed to failure. It is important to divide the work into task and work in a structured manner, in this way we will have a better understanding of where the project is and where it is headed.

3.1.1 Waterfall Model

The waterfall model is a software development process where each task is performed in a sequential order. Before moving to the next phase the preceding task needs to be finished. The progress of the project is seen as flowing downwards through the different phases, hence the name waterfall. This is represented in figure 3.1 where it is possible

to see how the progress is flowing. In the original model the phases consisted of seven different tasks:

1. Requirements specification
2. Design
3. Construction (implementation or coding)
4. Integration
5. Testing and debugging
6. Installation
7. Maintenance

Because each phase needs to be perfected and completed before moving to the next phase, it will bring up some difficulties if the requirements were to change during the development process. However, the model is easily understandable, structured, and disciplined. All the phases are divided into different sections, and this makes it easier to understand the progress of the project. In practice it can be very hard to adapt to this kind of development model. It can be hard for a system designer to predict future implementation difficulties of a type of design, hence the design of the system may change during the process. Another problem is that the customer is not always sure about the system requirements, and often the customer will change them during the development.

3.1.2 Scrum Model

The scrum model is an agile software development process that is iterative and incremental. It consists of multiple sprints, where a sprint usually lasts from 14-30 days depending on the size of the task. Each sprint is focusing on a set of concrete goals that are in the sprint backlog. The sprint backlog consists of tasks that are chosen from the product backlog. They are usually chosen in the sprint planning meeting that is performed before each sprint. The product backlog consists of all the features the product should contain, and is usually made in the initial phase of the project. It can

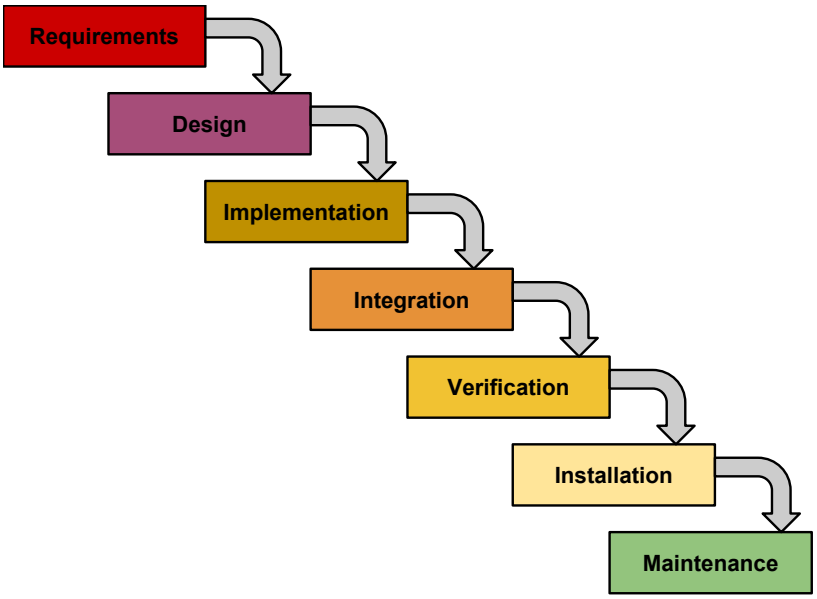


FIGURE 3.1: Waterfall model

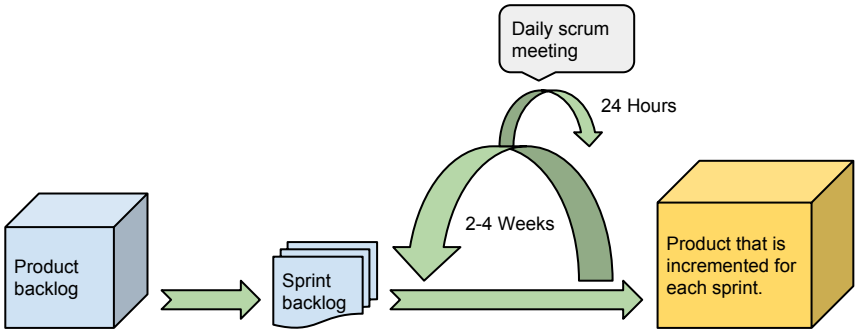


FIGURE 3.2: Scrum workflow

however be changed and adjusted during the development of the product. For each day a daily scrum meeting should be performed. Usually a scrum meeting consists of getting to know what each person did yesterday, what they will do today and if they are facing any problems. If there are any problems the Scrum master is responsible to resolve the problem. After each sprint a sprint review meeting should be held. An overview of what goals where achieved and which one where not should be made. The meeting can also consist of a demo of the new features implemented. Figure 3.2 shows the workflow of scrum.

3.1.3 Conclusion

We decided to choose the scrum model as our development process. The reason for this is that we didn't have a formal requirement list of the project from the beginning. Thus it was a high probability that we didn't understand the problem completely, and might have done some wrong design decisions in the first phase of the project. Our customer also advised us to use an agile method as our development methodology. We thought that scrum would be a good methodology as it is a widely known and respected development process in the programming community. The ability to change the design and choices we had made earlier in the project was very important to us. If we for example were to make a choice that the customer wasn't pleased with, we would have the ability to fix the problem in the next sprint. This would have been very hard to do in a non agile development methodology like the waterfall model.

3.2 Earlier Experience

This section contains a short description of our earlier experiences. In the conclusion we give some short details of what kind of effect this had on our project.

3.2.1 Competencies

One of the first things we did during our first meetings was to get to know each others experiences and competencies. This was important for us, so we could decide what type of technologies to use in our project. In table 3.1 we have listed our competencies in the different languages and frameworks. We gave each technology a score from 1 to 5, where 5 indicated that the language or framework was well known and 1 indicated that it was not known.

3.2.2 Conclusion

We managed to find out our competencies, and fortunately many of us already had the experience that was needed to start a project like this. This was of great help to us when deciding what technologies to use in our project. In table 3.1 it is also visible that

	Emanuele	Anders	Sebastian
HTML	4	4	4
CSS	1	4	4
JavaScript	2	4	5
Java	5	5	5
Maven	2	3	1
Angular	1	2	1
Git	4	5	4
NodeJS	1	1	1
jQuery	1	1	4
SQL	3	4	4
Spring	1	3	1

TABLE 3.1: Competencies

some of us were more familiar with some technologies than other. This made it easier to divide the tasks between each other in the project.

3.3 Existing Solutions

This section contains some of the similar existing solutions that are already created. The following technologies were a good inspiration source and help in our project, as they already were similar or did contain components that our application also would contain.

3.3.1 HealthVault

TODO

3.3.2 Open eHealth Foundation

Emanuele could write something here...

TODO

3.3.3 Heart Rate App

Maybe we should write something about the open source heart rate app here?

TODO

3.3.4 human/api

The human API is a platform for human health data. They have an API that contains multiple different well defined JSON strings for different kinds of human related data. Each JSON string contains all the necessary information that is needed to represent each type of health data. For example heart rate is defined by an id, user id, time, value and unit in the following way:

```
{
  "id": "string",
  "userId": "string",
  "time": "date",
  "value": "int",
  "unit": "string"
}
```

3.3.5 Conclusion

TODO

3.4 Used Technologies

This section contains the technologies we used in our project. We are going to go through what type of technologies we used for the server (back end), web page (front end), database and for the development of applications for mobile units. It also contains some other technologies that were needed in the project.

3.4.1 Server

Java

Java is a general-purpose programming language, which means that it can be used in a wide variety of application domains. It is platform independent, and thus makes it easy to develop software for different operation systems. Web programming is one of

the domains that Java is used for, hence makes it possible to write a service that runs in the back end of a web page. The web services created with Java allows creating a bridge between the database and the front end. This makes it possible to create a secure communication between the user and the database at the server.

Spring Framework

Anders could write something here...

TODO

Apache Tomcat

Apache Tomcat is an implementation of the JSP (JavaServer Pages) and Java Servlet technologies. It makes it possible to deploy and run a web page with its services on a server. Numerous industries and organizations use the Tomcat server as it has support for large-scale and mission-critical web applications.

3.4.2 Database

MySQL

MySQL is one of the most widely used relational database management system. The MySQL Community Edition is open source and freely available. It is developed to handle large databases, support many users at the same time and it is also scalable. It makes it possible to store and retrieve data in an efficient and structured manner.

3.4.3 Web Page

HTML5

HTML is the standard World Wide Web's markup language, where HTML5 is the newest HTML version as of this writing. It is used to structure and visualize web pages on the internet. By writing a document with HTML a web browser is later on able to interpret the document and visualize it in a structured manner.

CSS3

CSS describes the look and format of a document written in HTML. It allows one to use different fonts, colors and adjust the layout of a web page. By using CSS and separating it from the HTML, it is possible to allow multiple pages share the same style sheet. Thus it is easier to maintain and adapt the web pages to different environments through CSS.

Bootstrap

Bootstrap contains HTML and CSS templates for web designers. This makes it easier to make a good looking web page without putting too much effort into the design.

JavaScript

JavaScript is an interpreted computer programming language that runs in the browser of the user. It is allowed to make changes in the HTML DOM, interact with the user, control the browser and communicate asynchronously. Since it can communicate with the server asynchronously, it makes the web page more dynamic. What this means is that a web page can acquire new information and change the site without reloading.

jQuery

jQuery is a JavaScript library for manipulating and traversing the HTML DOM. All the features jQuery contains are also possible to do with pure JavaScript, but jQuery helps the developer to implement the different features in an easier way. For example it contains predefined methods for event handling and animation. It also makes it easier to communicate with the server through AJAX.

Chart.js

Chart.js is a JavaScript library for creating graphs and charts. It helps the developer to visualize data through different types of graphs in an easy manner. The library has support for different types of two dimensional data, e.g. value per time. It also has the ability to display multiple graphs in the same chart.

3.4.4 Mobile Technologies

Android SDK

Android SDK contains the tools necessary for developing, debugging and testing an Android app. With the SDK it is possible to write and modify applications for an Android phone.

3.4.5 Other Technologies

Maven

Maven is a software tool for managing a programming project. It has the ability to build and compile programming code based on the content of a POM (Project Object Model) file. It keeps track of all the frameworks used, and is able to download them before building the project.

Git

Git is a version control system that is free and open source. This is an important tool to keep track of all the changes made to the source code, it also makes it easier for multiple developers to work on the same source. Git makes it easy to roll back changes made to the code, in case something was wrongly implemented. It also has the possibility to divide the project into different branches. Which means that the code can be copied into multiple different places, and developed separately in cases where trying out different solutions is necessary. If a good solution is made, the branch can later on be merged together with the main branch. It is also possible to have a branch for release versions and a development branch.

3.4.6 Conclusion

We chose the technologies specified above based on our earlier experience and what we found most appropriate for our solution. Before we started this project we discussed our competencies. This was of great help in choosing the right technologies that were going to be used. Most of us already had some different degree of experience in most of the technologies mentioned above. This made it much easier for us to chose the right set of frameworks and languages to work with. For example from table 3.1 we can see that all of us were very familiar with Java, Git and SQL. This made it clear for us to use those technologies.

3.5 Testing

TODO

3.5.1 JUnit

TODO

3.5.2 Conclusion

TODO

3.6 Summary

In this chapter we had a discussion of two common development methodologies. We also found out that the scrum model was the best development process for our project. Our earlier experience was described in table 3.1. We also discussed some of the earlier existing solutions that were already created. A description of each technology that was used in this project was given. At the end of the chapter we gave a description of some of the testing methods used.

Chapter 4

Requirements specification

4.1 Stakeholders

4.2 Funcional requirements

Functional Requirements for Integration platform

FRIP1: The integration platform must support api endpoints for receiving heart rate data. Priority: High Difficulty: Medium FRIP2: The integration platform must support API endpoints for receiving weight data. Priority: High Difficulty: Medium FRIP3: The integration platform must store the data it receives Priority: High Difficulty: Low FRIP4: The integration platform must display the user data on user request Priority: High Difficulty: Low

Functional Requirements for Heart-rate application FRHR1: The user should be able to measure his heart rate. Priority: High Difficulty: Low FRHR2: The user must be able to send data to the integration platform using the application. Priority: High Difficulty: Medium

Functional Requirements Withings 'adaptor' FRW1: The service shall receive notification from the third party product (Withings) Priority Difficulty FRW2: The service shall forward data received to the IP. (that could be done manually or be scheduled) Priority Difficulty

Functional Requirements for HealthVault... FHV1:

4.3 Non-funcional requirements

4.3.1 Quality requirements

4.4 Use cases

Chapter 5

System architecture

5.1 Overview

5.2 NIPEN

5.3 Front-end

5.4 Heart rate

5.5 Weight

Chapter 6

Sprint 0

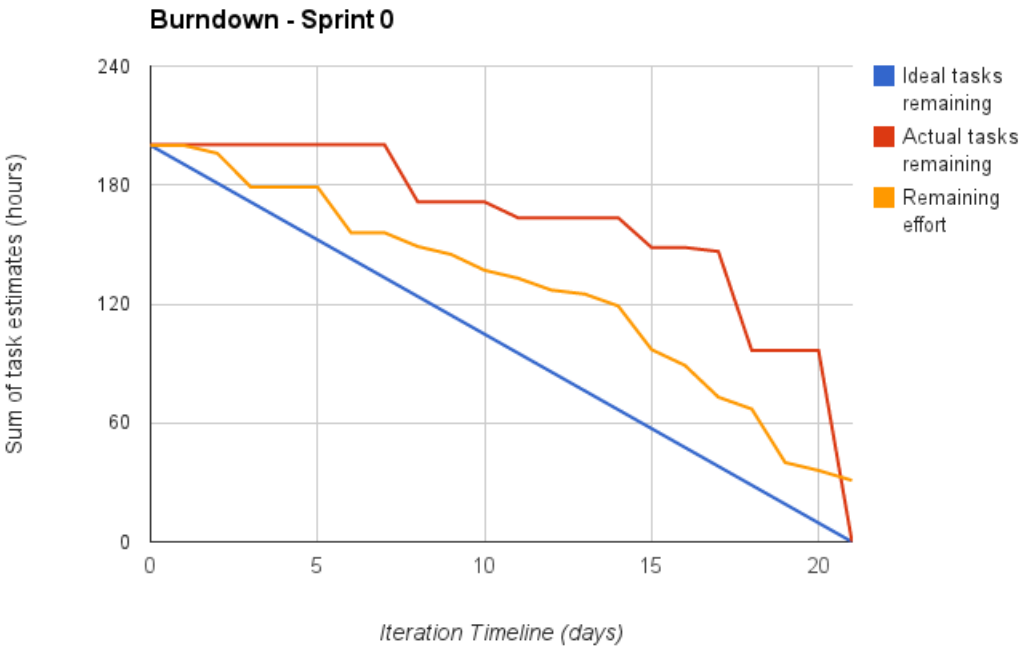
This chapter is meant to give an overview of sprint 0. Section 9.1 gies and overview of the planning. Section 9.2

6.1 Planning

** We planned to have 2 weeks sprint but in the middle of the sprint we changed the first sprint to a 3 week sprint. That makes some of our numebrs inconsistent. Refering to the status reports and the weekly meetings. We usually estimated 60 hours per week of work but the first sprint ended up beeing estimated at 200 hours for 3 weeks**

what we planned to do shall we include some data from scrumdo? definitely a chart..

6.2 Duration



6.3 Goals

what did we expect to achieve by the end of this sprint (general progress in the project)

6.4 Feedback

from customer, from supervisor

6.5 Problems

6.6 Evaluation

our thoughts about this sprint

Chapter 7

Sprint 1

7.1 Planning

what we planned to do shall we include some data from scrumdo? definitely a chart..

7.1.1 Expected results

what did we expect to achieve by the end of this sprint (general progress in the project)

7.2 Feedback

from customer, from supervisor

7.3 Evaluation

our thoughts about this sprint

Chapter 8

Sprint 2

8.1 Planning

what we planned to do shall we include some data from scrumdo? definitely a chart..

8.1.1 Expected results

what did we expect to achieve by the end of this sprint (general progress in the project)

8.2 Feedback

from customer, from supervisor

8.3 Evaluation

our thoughts about this sprint

Chapter 9

Sprint 3

9.1 Planning

what we planned to do shall we include some data from scrumdo? definitely a chart..

9.1.1 Expected results

what did we expect to achieve by the end of this sprint (general progress in the project)

9.2 Feedback

from customer, from supervisor

9.3 Evaluation

our thoughts about this sprint

Chapter 10

Sprint 4

10.1 Planning

what we planned to do shall we include some data from scrumdo? definitely a chart..

10.1.1 Expected results

what did we expect to achieve by the end of this sprint (general progress in the project)

10.2 Feedback

from customer, from supervisor

10.3 Evaluation

our thoughts about this sprint

Chapter 11

Sprint 5

11.1 Planning

what we planned to do shall we include some data from scrumdo? definitely a chart..

11.1.1 Expected results

what did we expect to achieve by the end of this sprint (general progress in the project)

11.2 Feedback

from customer, from supervisor

11.3 Evaluation

our thoughts about this sprint

Chapter 12

Testing

12.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

12.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

12.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio

aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

12.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

Chapter 13

Conclusion and further work

13.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

13.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis.

13.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales.

13.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas.

Chapter 14

Reflection

14.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

14.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

14.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio

aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

14.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

Appendix A

Assignment for Customer driven project

Assignment for Customer driven project

Title: National Integration Platform for Citizen Centric eHealth in Norway
Customer (Company): The Directorate of Health, Department of the Health Portal
Address: Universitetsgata 2, Oslo

Assignment text:

Background

The Directorate of Health has a national the task of digitalizing Norwegian healthcare, both by providing coordinated services for specialist healthcare (hospitals) and by providing digital services for citizens in general and patients specifically. Examples of such services are ePrescriptions, that is implemented on a national basis, the National Summare Care Record, that will go live in Trondheim in August 2013, and the citizen centric health portal (helsenorge.no) that has been live since June 2011.

National eHealth projects are complex, long running and costly. There are obvious reasons for this. Among these are the complexity and criticality of healthcare, and the scale that national eHealth services represents.

At the same time, the trends in technology development and consumer adaption of new technology continue to develop. Moderate prices and consumer friendly devices

that monitor individuals' health and wellness are increasingly becoming available in the market space. Combined with a continuous increase in digital competence in the population, they will influence citizens' behavior and perspective on their own health situation in the future.

In addition to this, private providers develop great eHealth solutions with consumer and patient orientation. Medhelp.org and Healthvault.com are only two among many examples. Ambient assisted living has the potential of revolutionizing life for senior citizens with failing health.

The relevant question is: How can the substantial and long running eHealth projects of the government sector connect to and leverage the dynamics in the market and consumer behavior? The answer under investigation is the National Integration Platform (NIP) for Citizen Centric eHealth in Norway.

Assignment
The assignment is to plan, design and describe a NIP, and to develop a prototype.

The task such a platform should fulfil is to offer interoperability with third party solutions based on available application programmable interfaces (APIs). All third party solution providers must adhere to specified and standardized rules regarding authentication, security model, messaging and privacy to interact with the NIP.

The intention of such a platform is to enable the following:

- Citizens' ability to publish information they produce from devices in their possession and third party software solutions, including smart phone and tablet apps, into the government run citizen centric health portal (helsenorge.no)
- Citizens' ability to fetch information about themselves from helsenorge.no to import it into third party software solutions of their own choosing

The assignment is to describe the architecture and major components of the NIP, how it will function on the "outside" regarding third party integration, and on the "inside" regarding integration with helsenorge.no.

It is also essential that the solution adhere to Norwegian privacy regulation and information security. Its requirements for integration should also encourage privacy by design within third party solutions.

The prototype should make use of one or more use cases to demonstrate how interaction is performed, how privacy and security concerns are managed and how the end user experience will be.

Contact details:

Name: Helge T. Blindheim
Tlf: 466 75 321
Mobile: 466 75 321
Fax:
E-mail:— helgetb@helsedir.no

Name:
Tlf:
Mobile:
Fax:
E-mail:

Appendix B

How to build the project

Tomcat

Android app: The APK can be found here: https://github.com/Andersos/NIPEN/tree/master/nipen-heart-rate

Appendix C

Templates

C.1 Weekly status report

Weekly status report #X

Week NN

Dates 2013-MM-DD - 2013-MM-DD

TDT4290 Customer Driven Project - Group 17

1. Work done

1.2 Meetings

2. Plan for next week

3. Milestones

4. Problems

Activity	Planned	Actual
Studies	Number	Number
Project management	Number	Number
System developement	Number	Number
Application development	Number	Number
Database developement	Number	Number
Testing	Number	Number
Report	Number	Number

Table C.1: Activity chart

Time and Data	2013-MM-DD HH:MM
Place	Room
Attendees	Full name
Referent	Full name

Table C.2: Activity chart

C.2 Agenda for advisor meeting

Agenda for advisor meeting #X

2013-MM-DD

1. Approval of agenda
2. Approval of minutes from last advisor meeting
3. Comments to last weeks minutes
4. Approval of the weekly report

4.1 Work done

4.1.2 Meetings

4.2 Plan for next week

4.3 Milestones

4.4 Problems

5. Other
6. For next meeting

C.3 Minutes of advisor meeting

Advisor meeting X

2013-MM-DD

Time and Data 2013-MM-DD HH:MM

Place Room

Attendees Full name of attendees

Referent Full name

Time and Data	2013-MM-DD HH:MM
Place	Room
Attendees	Full name
Referent	Full name

Table C.3: Activity chart

1. Approval of agenda
2. Approval of minutes from last advisor meeting
3. Comments to last weeks minutes
4. Approval of the status report

4.1 Summerise status report

4.2 Work done in period

4.3 Work for next period

4.4 Problems in period

5. Other
6. For next meeting

C.4 Agenda for customer meeting

Agenda for customer meeting #X

2013-MM-DD

1. Approval of agenda
2. Approval of minutes from last customer meeting
3. Comments to last weeks minutes
4. Scenario
5. Decisions
6. Other
7. For next meeting

C.5 Minutes of customer meeting

Customer meeting X 2013-MM-DD

Time and Data 2013-MM-DD HH:MM Place ROOM Attendees Full name of attendees Referent Full name

1. Approval agenda 2. Approval minutes from last customer meeting 3. Comments to last weeks minutes 4. Scenario 5. Decisions 6. For next meeting

C.6 Agenda for internal meeting

C.7 Minutes of internal meeting

Appendix D

Advisor meeting documents

D.1 Weekly status report

D.1.1 Week 35

D.1.2 Week 36

D.1.3 Week 37

D.1.4 Week 38

D.1.5 Week 39

D.1.6 Week 40

D.1.7 Week 41

D.1.8 Week 42

D.1.9 Week 43

D.1.10 Week 44

D.1.11 Week 45

D.1.12 Week 46

D.1.13 Week 47

Appendix E

Customer meeting documents

E.1 Agenda for customer meeting

E.2 Minutes of customer meeting

Appendix F

Internal meeting notes

F.1 Agenda for internal meeting

F.2 Minutes of internal meeting