



T-404-LOKA, Final project

Heilsan mín í raun og VERU

Guðleif Harðardóttir

Ingvar Sigurðsson

Jóhannes Gunnar Heiðarsson

Jón Atli Baldvinsson

Kristjan Broder Lund

Spring 2014

BSc Computer Science

Instructor: Hlynur Sigurþórsson

Examiner: Stefán Freyr Stefánsson

Abstract

Vera is a web solution which encourages the public to take their health into their own hands. Vera allows the public user to see their medical records, such as Vaccinations, Doctors appointments, and drug prescriptions. This web solution is in development by the company TM Software.

The assignment which was at hand, was to allow users to track their general health values, such as height, weight, blood pressure, etc. The system allows the user to compare two health values together, like for example weight and blood pressure. Along with the health trackers, the system is equipped with a diary, which the user can write their reflections into. The solution was developed with the help of usability testing, to indicate a direction in user interface design. The usability testing lead to the idea of a food diary, which would allow users to record their eating habits. This feature would give the user an idea of how many calories and other essential nutrients they had consumed over time.

Contents

Table of Contents	2
List of Tables	4
List of Figures	5
1 Introduction	6
2 Background	8
2.1 Webmaster	8
2.2 Vera	8
2.3 Islandslykill	8
3 Vera Heilsan mín	9
3.1 Architecture	9
3.1.1 Heilsusaga	10
3.1.2 Webmaster	10
3.1.3 VMMAin	10
3.1.4 Heilsusaga Provider	10
3.1.5 Vera API	10
3.1.6 Matar API	10
3.1.7 Web Crawler	11
3.2 User interface	11
3.3 Features	11
3.4 Nutrition diary	12
4 Development tools	13
5 Development process	15
5.1 Methodology	15
5.1.1 Scrum Roles	15
5.2 Work process and planning	16
5.2.1 Sprint Organization	16
5.2.2 Overview of sprints	16

5.3	Work hours	17
5.3.1	Planned work hours per team member	17
5.3.2	Planned work hours on project	18
6	Progress	19
6.1	Burndown in points for the period	19
7	Usability testing	22
7.1	Summary	22
8	Future works	24
9	Conclusion	25
9.1	What went wrong	25
9.2	What went well	25
9.3	What did we learn	25
9.4	Final words	26

List of Tables

1	Overview of sprint organization goals.	16
2	Overview of sprints.	17
3	Work hours each team was willing to contribute per work day. . . .	18
4	Planned total work hours broken down by day, week and project scope.	18

List of Figures

1	Architecture	9
2	Color code for each sprint in graph	19
3	Total burndown in points: planned vs real for each sprint	20

1 Introduction

With the rapid development of information technology the public has been calling for a solution that would allow them to access their health care information and to have a tool to both record and analyse their information over time.

Vera, is a website, which is being developed by TM Software, which answers this call. The main objective with Vera is to give the public access to its health care records, such as prescriptions, and vaccinations to name a few. This report contains, describes a new extension for Vera, entitled Vera - Heilsan mín. This extension adds new functionality to Vera and allows users to keep records about their own health like blood pressure, weight and pulse.

Heilsan mín enables the public to keep track of the aforementioned metrics in one place and view them in accordance to recognized references and known health indicators. Users can view a metric within tables and in a graphical form, as well as print them out. In addition to that it was decided to implement a food diary which allows the user to keep track of their daily food intake.

At the time of writing, there is a gap on the Icelandic market for a comprehensive health tracker and food diary application where a user can keep detailed track of these items. We strongly believe that such a solution can be a step in encouraging people think about their health and food intake in a easy manner.

In the global scope there are applications that have achieved this. One such application is Health Vault by Microsoft. We used this application as a benchmark in our work and how we could improve and adopt many of its great features to the Icelandic market.

One of the design goals in this project was to create a solution that is both simple and user-friendly. Since we, the project group members, were also responsible for designing the user interface, then our understanding of the usability and ease of navigation for the website was likely to be totally different from users that have never used Vera. To get an understanding on how easy the website was to use, we decided to perform a study using usability testing. The purpose of this study was to identify areas of difficulty for users within the Vera – Heilsan mín website, but most of all to gauge the usability and ease of navigation. The information gathered was and will be used to suggest further improvements to the design.

The design of the user interface is certainly on the right path, and the consid-

ered goal of creating a simple and user-friendly website is within reach. Post-survey analysis showed that about 83% to 89% of the users thought that the website was user-friendly and at the same time simple and easy to understand. These results indicate that our website is accessible for a broad range of users with different backgrounds.

The work described in this report is a final project in the School of Computer Science at Reykjavík University. This project was conducted in participation with TM Software in the spring of 2014.

2 Background

In this section we will be explaining briefly some of the background work that has already been done on the project prior to our involvement.

2.1 Webmaster

Webmaster is a web content management system written and developed by TM Software. It enables us to provide any static content that is needed for the site, takes care of routing for us. It also has a user/admin interface to keep track of users allowed on the site.

2.2 Vera

Vera is a website that is being developed by TM Software. Vera enables the public to perform health service related tasks such as setting up an appointment with their doctor, renew and view prescriptions, and access their medical records.

2.3 Islandslykill

Islandslykill, or Islykill, is an on-line identification system, which allows users to access websites with their unique username and password. Any Icelander can gain access to their Islandslykill (username and password), through their Internet bank or through post. Our project uses this system as a method for logging in.

3 Vera Heilsan mín

The solution that we came up with to implement into the *Vera project* was primarily the project description that TM-Software handed us but with some deviations. We wanted to be able to allow the user to be in full control of what health values he/she was keeping track of and displaying it in the most simple way so the context would not get obscure. There was a lot of focus on the usability part and *Guðleif* took on that project creating the usability test and write the report, which is very extensive.

3.1 Architecture

The architecture is not that simple since there are many services that are communicating with the project.

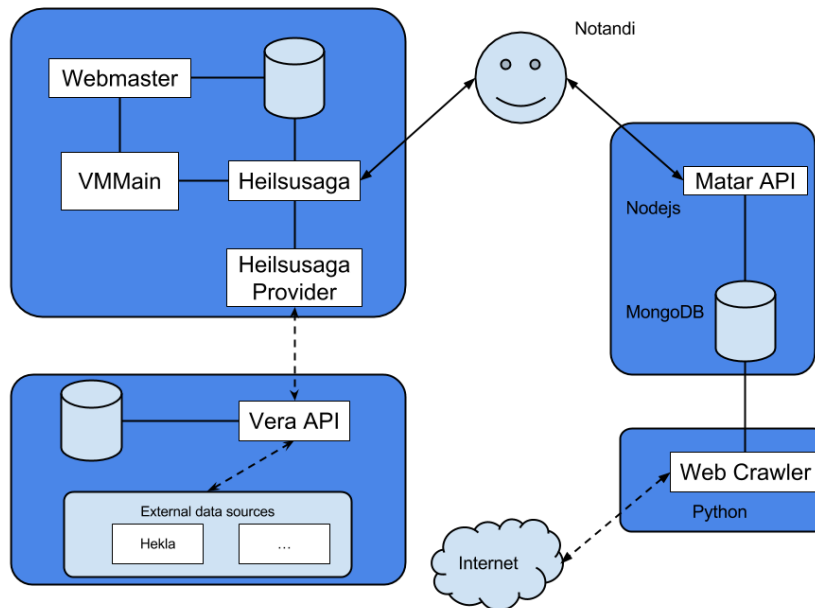


Figure 1: Architecture

3.1.1 Heilsusaga

The main project is hosted in this layer, it communicates with all of the other layers, both directly and indirectly. Here are all the HTML, CSS and JavaScript are served up to the client.

3.1.2 Webmaster

Webmaster takes care of routing and identifying if a user exist in the system. It hosts all the static content that is displayed on the web pages at Heilsusaga.

3.1.3 VMMAin

VMMAin is a helper utility library for Webmaster.

3.1.4 Heilsusaga Provider

This is where all of the services that *Heilsusaga* relies on. It's the gateway to *Vera API*.

3.1.5 Vera API

This is a fully fledged REST service that the *Heilsusaga provider* talks to, it's where all the user created content is stored. It also takes care of communicating with all the health services that TM-Software has been developing over the years. These health services consist of the following:

- Saga medical records
- Hekla healthcare network
- Medicor pharmacy prescriptions network

3.1.6 Matar API

Matar API is a solution that the team came up with so we could implement a nutrition diary into the application. Here we are able to provide a user with an extensive service to a food database that was created by a web crawler, which will be talked about in the next section.

3.1.7 Web Crawler

The web crawler takes care of gathering health related information from the Internet. For now we are only using the information that he gathered around food items such as nutritional data.

3.2 User interface

The user interface was co-developed with Atli Mar, the project manager of the main *Vera project*. There were meetings set up with him to go over what was done and if he had any input. Many good suggestions came out of those meetings. The Designer of the main Vera project, Ragnar Freyr, took our proposed user interface, and improved upon it, which is what you see in the final product.

3.3 Features

The features of the application are the following.

- Interactive graphs
- Weight tracker
- Height tracker
- Glucose tracker
- Blood pressure tracker
- Pulse tracker
- Mood tracker
- Temperature tracker
- Dashboard, overview of the sub components
- Personal diary
- Food diary

- Comparison of each health tracker
- Print functionality of all the health values
- PDF functionality of all the health values

3.4 Nutrition diary

The nutritional diary was kind of a side project that we always wanted to create with our application. At the beginning of the project we talked about having this as an A requirement but always thought it was too complicated to implement. Since we were such a big group we had to implement something more substantial than just what TM-Software had suggested in their proposal. The first thing we needed was data, a whole lot of data. We found out that *Matis* had developed a food database with detailed nutritional information about many food items in Iceland. We started developing a web crawler that was able to retrieve the data about each food item and store it to a Mongo database instance.

From the data that the web crawler was able to extract we created a REST service in Node.js that allowed us to search for a particular item in the database. This was a huge step for us in the project development because it meant that we were able to implement the feature that we always thought was too hard to do. We set up some more endpoints in the REST service so we could keep track of users, what that user had eaten, the amount of which he/she had consumed of a particular food item. There was also set up endpoints to be able to update and delete a particular item that a user had consumed.

There was maybe only one problem that we did not know of and that was the cross-origin resource sharing, which means that a server will not provide access unless you explicitly tell him to allow requests from other servers, this was solved quite easily with a npm module called cors.

4 Development tools

During the development many tools were used, following is a list of tools and an explanation of what they were used for.

- Microsoft Visual Studio 2012 IDE
- ASP.NET Framework
- SQL Server from Microsoft
- Nodejs Express framework for the food Api
- MongoDB for database in the food Api
- Scrapy framework, python framework for web crawling
- Webmaster provided by TM-Software for backend
- Html, CSS and Javascript for frontend
- Git for version control
- Jira for project management.
- Google Drive/Confluence for documentation

For a communication solution the team created a Facebook group, which was used as a forum for suggestions, problems and general communication.

As for the choice of an IDE the team decided upon **Microsoft Visual Studio 2012** because it has a built-in support for C# **ASP.NET**, offers connection to SQL server databases (which was used in this project), and it is one of the best environment when developing for Windows applications. TM Software uses Visual Studios for their development projects in ASP.NET. We were also using a content management system called *WebMaster* which is directly from TM-Software.

We used the **Express framework** to create the REST service for the food api. It also uses two modules which are **Mongoose** module for connecting to the MongoDB instance, and **Cors** module which helps us with the cross-origin request sharing.

The **Scrapy framework** for scraping and crawling the websites that we wanted to extract the data which we stored into the MongoDB instance, the data is then

serviced by the food api. There was also used PyMongo module for connecting with the database instance.

We used **MongoDB** as a database for the food api. This is a NO-SQL database and has high performance features as being highly customizable. Helped us with not having to think about how the data is stored into the database after the web crawler had finished his job.

Git was used for version control of source code, and Stash server provided by TM Software was used to store source code. Git was also used for version control of Latex files.

Jira was used for project management. Access to it was provided by TM Software. Jira is a bug tracking, issues tracking, and project management system developed by Atlassian Software Systems. Jira enables you to create collaboration boards that simply visualize the different stages of progress for a project.

Google Drive was used mainly for the Excel documents concerning the product backlog, reports, etc. As well a forum to throw in ideas of reports, and other interesting files that could be used in regards to planning the project.

5 Development process

This section explains what our development process, who played what roles, what management system was chosen why. We will go through the roles of each person on the project, our initial planning of the project and what each person was estimating to work on the project each day.

5.1 Methodology

We decided to use Agile-Scrum methodology for project management. The methodology implemented is according to Agile-Scrum which is used by TM Software, all team members know something about Scrum so we felt comfortable with using it. Scrum gave us the flexibility that we want like iterative process of always estimating before and after each sprint, having the capability of adding a new feature request on the fly. Scrum also provides the *big picture* concept of being able to look at the product backlog and evaluating the whole project and also the benefits of taking on a small task and focusing just on that by a developer.

5.1.1 Scrum Roles

Product owner:

Atli Mar Gunnarsson

Scrum master:

Guðleif Harðardóttir

Team members:

Guðleif Harðardóttir

Ingvar Sigurðsson

Jóhannes Gunnar Heiðarsson

Jón Atli Baldvinsson

Kristjan Broder Lund

5.2 Work process and planning

Here we will go briefly over the work process and planning the project.

5.2.1 Sprint Organization

At the beginning of the period the team decided upon a few sprint organization goals. [Table 1](#) lists up these goals.

Organization goals	Objective	Notes
Sprint length	2 weeks	
Scrum meetings	1 - 2 hours	
Retrospective	1 hour	
Daily Standup	15 minutes per day	Status meeting every Thursday at TM Software, otherwise Facebook.
Sprint naming theme	Star Wars	

Table 1: Overview of sprint organization goals.

5.2.2 Overview of sprints

At the beginning the period was split up into sprints, and they were in total 9. The first sprint was mainly used to setup environments, IDE's, version control, and so forth. The last sprint was used for bug fixes, documentation and reports. [Table 2](#) lists up the sprints.

Sprints	Sprint name	Period	Days	Comments
0	Sprint 0	16 Jan 2014 - 2 Feb 2014	18	Setup environment, IDE's, version control etc.
1	Jar Jar Binks	3 Feb 2014 - 9 Feb 2014	7	First sprint.
2	Luke Skywalker	10 Feb 2014 - 23 Feb 2014	14	Start programming
3	Princess Leia	24 Feb 2014 - 9 Mar 2014	14	TBA.
4	Yoda	10 Mar 2014 - 23 Mar 2014	14	TBA.
5	Obi-Wan Kenobi	24 Mar 2014 - 6 Apr 2014	14	TBA.
6	Darth Vader	7 Apr 2014 - 20 Apr 2014	14	Exams, Easter.
7	Return of the Jedi	21 Apr 2014 - 4 May 2014	14	All major work should be done.
8	Death Star	5 May 2014 - 14 May 2014	11	Code Freeze. Final touch of reports.

Table 2: Overview of sprints.

Each story within each sprint was split up into sub-tasks. A story was done when all sub-tasks had been resolved and it had passed acceptance testing. Programming for each sub-task within a story was completed when programmer thought he had finished it and other team member had completed reviewing the source code.

5.3 Work hours

Team members had a Excel file in Google Drive for logging work hours.

5.3.1 Planned work hours per team member

Each team member stated at the beginning of the period on how much they were willing to contribute in work hours per work day. [Table 3](#) lists up planned team member work hour contribution.

Team member	Hours
Guðleif Harðardóttir	4 hours
Ingvar Sigurðsson	4 hours
Jóhannes Gunnar Heiðarsson	4 hours
Jón Atli Baldvinsson	3 hours
Kristjan Broder Lund	3 hours

Table 3: Work hours each team was willing to contribute per work day.

5.3.2 Planned work hours on project

By using the contributory work hours ([Table 3](#)) the team estimated that total planned work hours per day was 18 hours, for a week (assuming 5 day work week) 90 hours and for the period 1530 hours. See [Table 4](#).

Project plan	Hours
Total work hours per workday	18 hours
Total work hours per week	90 hours
Total hours on project	1530 hours

Table 4: Planned total work hours broken down by day, week and project scope.

6 Progress

This section details the total burndown in points over the period in Spring 2014. As mentioned in subsection [5.2.2 Overview of sprints](#) the period was split up in 9 sprints. Detailed progress report can be found on CD-disc. The first sprint and the last sprint Death Star are not included in the burn-down graph since no story points were estimated for those sprints.

6.1 Burndown in points for the period

In the following graph you can see the total burndown in points, estimated vs real for the period. To get a sense of the graph take a look at the coloring index.

[Figure 2](#) shows what the colors mean in the graph.


Colors	Sprints
	Jar Jar Binks
	Luke Skywalker
	Princess Leia
	Yoda
	Obi-Wan Kenobi
	Darth Vader
	Return of the Jedi

Figure 2: Color code for each sprint in graph

[Figure 3](#) shows burn-down in points over the period.

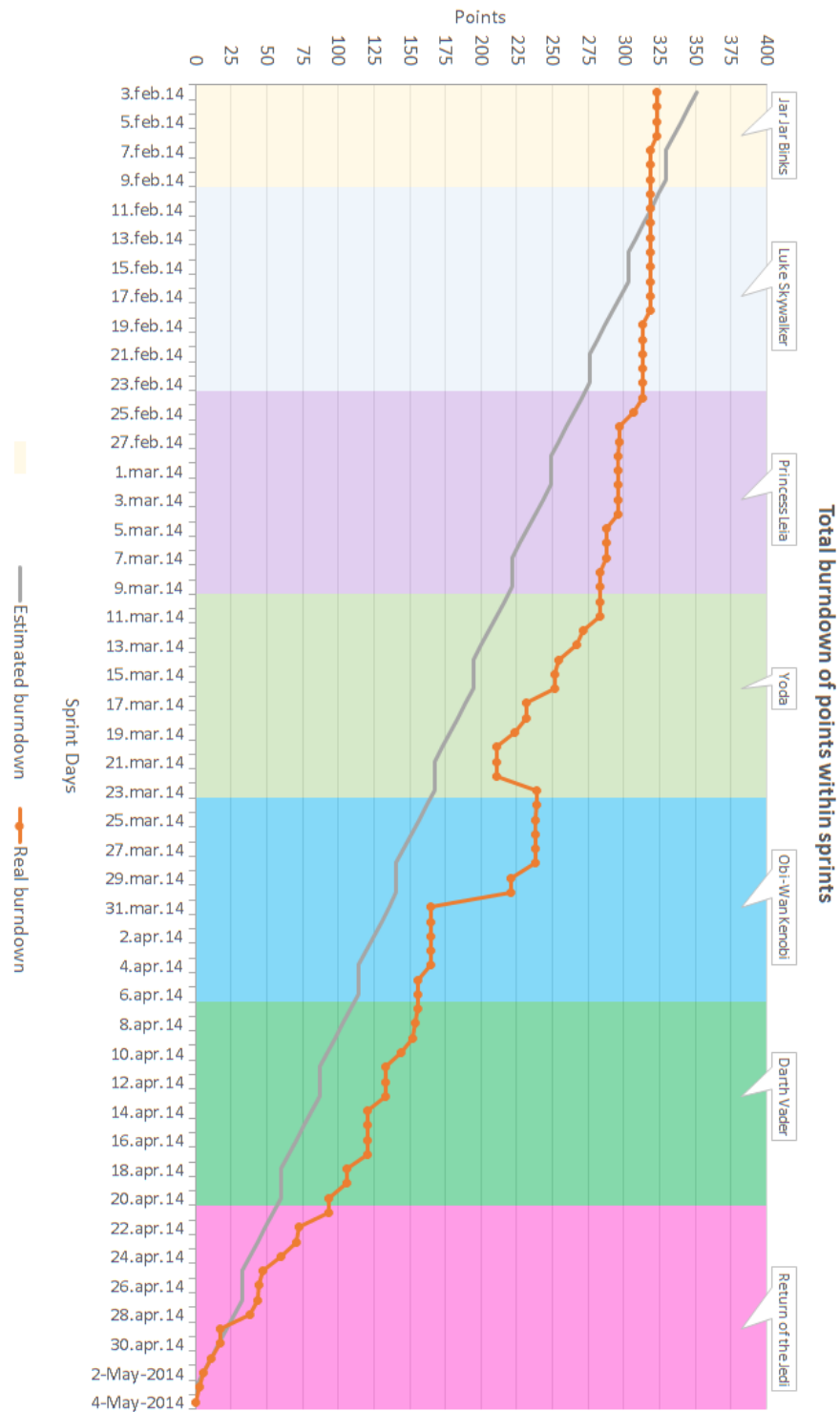


Figure 3: Total burndown in points: planned vs real for each sprint

The x-axis shows days, the y-axis shows story points. Each sprint is color coded in the graph. The orange line shows the real burndown in points and the gray line shows the estimated burndown in points.

At the end of sprint Yoda, the project scope changed by 28 story points, as seen by the spike in the graph on 23. March. That is when the stories for the implementation of the food diary were added to the project.

7 Usability testing

This section details a general summary about usability testing that was conducted for the final project. For a more detailed report read the Usability report that can be found on the CD-disc.

7.1 Summary

One of the goals of the final project was to create a website that was simple and user-friendly. As programmers of the website, the group's understanding of the usability and ease of navigation for the website can be totally different from users who have never used Vera.

During our implementation of user test sessions, we used the following three methods. "Think-Aloud", which is widely used in usability testing, "Arts Across the Curriculum" functionality, which is a method for a facilitator on how to conduct the usability test sessions, and finally "Convenience sampling", which is the method used for recruiting participants.

User testing was conducted over the period March 24 to April 13, 2014. Ten participants were recruited who had never used Vera. Entrance and exit surveys were administered before and after each test session. During the 30 – 40 minute test sessions, each participant was given a set of tasks to complete, which were read aloud to them by the facilitator. Key functionalities, such as insert, edit and delete were selected for the usability testing, for both diary entries and health values and the users were also asked to compare health values. A health value is for example blood pressure.

We assumed that the users should take no longer than 40 seconds to 1 minute to solve each task, and that each key functionality would not take more than 5 clicks to complete. We also assumed that 90% of the users would think that the website is user-friendly and easy to use. We also assumed that users would complete each task without major problems.

It took a user over 1 minute to complete a task on average, so our hypothesis

failed. In hind-sight it would have been better to do a separate task for the time trials, because the users were encouraged to think-aloud while performing a task and that prolonged the performance time. If there was enough time, a separate time trial could be performed for those same tasks.

It took a user on average 3 - 5 clicks to complete a key functionality such as insert, so our hypothesis about a 5-click rule was correct. The majority of the users completed a task without major problems.

The post-test survey analysis showed:

- About 84% of the users thought that the website was user-friendly.
- About 83% of the users thought that the mobile interface was user-friendly.
- About 83% of the users thought that the website was easy to use.
- About 84% of the users thought that the mobile interface was easy to use.
- About 89% of the users thought that the UI simple and easy to use at first sight.
- About 85% of the users thought that the mobile interface was simple and easy to use at first sight.

The target goal of 90% was not reached, but came very close. So we can say with some certainty that 83%-89% of the users think that our website is user-friendly, simple to use and easy to understand.

User comments and observations collected during the UI test sessions shed some light on what works and what does not work in the UI, and were used to improve the design on some parts of Vera - Heilsan mín.

8 Future works

The goal was to create a prototype for TM-Software based on their idea. The prototype will hopefully provide a platform to extend the idea and create some more features that we have thought about to put on top of the existing ones.

Setting up a goal system for the user, that is we would like to see that a user can set a goal for a health value that he/she is interested in. Such as setting a goal for weight, this is probably the most thought about health value in modern times so you would want to set a goal for that, but this could apply to blood pressure, glucose levels etc.

Creating a workout tracker that would be connected to the user smart phones, track their workout and upload the information to our server. This would be in the same orientation as the RunKeeper app, would ease the data creation.

Create more specialized trackers for persons with more severe health conditions such as diabetes, cancer, etc. Set it up such that it would notify the user if he/she are getting worse from the data that they provide.

Create a diet plan system for the food diary. It would create a diet plan for the user and he/she would check of the items that he/she is recommended to eat over the day. This could help people which would want to get into better eating habits.

Recommendation system for the food diary. We would be collecting some statistics about users eating habits and out from that information we could either suggest him/her with food items that he could be eating based on what goals he/she has set for him self.

Calorie goal system in the food diary. User could set a goal of eating a specific amount of calories over a day from which we could deliver information on how he could be achieving his/her dream.

These are some of the ideas that we have thought of for the application in the future context. Probably the biggest future work would be to create a smart phone app for all the major platforms. That would clearly be a good thing since it could allow the user to more easily collect data about their health activities and health information.

9 Conclusion

In conclusion the team is satisfied with the developed project, as well as the chance we got to work with a well developed company.

9.1 What went wrong

During the first sprint the team did not utilize the stand up enough. The team had quite the difficulty in creating well defined stories, due to the lack of experience of story creating, the first sprints got affected by this and a lot of stories flowed over. As well as badly defined stories in the first part of development, the project itself was rather directionless. Due to the team size the project itself seemed not demanding enough, therefore the food diary was created.

9.2 What went well

The team worked very well together, and when a problem occurred the team worked together to find a solution. Despite the size of the team, the team managed to keep good communication between all its members. As well as good communication, the team spirit was not affected by the difference in hours, logged compared to each other. The project got a solid definition of direction after, the team learned to define stories accordingly. Usability testing went well, which was used as an input on the user interface design.

9.3 What did we learn

As the team often had either stories flowing over or not enough stories for each sprint, the team learned that it was important to know the capabilities of your team, when working in the scrum methodology. It was also a learning curve to create well defined stories and managing the product backlog for this project. The stories were better defined at later stages in the project, and in accordance to that the burndown picked up (see [Figure 3](#)). While working on the project, the team got good practise in HTML, CSS , ASP.NET, Javascript and database connections. Above all, the best that we got, was an invaluable real life working

experience, on how it is to work in a team of coders and, even though it can get rough at times, you can always count on your team mates.

9.4 Final words

The team would like to thank TM Software and Reykjavík University for the opportunity this project provided in getting a real life working experience. We would also like to thank the more experienced developers at TM Software for any assistance they provided us, they were always willing to help or answer any questions if we had any.