Gas Teddy Bear Bobo
Final report

Fall 2013

Arnar Þór Úlfarsson
Bjarki Dagsson
Björn Heiðar Rúnarsson
BSc in Computer Science

Examiner: Torfi H. Leifsson
Instructor: Hlynur Sigurþórsson

Contents

1. Abstract

This report details a final project in Reykjavik University school of Computer Science conducted in the fall of 2013. The goal of this project was to create a mobile web application named Gas Teddy Bear Bobo, often referred to as Bobo. Bobo is a mobile application which allows users to see real time gas prices, offers that the gas companies have active and the user's fuel consumption. Additionally, Bobo lists and shows gas stations in Iceland as well as providing driving route to individual stations with respect to users position. Detailed overview of the web services Bobo connects to, the methodology used in designing and implementing the application and how the project progressed through the 15 weeks of the project. Finally it gives the results of the simplicity tests and ideas on future development of the application.

2. Introduction

Are you located in an unfamiliar part of Iceland and wondering where the next gas station is situated? Do you follow the price closely and utilize offers as much as you can? Perhaps keep close eyes on your fuel spending?

Bobo is a mobile application that works on most common mobile operating systems, that provides you with the information you need to be up to speed when it comes to fuel purchases. The prices and offers in real time, keep track of your refills and find gas stations in your vicinity as well as seeing the driving route there.

Maybe you represent a gas company and are trying to inform your customers of your great prices and offers? Log into the Entrio Business web service, make the offer and publish it to all Bobo user who will have access to the details immediately. The Entrio Business web service gives you easy access to numerous services to get your offers known, without needing to go through a middleman.

Bobo is a simple yet functional application, as shown by our user testing: on a scale from 1 to 10, Bobo scored on average 8.89 on simplicity and 9 on usability.

This report will take you through the various parts of this final project, Gas Teddy Bear Bobo. In chapter 3 we explain Bobo, the web services it relies on and the architecture as a whole. In chapter 4 we cover the work methodology, tools used and the risk analysis. In chapter 5 we cover sprints and their progress. In chapter 6 we cover simplicity tests and their results and in chapter 7 we cover future ideas for Bobo and the conclusion.

3. Gas Teddy Bear Bobo

3.1 Introduction

Bobo is a mobile application. With it, you can see the cheapest gas prices. You can easily find the location of the nearest gas station from your own location. The application lists every gas station in the country and you can get directions to each of them. Application users can see offers from gas companies listed and they can also see what benefits they get by acquiring gas company's keys and cards. In addition, you can fill in information about your vehicle's consumption.

3.2 Architecture

The application is not the only thing to this project. It needs web services to provide the user with the information he needs. We now take a back-to-front look at the project as a whole.

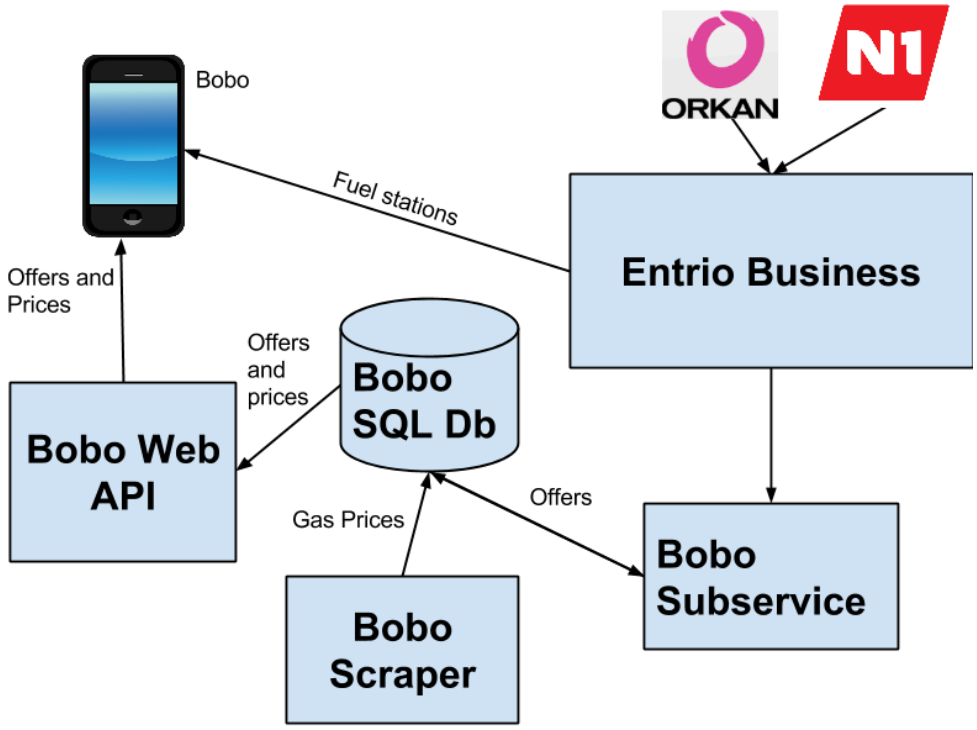

Figure 1, Bobo architecture overview

In figure 1 we see the Bobo architecture overview. We start by looking in the top right part where in our hypothetical scenario created to describe the architecture where two companies (Orkan and N1) are used. They are the company representatives who log onto Entrio Business, which is the entry point for the company representative. The representative then selects the Bobo subservice button, bringing him into the Bobo sub service which has the same looks as the rest of the web service and the representative should therefore not feel like he had started talking to a different service. Inside the Bobo subservice he can add, edit and delete offers as needed and all the changes will be made in the remote Bobo SQL Db (Database).

The Bobo Scraper is a web scraper which, at the time of writing, runs every 5 minutes and fetches the most recent fuel prices of the site http://fib.is/, with permission from the website owner. It writes those prices into the Bobo SQL Db.

When the Bobo application (phone) queries the Bobo Web API for data, the Web API gets the data from the Bobo SQL Db and serves it to the application. The application also queries Entrio Business for the gas stations data, because the information is stored in the central service (Entrio Business).

3.3 Application

Although Bobo installs and feels like a native application on mobile devices it is, technically speaking, a web site run by the device's browser. This gives the application cross-platform capabilities because all devices have browsers that can render HTML, which is the cornerstone of mobile web applications along with JavaScript and CSS. This comes at a price, the application is restricted in the same way as web pages. This means, among other things, that it has limited storage available to it and restricted access to the device's hardware (e.g. camera, notifications) and file system.

The application uses the Sencha Touch 2[1] framework, which is a high-performance HTML5 mobile application framework. It is designed to make mobile web applications quickly and easily. Along with Phonegap Build[2], which is a platform for building mobile web application, Sencha Touch can actually shrink the gap between mobile web applications and native applications by allowing access to native device function.

3.3.1 Use cases

Bobo is an application that allows the user to find cheapest gas prices, nearest gas station and to see offers from gas companies. The user can find driving routes to stations. Here below we take a look at few examples.

[1] http://www.sencha.com/products/touch/
[2] https://build.phonegap.com/

95 okt gas price

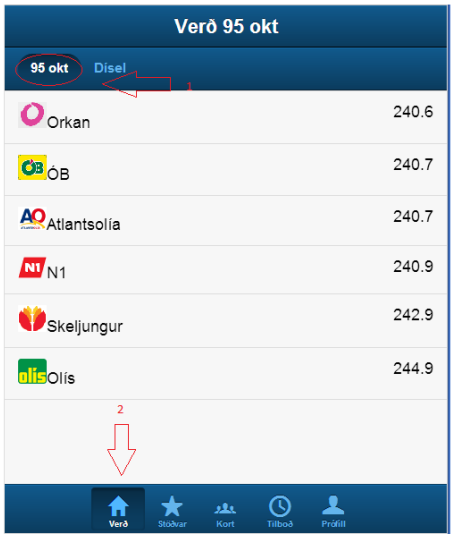

Figure 2, 95 okt gas prices

In figure 2, we see the starting page in the application, there we can see price for 95 okt gas in sorted order where cheapest is at top. In the figure, the red arrow marked with 1 is pointing to the active tab (95 okt) and the red arrow marked 2 is pointing to the active sheet. „Verð“ (price). The user can also click on the tab „Dísel“, (diesel) to see the list for gas type diesel.

Nearest gas station

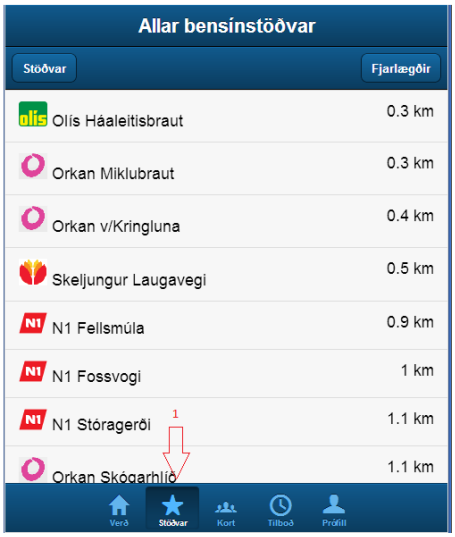

Figure 3, stations

In figure 3, we can see all stations in sorted order from the user's location where the top station in the list is the closest one from the user's location and so on. In the figure, the red arrow marked with 1 is the active sheet „Stöðvar“, (stations). The user can push

„Stöðvar“, (stations) button in the top left corner to check/uncheck gas companies he wants to view. The button in the top right corner „Fjarlægðir“ (distances) the user can use to only see stations within a certain distance from his location in airline.

Offers

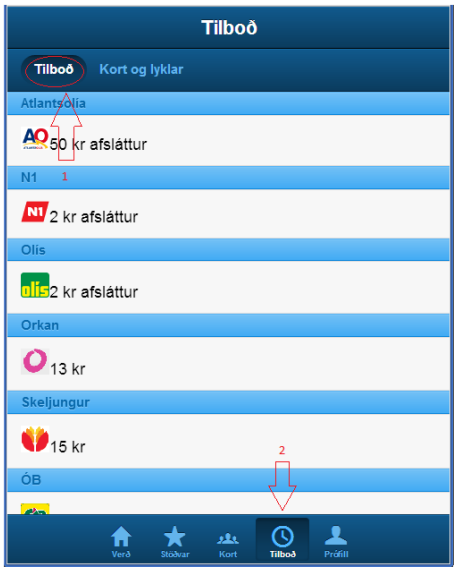

Figure 4, offers

In figure 4, we can see offers from gas companies. Offers from companies are listed together. The red arrow marked with 1 is pointing to the active tab „Tilboð“, (offers) and the red arrow marked with 2 is pointing to the active sheet „Tilboð“ (offers). The user can also click on the tab „Kort og lyklar“, (cards and keys) to see what benefits they gain from acquiring cards and keys from the gas companies

Driving distance

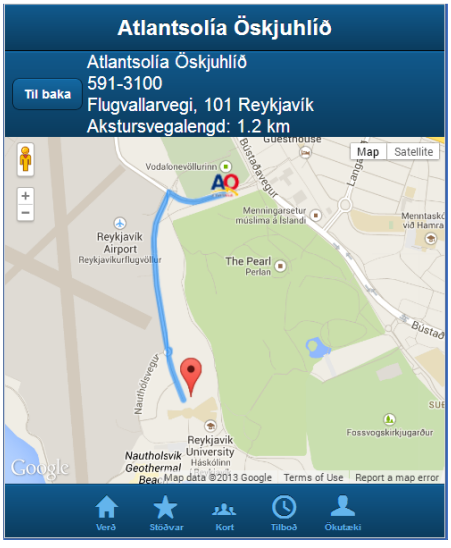

Figure 5, driving distance

In figure 5, we can see the driving route from the current user's location here in Reykjavik University to the station „Atlantsolía Öskjuhlíð". We can see information about the station in question and driving distance in kilometers.

3.4 Services

Bobo uses two web services, one to provide gas prices and offers to the application and one to provide information about gas stations and manage the offers. Both services are implemented using C# ASP.NET MVC 4. Additionally there is a web scraper that retrieves the current gas prices for all the gas companies every 5 minutes (from http://fib.is/).

The service that provides the data on gas prices and offers is an ASP.NET MVC4 Web API that responds to GET requests with data on JSON format. There are currently 3 functions that can be called: GetGasPrices, GetOffers and GetOffersToMember, where OffersToMember are the benefits you can attain with discount cards and keys. This service connects to a SQL database remotely.

The service for managing offers is a part of a larger web service, Entrio Business. Entrio Business is a center point where companies can manage their basic information (e.g. branches) and then connect to sub services, one of which is for Bobo which was a part of our project. In these sub services company's representatives can manage information specific to that sub service's endpoint (application), in this case it is the offers that companies are offering application users and what benefits the user gains from acquiring cards and keys from gas companies. The idea is that the user never notices the change from the Entrio Business service to a sub service. The Bobo sub service connects to the same remote database as the web API so that changes made, even when running the Entrio Business locally, will affect the remote database and therefore the data the application gets when it connects to its web services. The information about branches is uniform for all sub services of Entrio Business and outside the scope of this project.

Figure 6, gas company employer has logged in.

In figure 6, we can see the view when gas company employer has logged into Entrio Business, there he can see the sub services Entrio has linked to his account.

Figure 7, Add, edit, delete offer view for employer

In figure 7, we can see the view when an employee from gas company N1 has logged in and clicked Bobo sub service that we saw in figure 6. Here the employee can see offers from N1. He can add, edit or delete offers here, which will display accordingly in the Bobo application.

4. Project Plan

In the beginning of the project, all we knew was that we were building this mobile application Bobo. We needed to decide what tools we were going to use and what methodology we would follow.

4.1 Methodology

The team decided on using the agile methodology Scrum because all members of the team had worked with it before and we knew we had to be well organized. The team would meet regularly and spend a lot of time together at the same place. Each day we held a daily stand-up meeting, where team members shared their progress from the day before, what they would do today and if there was anything standing in their way, so they could not continue their work. We used a virtual Scrum board located on http://trello.com.

4.1.1 Scrum

One of the main benefits of using Scrum is that the methodology allows the team to deliver as much value as possible over short time period.

The team made a product backlog which included all requirements the application should/could have. The backlog is listed with A, B or C kind of requirements, where A is an essential part of the application, B is good to have and C is nice to have. Product owner listed these requirements. The team along with product owner evaluated the user stories using planning poker, where each individual gave stories story points and the team finds the appropriate story point value for each story.

Next the team makes a sprint backlog which includes user stories that will be worked on during each sprint. Then the stories are broken down into smaller tasks.

Daily scrum meetings were held where team members talked about what they were doing last day and what they are going to do today.

Burn down charts were updated daily so team could be up-to-date on project status.

After each sprint the team holds a retrospective meeting where they discuss what went well and what could be done better in coming sprints.

4.1.2 Why scrum

The main reason why scrum was chosen is because it is very flexible, and we knew we needed flexibility as tasks or story points can change during the project. Stories and requirements can also be added to the product backlog at any time. We can react on changes instead of just following a certain schedule.

4.1.3 Scrum roles

In each Scrum team there is a Scrum master who is sort of a project manager, responsible for the team to have harmonious work environment. He also removes any impediments to progress. The Scrum master is the one who communicates the most with the product owner.

Product owner:	Vignir Örn Guðmundsson
Scrum master:	Arnar Þór Úlfarsson
Team members:	Arnar Þór Úlfarsson
	Bjarki Dagsson
	Björn Rúnarsson

4.2 Facilities

As mentioned before, the project was done in cooperation with Entrio ehf. At the moment, Entrio does not have any actual workplace. We were assigned work facilities in „Úranus“, which is located at Reykjavík University.

4.3 Tools

When deciding what tools we were going to use we knew that we wanted the application to be „cross platform“, (useable on all operating systems).

4.3.1 Sencha Touch

Having little or near no experience in building a mobile application, we were guided by our Entrio correspondent to Sencha Touch. This is a high-performance HTML5 mobile application framework. We took some time to look into Sencha Touch to see what it was about. After reconvening we were all on board with Sencha Touch.

For more details on Sencha Touch, see the developers guide and http://docs.sencha.com/touch/2.3.0/

4.3.2 Github

Github is a social networking for programmers. All of our code is hosted there. We decided to use Github because everyone in our team was familiar with it and all team members agreed on that Github is the best version control out there. When building an application like this each team member can copy the master branch when trying new features and when they are finished they are merged to the master branch.

4.3.3 Phonegap Build

Phonegap Build is the tool that brought our application to life. It takes our code from Github and builds it for the most popular operating systems. Building the application with Phonegap Build we were able to download the application to our phones by scanning a QR code. That made it easy for us to test the application on different phones.

4.4 Documentation

The team used Google Drive and Dropbox to manage most of the documentation and set up a diary on Google docs to keep track of time spent for each team member. That and the burn down chart made it easy for the team to calculate the teams velocity.

4.5 Estimated work hours

At the beginning of the project we decided to plan our work rate for each week. The first six weeks the plan was that each member of the team would deliver 20 hours of work per week, resulting in a total of 360 hours. The remaining weeks we planned that each member would add 5 hours of work per week. That gives us a total of 1035 hours of work estimated for the project.

4.6 Risk Analysis

At the start of this project the developing team made a risk analysis to identify and be ready with solutions to possible problems.

Table 1 below contains the risks the team identified and their potential effects on the project. The severity of these effects was valued on the scale 1-10, where 1 is a very minor effect on the project and 10 is catastrophic, to the point of project failure. Odds are given on the scale of 1-6 where:

- 1 = 0-10% odds. This is very unlikely to happen.
- 2 = 11-30% odds. This is unlikely to happen.
- 3 = 31-50% odds. There is a chance this might happen
- 4 = 51-70% odds. More likely to happen than not.
- 5 = 71-90% odds. This is likely to happen.
- 6 = 91-100% odds. This will almost certainly happen.

The risk factor is calculated by multiplying the severity score with the odds score.

#	Risk	Effect	Solution	Guarantor	Severity	Odds	Risk factor
1	Back end not ready at the end of project.	Offer part not ready, offers in application missing.	Get access to Entrio Business ASAP.	Bjarki	10	2	**20**
2	We don't get access to gas prices from gas companies.	More time consuming and more fragile.	Scrape the gas prices from a reliable website.	Björn	6	3	**18**
3	Bobo not ready before deadline.	Delays the release of Bobo.	Entrio ehf. finish Bobo.	Arnar	9	2	**18**
4	Gas companies have no interest in using Bobo.	Crucial that the gas companies can add, edit and delete their offers. So that users can view offers in the application.	Simplicity tests give feedback from users on usability Making the application more usable increases interest from gas companies.	Björn	6	3	**18**
5	Bobo does not work cross-platform	Fewer people can use the application.	Test Bobo on devices we have access to. Find tools to test on other systems. Focus on IOS and Android.	Arnar	5	3	**15**
6	Bobo does not get good receptions.	The project could be a failure.	Make Bobo as simple as possible	Arnar Bjarki Björn	4	3	**12**
7	Lack of knowledge on development tools.	Delays work on Bobo.	Read documentation on development	Arnar Bjarki Björn	3	4	**12**

			tools.				
8	Workload in other courses.	Delays work on Bobo.	Organize and make time for Bobo.	Arnar Bjarki Björn	3	3	**9**
9	Computer failure	Delays work on Bobo, data and code can be lost.	Use version control, Github, Google docs, Dropbox.	Arnar Bjarki Björn	2	2	**4**

Table 1, risk analysis

The biggest risk in the development, in the team's opinion, was that Entrio would not complete their web service. In that case the team would not be able to start work on their part of the service (Risk #1) and therefore have to rely on either hardcoded data or setting up our own web services. The web services were a part of the project, making it impossible to properly finish the project without it. To avoid this, the team "had to twist some arms" to put pressure on Entrio to get the web service part done and additionally made a web service with hardcoded data. Entrio delivered in a timely fashion and the risk was eliminated.

The team did not get access to real time gas prices from the gas companies (Risk #2). This was solved by making a web scraper which gets the prices every five minutes of the site http://fib.is/.

While Bobo could be improved upon almost endlessly the team made sure to know when the application could be considered finished (Risk #3) and concentrated on finishing the tasks needed to get it done.

5. Project progress

5.1 Introduction

During the development of the application the team went through 7 sprints, all sprints were 2 weeks long. In this chapter we talk about what was done in each sprint and how it went. Sprint 0, 1 and 2 we estimated 20 work hours per member per week but in sprints 3, 4, 5 and 6 we estimated 25 work hours per member per week.

5.2 Sprint 0

4th of September – 17th of September

In this sprint our main focus was on planning the project. Meetings, installing tools and devices needed. We made a vision statement for the project and did a lot of paperwork, risk analysis, work schedule and work planning.

The team had not been given a work station which downgraded team efforts. One member of the team was abroad for the majority of this sprint.

No story points were put in this sprint.

5.2.1 Worked hours

Total hours planned for each member of the team for sprint 0 was 40 hours or total of 120 hours for the team. Group members worked 155.5 hours in this sprint as we can see in figure 8 below which is about 30% over the planned schedule.

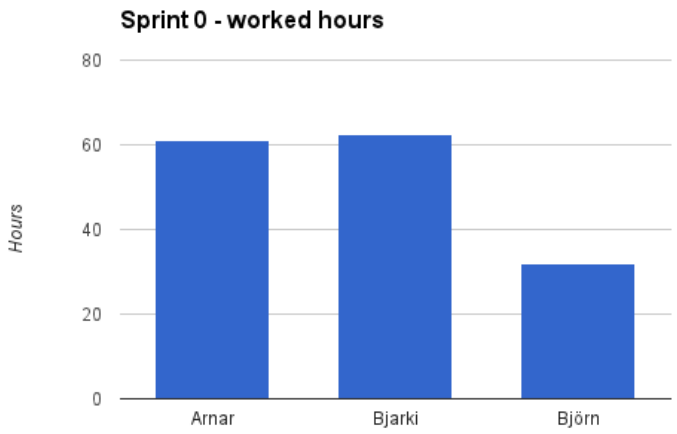

Figure 8, worked hours in sprint 0

5.3 Sprint 1

18th of September – 1st of October

In sprint 1 we started programming. A lot of time went in getting to know the tools we were working with better, those tools were mainly Sencha touch but also some time for Github and PhoneGap. When that was done we started on the stories in the sprint.

We decided to make our own web service with hard coded data because Entrio‘s web service was not ready.

We got assigned a work facility in „Úranus“, which helped us.

Our main focus in this sprint was to connect web service with our application to get real time fuel price. User story 1 which we had split into two tasks was changed so it was one story and one task and we increased the number of hours needed to complete it due to more time in researching how to get this done.

5.3.1 Sprint burn down chart

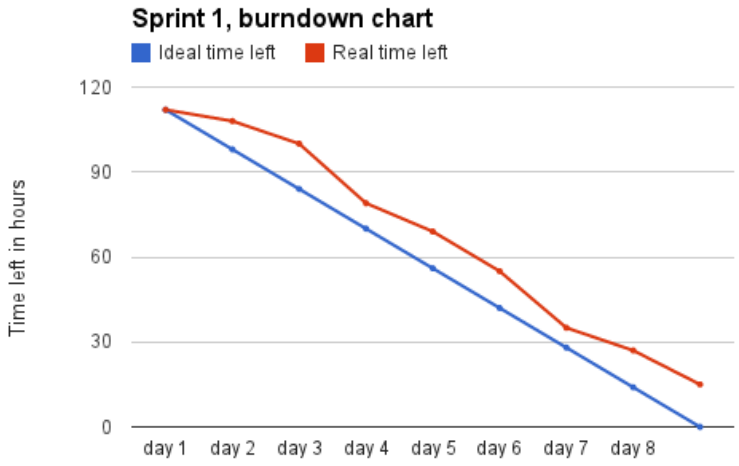

Figure 9, sprint 1 burn down chart

In sprint 1 the team completed 23 story points.

5.3.2 Worked hours

Total hours planned for each member of the team for sprint 1 was 40 hours or total of 120 hours for the team. Group members worked 131.75 hours in this sprint as we can see in figure 10 below which is about 10% over the planned schedule.

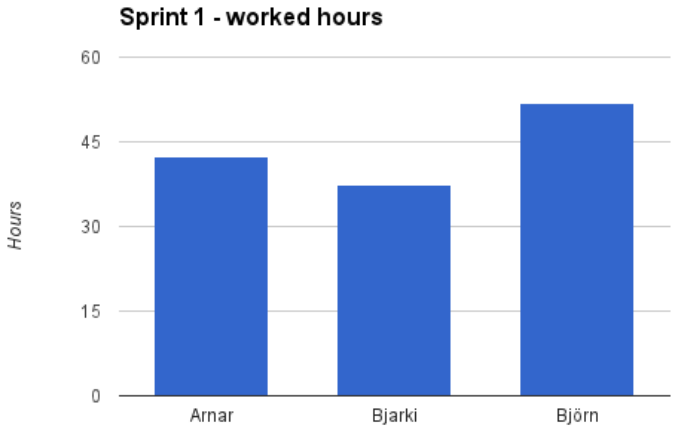

Figure 10, worked hours in sprint 1

5.4 Sprint 2

2nd of October – 15th of October

Our main focus in sprint 2 was to finish the biggest story, which was 20 story points. We were able to finish the story. We also took from sprint 1 design and look parts. We had a design session with the product owner to decide on final look of the application. We also went into some research work of gathering information about latitude and longitude of gas stations in Iceland. Our big story in this sprint was regarding the map part of our application.

5.4.1 Sprint burn down chart

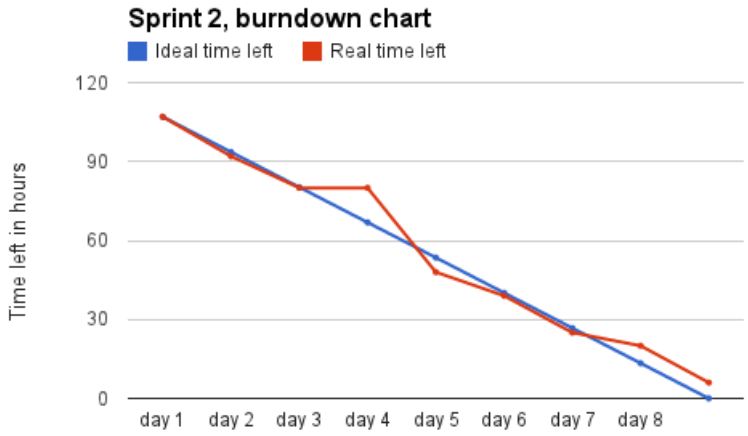

Figure 11, sprint 2 burn down chart

In sprint 2 the team completed 25 story points.

5.4.2 Worked hours

Total hours planned for each member of the team for sprint 2 was 40 hours or total of 120 hours for the team. Group members worked 112.75 hours in this sprint as we can see in figure 12 below which is about 6% under the planned schedule.

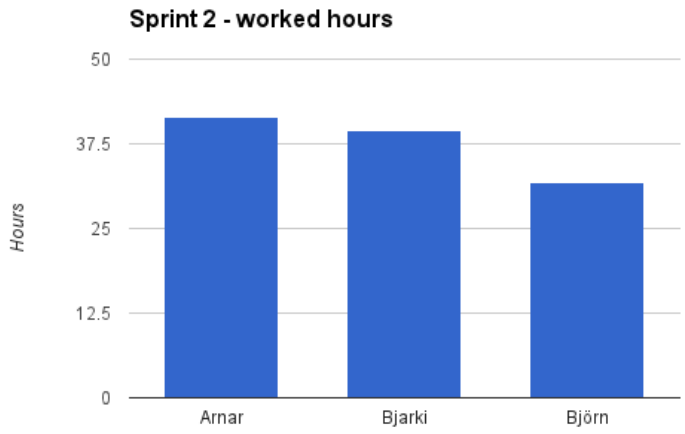

Figure 12, worked hours in sprint 2

5.5 Sprint 3

16th of October – 29th of October

In sprint 3 we got stories added to our backlog, those stories were regarding the web service and the backend regarding how gas company employees can add offers straight to application users.

One story was about scraping the website www.fib.is to get real time gas prices. In order to complete these stories, much research work was needed. We started by setting up Microsoft Visual Studio 2010 with MVC4. No member of our team had scraped a website before and some thoughts went into how to connect the web service to the application. This sprint went on to be a lot more time consuming then originally planned and after sprint 3 we realized that we would not be able to complete all user stories.

We decided to stop when profile and offers tab were completed, but some stories regarding increased functionality of profile and offers would not be completed during our work.

The team had decided on a rough estimation that each story point would take about 4 hours to complete. Due to underestimation of the size of some user stories, time put in and velocity of the team the team decided to increase this estimation to 5 hours in sprint retrospective 3.

5.5.1 Sprint burn down chart

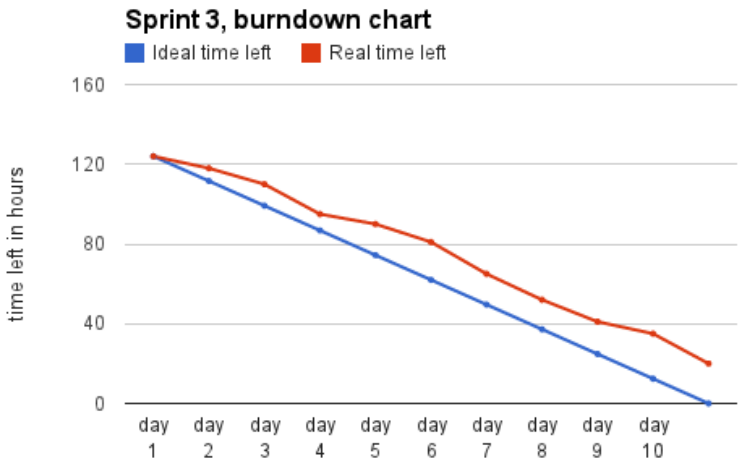

Figure 13, sprint 3 burn down chart

In sprint 3 the team completed 26 story points.

5.5.2 Worked hours

Total hours planned for each member of the team for sprint 3 was 50 hours or total of 150 hours for the team. Group members worked 198.25 hours in this sprint as we can see in figure 14 below which is about 32% over the planned schedule.

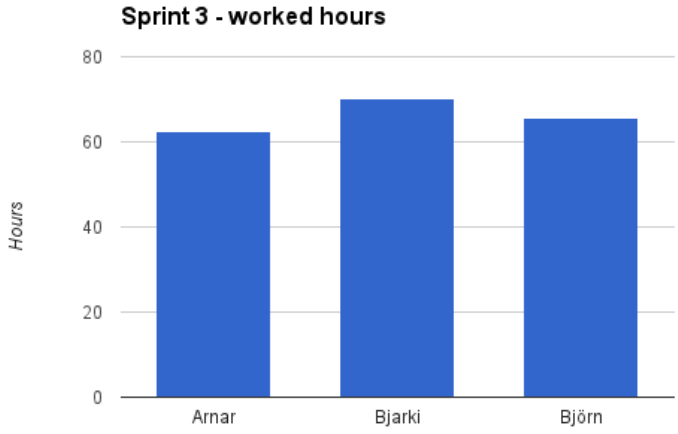

Figure 14, worked hours in sprint 3

5.6 Sprint 4

30th of October – 12th of November

In sprint 4 we took one story from sprint 3 about showing offers from gas companies in the application. Other stories in sprint 4 were about profile part of the project. This sprint went very well and all stories were completed

5.6.1 Sprint burn down chart

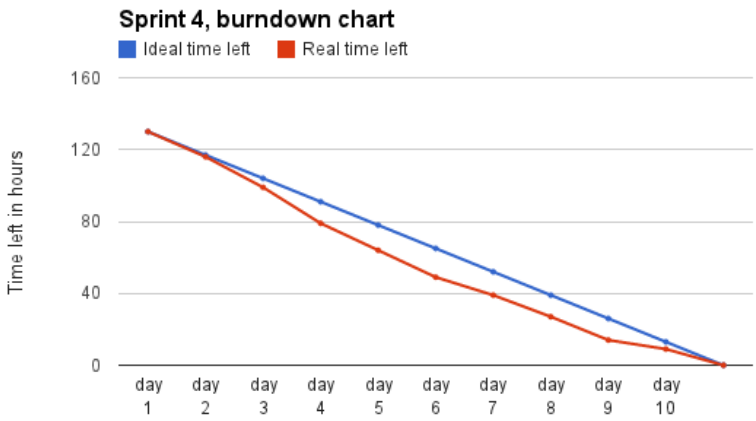

Figure 15, sprint 4 burn down chart

In sprint 4 the team completed all 26 story points.

5.6.2 Worked hours

Total hours planned for each member of the team for sprint 4 was 50 hours or total of 150 hours for the team. Group members worked 167.75 hours in this sprint as we can see in figure 16 below which is about 12% over the planned schedule.

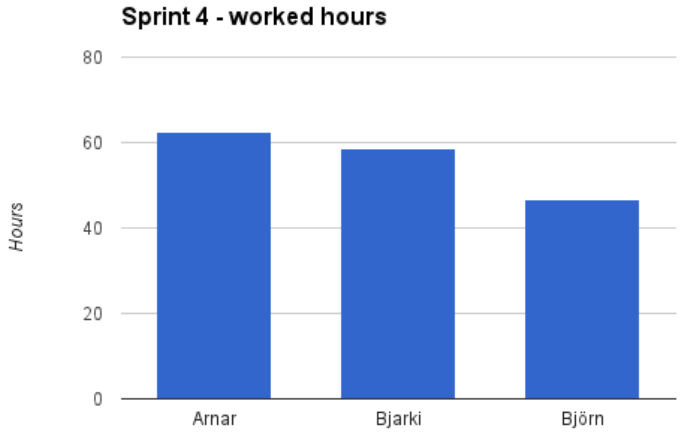

Figure 16, worked hours in sprint 4

5.7 Sprint 5

13th of November – 26th of November

In sprint 5 we took all stories that we wanted to complete in our project, we were not able to complete all of them. Two stories therefore were set for sprint 6, regarding simplicity testing and graph in the profile view. The team had to find time to do paperwork on user

guide and developer‘s guide. Time that went into these guides was over one third of our planned time for this sprint, so the team had to put in some extra efforts in this sprint.

5.7.1 Sprint burn down chart

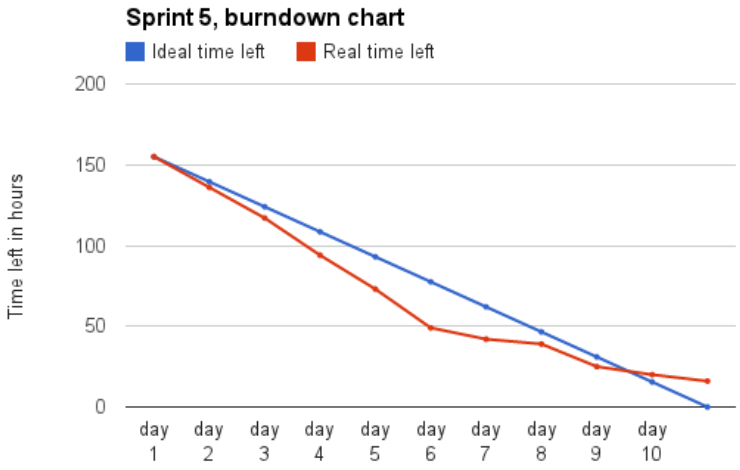

Figure 17, sprint 5 burn down chart

In sprint 5 the team completed 23 story points.

5.7.2 Worked hours

Total hours planned for each member of the team for sprint 5 was 50 hours or total of 150 hours for the team. Team members worked 154.25 hours on user stories in this sprint as we can see in figure 18 below. In sprint 5 the team worked on the developers guide and user guide, in total 53.5 hours went into the making of these guides. Team members worked for 207.75 hours in total in sprint 5 which is 38.5% over the planned schedule.

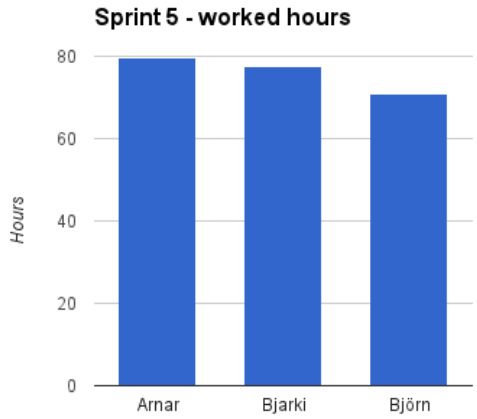

Figure 18, worked hours in sprint 5

5.8 Sprint 6

27th of November – 11th of December

In sprint 6 we took the two remaining stories from sprint 5 about simplicity testing and graphic part of the profile. The tests were done very quickly in this sprint and minor fixes were made in the application. Those minor fixes can be seen in the simplicity testing report. A lot of time went into paperwork, and the chart in profile view got completed.

5.8.1 Sprint burn down chart

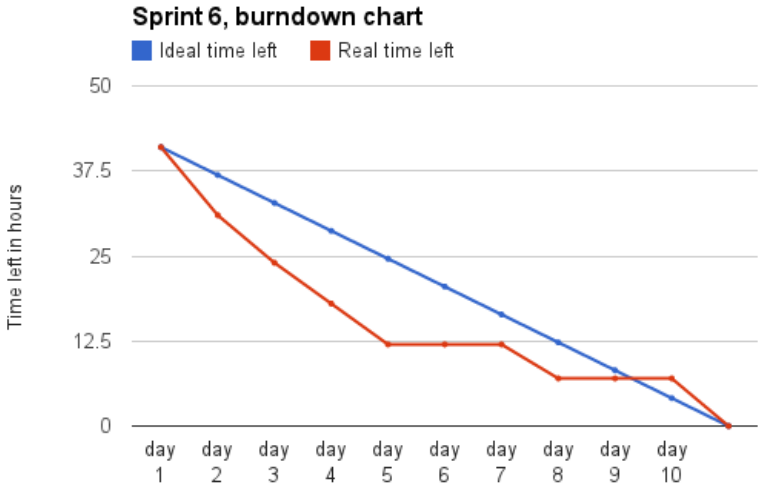

Figure 19, sprint 6 burn down chart

In sprint 6 the team completed all 8 story points.

5.8.2 Worked hours

Total hours planned for each member of the team for sprint 6 was 50 hours or total of 150 hours for the team. Group members worked 45 hours on user stories in this sprint as we can see in figure 20 below. In sprint 6 the team worked on the final report and other guides, in total 112 hours went into the making of the report and these guides. Team members worked for 157 hours in total in sprint 6 which is about 5% over the planned schedule.

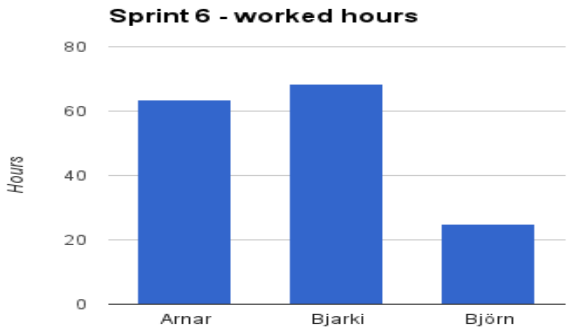

Figure 20, worked hours in sprint 6

5.9 Overview of hours worked in sprints

Figure 21 below shows the time each team member spent on each sprint.

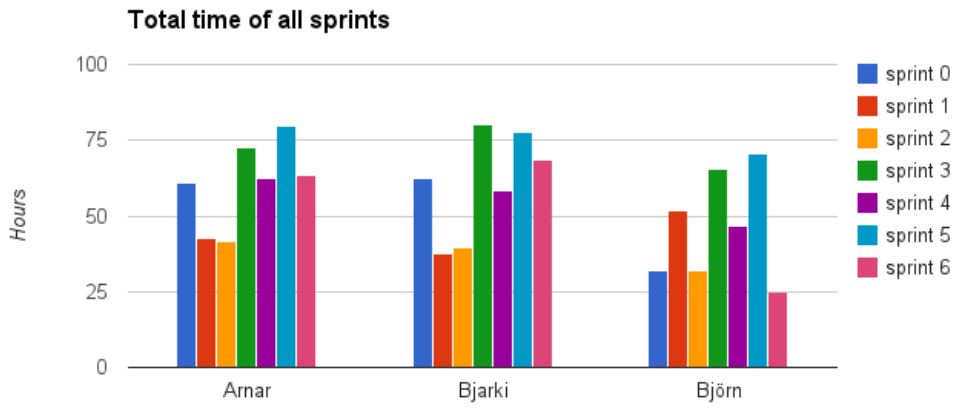

Figure 21, total time worked on user stories in each sprint

5.10 Total project burn down chart

Figure 22 below shows the project burn down chart. To measure the progress and how many story points our team was burning during each sprint. Our velocity was steady throughout the project. In the last sprint the team didn't burn down as many story points as in previous sprints, because a lot of time went into the final report and other guides.

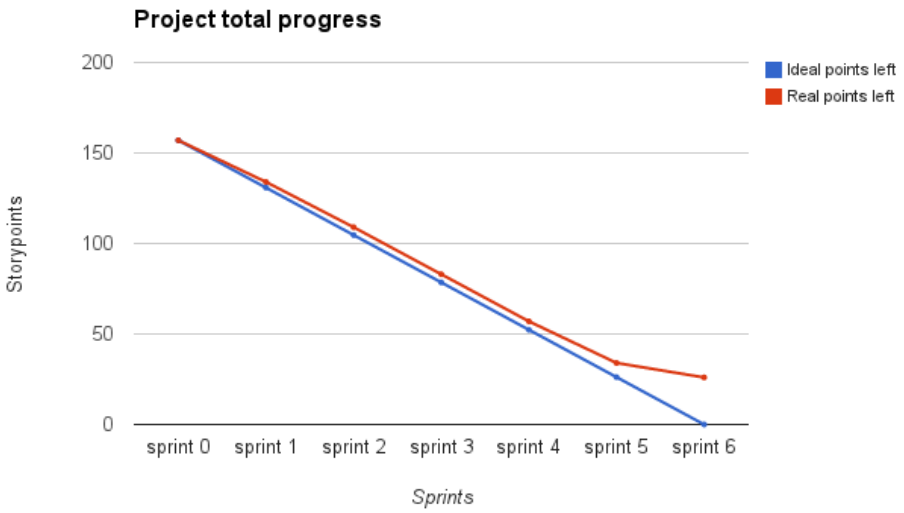

Figure 22, total project progress

5.11 Total worked hours

In the beginning the team planned 1035 hours for the project.

The team had spent 1170.75 hours on the project the day this report is written.

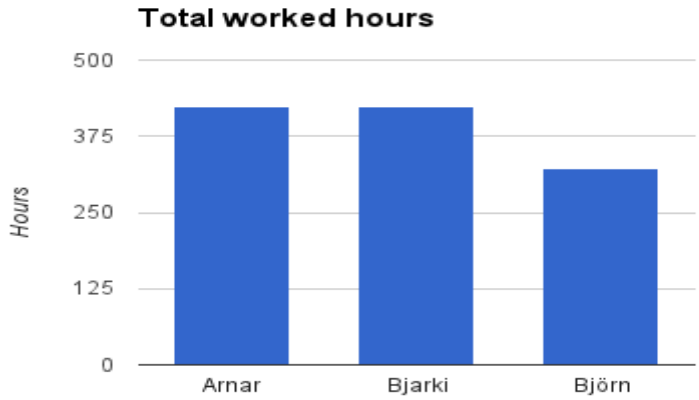

Figure 23, total hours worked on project

Total worked hours:

Arnar: 423 hours

Bjarki: 424.25 hours

Björn: 323.5 hours

Total : 1170.75 hours

6. Testing

One of our design criteria was to make the application simple in use. We found it important to verify that this goal had been met with our implementation. The testing was performed as follows.

First the individual was asked few background questions. Then they were asked to think aloud and let us know what they were thinking while pushing every button/tab. Everything that participants said was noted and afterwards they were asked to give 3 grades on the scale 1 -10 about certain aspects of the application and then asked about good and bad notes regarding their experience and finally they were asked if they would use this application.

The three questions were:

1. How simple is the application? (1 - very complicated, 10 - very simple)
2. How beautiful is the application? (1 - ugly, 10 - very beautiful)
3. How useful is the application? (1 - not useful at all, 10 - very useful)

6.1 Simplicity testing summary

Average from the results from the questions asked after performing the tests.

number	Question	average score
1.	How simple is the application?	8.89
2.	How beautiful is the application?	8.00
3.	How useful is the application?	9.00

Table 2, simplicity testing

Our participants completed the tasks very quickly in general, the longest time for a single task to be performed was 1 minute. In our task list the setup was that users were asked to perform something very similar two times in a row, almost every time the second task was completed faster than the previous task.

We set standards that if question number 1 on simplicity in table 2 would get an average score of less than 7.5 the application would not meet our standards in simplicity. The average score on the question met our standards and timing on tasks was very acceptable and these tests reassured us about the simplicity of our application.

For more information on tests, see the simplicity test report.

7. Conclusion

The team is proud of this project. We learned a great deal about application development and had a lot of fun during that time. The team started out with a project description and with hard work, determination and good organization we managed to create a working application that works on the most used platforms on phones today.

We would like to thank Entrio ehf. and special thanks to the two persons that we encountered the most during our work, the product owner Vignir Örn Guðmundsson and Entrio‘s only employee outside founders Gunnar Sigurðsson. Although the product could get „ more sexy “ with some extra features, it is a great starting point to build upon even more

7. 1 Improvements/ Future works

The profile part of the application can be improved further, with graphical display of data that the user enters, this can be in the form of gas per kilometer, gas prices and etc.

Publishing the application in English should get more users, because the tourist industry in Iceland is a fast growing industry and the application should be very useful for people that don’t know their way around Iceland.

Review from Entrio ehf. Correspondent

On behalf of Entrio, I had a great pleasure working with Arnar, Bjarki and Björn on the final project Gas Teddy Bear Bobo during the Fall of 2013. Right from the beginning of the collaboration it was clear that the group was ambitious to achieve quality results. The project was thoroughly worked on over the project period and the group showed initiative regarding product changes for the better. The final product, the mobile application Bobo, is more than consistent with the expectations that Entrio had when the work started.

Vignir Örn Guðmundsson

_____

Table of Figures

Tables