

SEM5640 Group Project
Individual Report

Go!Aber

*Submitted in partial fulfillment of
the requirements for the award of the degree of*

**MEng
in
Software Engineering**

Submitted by

110005643 Craig Heptinstall

Department of Computer Science



January 2016

Contents

1	Introduction	1
1.1	Personal evaluation	1
1.2	Group evaluation	5

Chapter 1

Introduction

This report looks into my personal views of the group project completed for the Go!Aber application. I will be evaluating the group performance, as well as my own personal performance in the following two chapters. I will provide critique and possible improvements for various times in the project, outlining what I felt when well, and what could have gone better. Throughout this evaluation, I will be using my personal blog that I wrote each week. The blog entries can be found in appendix

APPENFUX REFERENCE HERE FOR BLOG

1.1 Personal evaluation

To start by looking at my own contribution to the project, this began during the initial reading of the requirements specification. Alongside the other group members, I carefully read through and provided description about each specification, noting down any possible questions or issues. Following the minutes recorded (this was done by myself throughout the project), questions recorded were then put to the client. Again, I was responsible for creating the minutes from that meeting, and were placed in the project folder for access by any other members, client or project manager.

Staying on the subject of my role during meetings, in each sprint review with the project manager, I ensured that both the meeting room was free and also that major statistics such as burndown, sprint tasks completed, and hours worked were explained. This was an important task for me, as I was selected as scrum master. I was the member to suggest using scrum as the principle development methodology, following from my personal experiences during my industrial year.

I felt that both scrum, and my role as scrum master were both handled well, and that I ensured at the beginning and end of each sprint that the team as a whole were on track to complete the weeks tasks. Perhaps one improvement that could have bettered my performance as scrum master could have been to explain the roles and rules of the methodology better to other members after the workflow was selected. This could have improved the use of TFS (Microsoft visual studio online), where instead some members were slightly late on creating new tasks, and marking others completed when required.

Furthering the scrum responsibilities I held, I am happy with the way that I managed meetings for sprint planning. Where meetings had time restraints, I ensured that the team did not discuss certain issues for too long (unless necessary). For instance, in initial discussions of the class diagram, I suggested that because we were still only at the stage of designing the database, that the team should not get too far ahead, and take more time concentrating on current work.

Finally for my scrum master role, I worked closely with all other team members to ensure that the most suitable tasks were placed with a higher priority, therefore ensuring these tasks were completed first. For instance, creating a stable database was the first task that I agreed had to be completed before any other task could be started.

Another couple of tasks that I knew had to be completed early on in the project was the development of the Fitbit and Jawbone connections. Because I had some experience with API connections and SOAP/ Rest connectivity, I put myself forward for handling these tasks. Starting by browsing a few examples on the internet and some that were suggested in the practical lectures of this module, I was able to create external projects capable of connecting to the Fitbit servers from .NET. This was completed within the second week of the project, and the first sprint week.

Carrying this over, I was then able to create the same connection within the application .NET project. Although I managed to get the .NET connection working, the Java half was not started at the end of the first sprint. Mainly due to inexperience with the Fitbit API working with Java, I was unable to start this until sprint two, and this then pushed back my ability to start the Jawbone work. I asked help from a fellow member to complete the Fitbit connection, and put to use my experience of pair programming.

Although I enjoyed this way of development, and progress was made faster than by myself, we were unable to fix the issue with Fitbit on Java, and

therefore decided as a group (though advised by me) to move onto another task until more free time could be made to complete the Java connection. Though I felt that I could have prepared better for implementing the Java side of this task, I knew that there were still integral parts of the system to be completed.

In the following sprint I handled the task to control system permissions (revoking and checking connections) with Fitbit and Java. This task was completed on time in this case, and I also managed to help a fellow member with Jawbone .NET connectivity. The following sprints then consisted of me cleaning up the connectivity by refactoring and commenting, whilst also attempting to resolve the issue with Fitbit and Java.

Overall, I felt that the contribution towards the completed application was what was required, although I did not complete all my tasks. If I were to perform the same tasks again, then I would ensure my knowledge of API connectivity in Java was better. Also, if I knew the available libraries that I know now to help with such tasks, then the productivity of my task completion would be much faster. Even with my knowledge of using APIs with both languages from previous work experiences, I found that issues I may have had in the past are prevented. This project was a matter of refreshing my knowledge of writing good web applications.

Following on from this, during the initial stages of the project it was agreed that each member should apply the same coding standards to ensure consistency and readability. I felt that I stuck to the coding standards for each language well, and refactored after I completed tasks where necessary. I do feel however that I could have placed more comments in some parts of the work I completed, to allow other members to understand my code better, and also for the project manager or client if they wished to read into any of the code. I did ensure that all important methods in the code did have a description in terms of Javadoc or .NETdoc where needed.

Throughout the project, I also provided assistance to others where they needed advice on implementing certain features, such as the Jawbone connection, or help with the scheduler implementation concerning how often to update certain parts of the system. Although I feel I did not perform enough code reviews of others while in pull requests, I made up for this in the latter half of the project, and in most sprints reviewed and gave feedback on all tasks completed by others.

Once the majority of tasks were completed, I then volunteered to perform

cucumber testing on both projects, initially starting with the .NET application. Because I had lots of experience cucumber testing in the past, I let the other members of the team know that I would be best suited to this task. After trying out a few libraries, I found one that would work for both the .NET and the Java application, cutting down work time. Rather than creating two sets of tests, I was therefore able to use the same set for both applications, and write extension user interaction tests.

I was happy with the amount of tests and detail I performed, though as always with cucumber testing, more could have been added. If I had known about which library to use, and that a library could be used for multiple purposes at the start of the project, I could have created many more tests, and added more comprehensive boundary and extreme testing. I ensured that testing was completed with time to spare, to allow myself to complete documentation, and allow other team members to write about testing correctly in their sections of the report where necessary.

Due to the issues with Fitbit and other smaller setup issues at the beginning of the project, alike the other group members I did find myself behind schedule overall in the project, resulting in the documentation to be started quite late on. We initially planned for the project implementation to be completed at least a week before hand in, but instead I found myself writing the documentation only 3 or 4 days before hand in.

Even with this issue though, I was pleased to complete this a day early of the deadline, with time to create a skeleton project document for the other group members to add their own documentation to. Because I created the document, I also volunteered to clean up the document when all sections were added. My section (introduction and requirements) was proofread, while I proofread others. I selected to read through the testing documentation especially, as I had done quite a lot of testing (primarily cucumber).

At the end of documentation creation, I then did a general tidy of the document, and ensured that the group as a whole were happy with its final state. Once complete, I then handed this to the member who was responsible for submitting the project. Although I ensured I did not take too long creating the document layout, I made sure it was made to a very good standard.

In addition to the document, I also created a lot of the diagrams for the report and for the project manager to look at during the design stages. I chose to do this because during one sprint, I found myself completing my tasks for that week, therefore having time to create any UML diagrams required.

As I have had plenty of experience in the past creating UML, I feel that I have created a comprehensive set of diagrams explaining most the major components and usability features of the application. I also went through each of my diagrams with the other members on completion to make sure that the others knew what each diagram was for, so that in the documentation stage of the project the diagrams could be used and described.

In terms of keeping time and maintaining the schedule the group set out, although this started out rather negatively, I have been able to improve vastly over the second half of the project. I noticed that as I had completed tasks, I then knew what to expect in coming ones, therefore making it easier to assess the required workload for future tasks.

The blog I created to accompany this report was done on a weekly basis, and helped me greatly during the writing of this report, and also in providing useful information to both the project manager and the other group members in terms of my work efforts.

1.2 Group evaluation

In addition to reviewing my own performance and contributions to the group project, it is important to review and understand what and why specific parts of the project as a whole went well and did not go so well. To do this, I will evaluate each stage of the project and talk about how well I thought the team completed it.

Upon initially receiving the project, we set up a means of communication through Facebook and by exchanging phone numbers. I thought this was good, and made sure that all members were contactable if anyone had any questions for another. We used Facebook primarily, for arranging meetings, asking coding questions, and brief discussions about tasks. We did however ensure that meetings were used as the more detailed and formal approach of communicating tasks, as this way minutes could be recorded available for the client or project manager to read through.

When it came to understanding the requirements, we held a meeting as a group to discuss and question any specific details. Once this was complete, we then held a QA session with the client to clarify the questions we had made. I think that this was an important session to hold, as it then shaped the tasks we created for each sprint, and meant we could assign effort points

to each task.

After the first few weeks of implementation, the group had a good understanding of the scrum methodology, and the workflow was quite mechanical.