End of Project Report

SEG2012GP9

|  |  |
| --- | --- |
| Seabrook, Edward J F | [ejfs1g10@ecs.soton.ac.uk](mailto:ejfs1g10@ecs.soton.ac.uk) |
| Elliott, Kristian | [ke1g10@ ecs.soton.ac.uk](mailto:ke1g10@soton.ac.uk) |
| Yu, Brian | [by2g10@ ecs.soton.ac.uk](mailto:by2g10@soton.ac.uk) |
| Mariani Elola, Oscar N | [onme1g10@ ecs.soton.ac.uk](mailto:onme1g10@soton.ac.uk) |
| Chan, Yun Chak | [ycc1g11@ ecs.soton.ac.uk](mailto:ycc1g11@soton.ac.uk) |

# Notes to the Authors:

* *Max 20 sides A4 in total. This page limit excludes "wrapper" pages i.e. title page, contents page, document control page. Appendices will NOT be accepted; any appendices submitted will attract a penalty.*

Marks will be allocated as follows:

1. **20%** max - Presentation. High marks for
   * a well laid out document to house style
   * clearly written, spelling and grammar-checked narrative
   * well structured, clear charts and diagrams
2. **40%** max - Narrative account. High marks for
   * a comprehensive, appropriate, clear account
   * an account which correlates well with (i.e. explains) the other deliverables. Particular credit will be given for a strong account correlating with strong other deliverables. A weaker showing that still correlates will earn credit. Conversely, an account through rose-coloured spectacles will not match with other weak deliverables; this would reduce marks.
   * honesty - your account will be checked against your group supervisor's weekly reports and logbook marks. In extreme cases we would call logbooks in.
   * thoughtful reflection that shows what you've learnt about group process, and software development management in a group.
3. **40%** max - Project & time management accounts. High marks for
   * comprehensive, appropriate, clear accounts
   * an account which correlates well with (i.e. explains) the other deliverables.
   * an account of group members' contributions that correlates with the narrative project history account (above).

# Document Control

|  |  |  |
| --- | --- | --- |
| **Version** | **Author** | **Changes** |
| 1 | ejfs1g10 | Initial document |
| 2 | onme1g10 | Formation and Dynamics |
|  |  |  |

# Table of Contents

Notes to the Authors: 1

Document Control 2

Table of Contents 3

Introduction 5

1 Project History 5

1.1 Formation and Dynamics 5

1.2 Team Member Review 6

1.2.1 In general 6

1.2.2 Brian 6

1.2.3 Kristian 7

1.2.4 Kelvin 7

1.2.5 Oscar 7

1.2.6 Edward 7

1.3 Chronological Account 7

1.3.1 Week One 7

1.3.2 Week Two 8

1.3.3 Week Three 8

1.3.4 Week Four 8

1.3.5 Week Five 8

1.3.6 Week Six 9

1.3.7 Week Seven 9

1.3.8 Easter Vacation 9

1.3.9 Week Eight 9

1.3.10 Week Nine 9

1.3.11 Week Ten 10

1.3.12 Week Eleven 10

1.4 Evaluation 10

1.5 Individual Views 11

1.5.1 Brian 11

1.5.2 Edward 11

1.5.3 Oscar 12

1.5.4 Kristian 12

1.5.5 Kelvin 12

2 Project Management Account 13

2.1 Gantt Charts 13

2.1.1 Start Gantt 13

2.1.2 End Gantt 13

2.1.3 Reflection 13

3 Summary of Time Records 14

4 Time Management Account 15

4.1 Changes in Estimates 15

4.2 Analysis 15

4.3 Reflection 15

5 Conclusions 16

# Introduction

For all the members of the group, this was the first full software engineering group project that they have taken part in, from requirements analysis through to production and testing. Techniques learnt in previous courses we used: A strict software development lifecycle, the spiral model, was followed; and UML diagrams were used for communicating technical opinions and understandings.

Throughout the project, it was observed that working well as a team was more important that being able to work well individually. Weekly formal group meetings and more regular informal meetings enabled the team members to help each other out, and produce something far greater than could have been produced if they remained a mere collection of people.

This document contains a full narrative of the project, and reviews of the project. There is also a time management account and a thorough evaluation of the project.

# 1 Project History

A complete narrative account of the project history

## 1.1 Formation and Dynamics

Group Nine was formed originally in October 2011 to work on COMP2007’s (Software Analysis and Design) group project. The task was to develop requirements and design documents for a fictitious GPS device. Considering the four stages of Tuckman’s stages of group development (Forming, Storming, Norming and Performing), the team began COMP2012’s (Software Engineering Group Project) project with the belief that ‘Forming’ stages has already taken place.

Each team member had the opportunity to show their strengths and weaknesses. The group worked hard to discover, and complement, the strengths and weaknesses of the other members. The previous project was also very helpful in terms of solving possible differences in working styles, personal drive and expectations. It was felt that the ‘Storming’ stage had already been visited on a few occasions before the ‘Runway Redeclaration’ project began.

When the project plan was created during the first two weeks of the term, all team members agreed on dividing the workload equally and fairy. Each member showed a strong interest in taking part in every stage and area of the process. Everyone took part in the documentation, development and testing of the final product.

Each team member had a similar level of experience using the tools and techniques required for the project. A similar level of programming ability existed and everyone was familiar with the Eclipse IDE. Because of the similarity in ability, each group member was able to take part in every area of work, gaining the best possible set of experiences from the project.

The drive and enthusiasm of everyone lead to a working environment where communication was the main ingredient, everyone felt confident to contribute their opinion, and every opinion was taken into consideration. The group worked together towards a common objective and goal as a single entity. The group functioned as a “gelled team”[[1]](#footnote-1) rather than a collection of individuals.

Throughout the project, very few, if any, memorable conflicts arose. Most decisions that the group made, were made amicably, with each group member agreeing that it was the best choice. Fortunately throughout the project most members held heterogeneous opinions, however the few times there were differences, we were able to conduct ourselves professional and ensure nothing was taken too personally. Although there were no major conflicts surrounding it, the group did have to decide how the different classes would be separated out into model, view and controller. The team did initially have different opinions on this, but a compromise that everyone agreed was fair was soon reached.

Towards the end of the project, the team members became extremely busy with other coursework and their deadlines. This situation triggered stress and anxiety throughout the team. The fact that during the initial project plan, which was developed with little experience or knowledge of this kind of projects, the team failed to be accurate when assigning work hours to tasks, also resulted in an increased workload towards the final deadlines.

In conclusion, and especially when reflecting on the project, this team has the capability of self-evaluation. Planning, in particular, is a skill that was felt to have been developed and improved. Most importantly, it is the team’s belief that the experience gained in this kind of team development of reliable, well tested and documented software products is not only very valuable but also much necessary for a future of capable and experienced computer scientists and software engineers.

## 1.2 Team Member Review

### 1.2.1 In general

The amount of time that each member of the team contributed to the project is fairly similar. Unfortunately, there were times where everyone had fallen behind schedule, but as time progressed these hours tended to be made up for thanks to the commitment everyone felt to the group.

Since the Java skills of the team were fairly evenly distributed, all members invested time in programming. Oscar and Kelvin worked as a pair developing the model classes, this required them to have a good understanding of the problem domain, and translate this knowledge into code.

Kristian was working mainly on the visualisations, so also had to understand the logic of the model, and use this to create clear and concise representations of the information. Edward and Brian worked mainly on the Java Swing GUI, implementing the View components, and the Controller’s listeners.

Each individual member has put a lot of effort into the project, however there was a different pattern with respect to time for when the member puts the effort in and produces work. This is also reflected by the time expenditure graphs and SVN commit reports in the later sections of this report.

### 1.2.2 Brian

Contributed a lot of effort and work behind the scenes with Edward to ensure the GUI is well designed and polished. Throughout the project Brian ensured he understood the program so he could help the other members and offer advice. Near the end of the project, Brian shared implementing and tuning the Listeners and GUI classes with Edward. He also invested much of his time producing the reports.

### 1.2.3 Kristian

Effort and commitment was all dedicated to the visualisation aspect of View. Work on this began quite early but effort on this dropped in favour of other coursework near the middle of the project timeline. Work began again at the later stage of programming and showed rapid progress and improvement on the first visualisation. Although Kris was often difficult to motivate to work, the contributions that he made to the project were of a consistently high standard. This work was done very swiftly and team members were pleased to see regular commits and more pretty visualisations reflected by Kristian’s skill and effort. Kris also had the role of producing and updating the Gantt charts that were used throughout the project.

### 1.2.4 Kelvin

Worked in tandem with Oscar to produce great quality Model classes. Lots of meetings with Oscar ensured work was done on time and to a standard everyone else expected of the Model code. At the later phase of GUI programming, Kelvin helped out by changing and adapting the core Model – getters, setter and the main data structure - to a good standard and communicated the changes clearly. He also took on the role of time manager, collating, analysing and producing graphs of the time spent by each member of the group. Kelvin did a good job of providing a constant level of commitment throughout the project.

### 1.2.5 Oscar

During the early phases of the project, Oscar was placed in charge of the user interface design; he produced the initial wire frames on which the applications interface was based. Along with Kelvin developed the model code, thanks to frequent meetings, the two were able to produce good code, that fulfilled its requirements. During the later phase of GUI programming, Oscar was always swift to make the necessary alterations to his code to ensure requested features were implemented.

### 1.2.6 Edward

Edward began work early and contributed a lot to the group ensuring that the project plan was complete and that the first presentation went smoothly. Under the weight of other coursework his input to the project died down during the Easter vacation and the weeks surround it.

Towards the end of the project, once other coursework was out of the way, Edward began working a lot hard on the project, quickly making up for the lack of progress during Easter. He spent large amounts of time on the programming and on the production of the reports. Edward conducted several code reviews, ensuring that the classes were well written, correct and properly documented, he also proof read, and reworded both D3 and D4 to ensure they maintained a consistent tone where necessary.

Although the consistency of the amount of work Edward produced on a week by week basis was fairly poor, throughout the project he contributed more than any other member of the group, acting as a key force in bringing the group up to a good level of progress.

## 1.3 Chronological Account

The project progressed like this

### 1.3.1 Week One

Both the first formal meeting and the first informal meeting took place. The project plan was discussed and jobs were allocated to every members of the group. The team held a casual skills audit to decide which tasks should be assigned to which member. Large amounts of time were spent discussing hour allocation during the meetings; decisions about work allocations were generally agreed upon by everyone. In the cases where this was not true, compromises that were deemed satisfactory by the group were made. The project was broken down into small chunks of manageable tasks and assigned to members who were the most suitable.

### 1.3.2 Week Two

Kelvin was appointed as the time manager – it was made his responsibility ensure deadlines were stuck to and hours worked recorded in a consistent manner (demonstrated in the Project Plan). The role of Document Manager was assigned to Edward, his duty was to ensure that the documents submitted by the group were well presented, neatly structured and were easy to read. The Project Plan (deliverable 1) was handed in promptly on the morning of the deadline day. It was proof read in full by three members of the group the day before the hand in and printed by the Document Manager as agreed. By handing the document in with plenty of time, the group felt that it was good that we didn’t have a rush or panic to hand in this coursework.

### 1.3.3 Week Three

A good folder structure for the SVN repository was decided upon. Working pairs were appointed: Kelvin and Oscar were assigned the Model. Kristian and Edward were given the task of implementing the View. Brian was assigned the Controller. Architectural design and HCI design was discussed in formal and informal meeting as a group. UML was used extensively to help communicate the various aspects of the project. The working pairs help frequent gatherings to discuss and design their parts to encourage progress. Not a lot of communication to do with the design of the inner workings of Model or View between the working pairs happened at this stage. The progress of the design for Model and View was reviewed but not communicated much in the formal group meeting at the end of the week.

### 1.3.4 Week Four

The actual coding of the Model and View classes began. The respective working pairs worked almost completely separately. Communication and discussion between the pairs about the design of their parts was still essentially non-existent. Formal group meeting reviewed progress on the code. A small demonstration of what the Model and View code was capable of was presented in the formal group meeting. Each working pair continued to work separately. Brian, who was unable to begin implementing the Controller before the Model and View have been produced began helping out with View offering design consultancy.

### 1.3.5 Week Five

The group began discussing ideas for the presentation (deliverable 2). The UML diagrams that we would be presenting were finalised. A long library group meeting was held on the Sunday afternoon for this work to be done as a group because the presentation was to be presented by the whole group and needed to be dynamic and flow nicely. This meeting included designing and making a draft version of the slides and rehearsing it for the first few times. The group dynamics were relaxed but productivity was high in terms of the amount of work produced and how far we got in preparation. A middle Gantt chart was made for the presentation. There were small conflicts in the design and ordering of the slides and of the amount of progress we reckon we made for the middle Gantt chart but it was resolved quickly and everyone was happy. This was also a point where we discover who needed to sharpen up their presentation skills – group members were helpful in offering critical but constructive opinion on others who fell short of expectations and everyone showed patience and persistence in rehearsing the presentation multiple times until everyone was at a satisfactory level. The group agreed to freeze the code after this week in preparation for the application demo for D2.

### 1.3.6 Week Six

The agreed code freeze for the demo was carried out the day before the presentation. On the day of presentation we had a final rehearsal before the presentation of D2. Final rehearsal was done in an informal meeting an hour before the presentation so everyone knew what they are presenting and were fresh with the presentation in mind before the presentation. Everyone was satisfied with how the presentation went in terms of the performance and the immediate feedback from RJW. Coding continued in the working pairs separately after the presentation. D1 marks were received and everyone appeared happy with the mark. Banter and report of presentation and D1 marks in the formal group meeting at the end of the week.

### 1.3.7 Week Seven

Reviewed coding progress. Work begins on D3. D2 marks were received and met with approving nods all round from members of the group. Controller code begins. It was clear that some essential dialog boxes were not coded to enable listeners to be written for them so Brian had to help speed up the coding of these. Confidence level was high heading into Easter.

Lesson learnt: be more specific about work that needs to be done – it was clear that the need to specify which dialog boxes that the Controller needs to be able to start coding was missing.

### 1.3.8 Easter Vacation

Due to a large amount of coursework set by other modules, and the fact that the group members had been split up geographically, the group did not plan to do a large amount of work over the Easter vacation. The group did in fact make very little progress during these weeks.

### 1.3.9 Week Eight

Splitting /src folder into model and view was no longer a good solution as controller is starting to be developed more. /src was refactored and everyone was informed and adapted in good time. Discussed the plans for D3, D4 and D5. Code review.

Lesson learnt: be more specific at the planning/design stage: define folder structures for paper work and also define packages for coding.

Lesson learnt: discuss plans for looming deadlines earlier to prevent coding or documenting rush.

### 1.3.10 Week Nine

Not establishing interfaces well enough made programming Controller hard. Brian had to communicate frequently with both Model and View groups to get an idea of how those components work. Edward decided to make interfaces which needed another informal meeting to discuss with Brian. Rush to code up solution.

Lesson learnt: be more specific in the Design stage: define interfaces well before coding begins.

Lesson learnt: code review meetings need to be stricter about the amount of work done; not let other course works derail progress. Need to improve time management on an individual level.

### 1.3.11 Week Ten

Expected hand in for this document.

### 1.3.12 Week Eleven

Final presentation. Decided to hand this document in along with that to make way for revision.

## 1.4 Evaluation

As in every formal and professional process, evaluating it is a very important part of it. Keeping an objective point of view, being able to recognize own mistakes and suggesting new ways to approach previous issues is vital in the learning process. This team believes it is very positive and necessary. Therefore, when the project is close to reaching its end, the team members of this group self-evaluated their work, participation, and mistakes. After hours of debating about this subject, the the team realized that in general, each member reached the same kind of conclusions.

To begin with, the importance of proper and methodical planning was highlighted in every team member’s head. Mainly because at this point of the way, it is very noticeable and fairly easy to see that, with a better planning process at the beginning of the project, the team’s work would have been much more organized and most stressful moments or situations would have been easily avoided. In fact, truth is that the project planning of this team was far from accurate in many areas, not because of unwillingness to to a good enough plan, but mainly because none of the team members had enough experience as to for having a clear idea of how much time and dedication it section of the project could take. In fact, miscalculations in the initial project plan were huge. To many hours dedicated to programming, not enough for testing, and the list goes on. This weaknesses in the project plan were eventually identified, in fact, some of them were mentioned in the first presentation where gantt charts were already modified accordingly, but truth is that even today, when the end is near, some of the consequences of bad planning are still present. Each member got busy with other University modules, and the deadlines got closer and closer, and in order to adapt, some members worked extra hours, others did to much work on the same day, which is not ideal. It is clear now that planning is one of the most important parts of any serious project, that planning deserves serious work hours invested into it, and that with a good project plan properly developed and in place the rest of the project becomes much more organized from day one.

The other big aspect of the project that this team realized it has to be done better, it must be improved on a future project, has to do also with planning, but in a different shape. This time regarding the actual code. In a team with several members, where a particular architectural pattern is chosen in order to modularize and share the coding amongst all team members, apart from the obvious benefits of adopting a particular architectural pattern; integrating everyone’s classes into a single working piece of software can become extremely tricky, in fact, it did. And it was thinking about this that this team derived its second big area that must be improved on a future project, and that is developing clearly defined interfaces beforehand. Good interfaces, represented on a good class diagram which has a good level of detail is very necessary in order to improve a later integration of the modules of any project. Once a serious class diagram with clear interfaces has been designed, programming each module becomes simpler, but the biggest gain is when the modules are integrated into one. If the way classes interact with each other is decided in advance, no serious modifications to the code need to be made at late stages, where even simple modifications generate errors and bugs on every other module of the system. It is this team’s opinion that the workload generated by the lack of good interfaces thought of in advance is nowhere close to the effort needed to produce high quality, professional interfaces at the design stage of any project.

Finally, this team has truly learned the importance of fully understanding the whole situation or problem the team is trying to be solved, beforehand. The importance of having a very clear and complete understanding of what the team is dealing with is notoriously higher than the one this team initially thought. It is important that every team member is comfortable with every aspect of the situation so that over meetings and discussions every team member’s contribution becomes a solid, informed opinion about the matter. This paragraph is not trying to imply that this team was not familiar with the task, it is merely stating that each of its members can now seriously value the importance of it.

In conclusion, this team believes that, despite any mistakes or errors done, despite overseen areas of work, despite ay differences amongst any of the team member, and despite any stressful situations that could have been generated; this project, taught each of the members many important lessons. Each of the member of this team will face future similar situations in a different way, each member learned the importance of many areas and stages in the development of a serious software product. This team knows that, this approach to what a project in real life would be will prove to be of great benefit in the future; in future University projects yes, but mainly, and most importantly, it will have a deep and beneficial impact on each team member’s careers as professional and proficient computer scientists or software engineers.

## 1.5 Individual Views

### 1.5.1 Brian

“The group was generally great to get on with but people held their views on design quite strongly and took a lot of time for small but important decisions to be made. My opinion on not having proper interfaces done earlier on is that people felt it was too complicated to figure one out as a group (in a meeting setting or otherwise) which meant that people had to do a lot of running around to find things which should, really, have been agreed at the design stage especially me as I am in charge of the controller part of the code. This strategy is not ideal or efficient and would scale badly to larger projects.”

### 1.5.2 Edward

“Our group works well together and I feel we are able to be very productive. Unfortunately I don’t feel that we put enough time into planning the inner workings of the system in the early stages of the project and as such met slight problems when it came to the integration of the project. We were able to produce a system that works well for a project of this scale, but if it were to grow substantially then a slight refactor would probably be needed - I don’t think this would be a huge task, but it would probably require us to step back and rethink some of our strategies. On a whole our group managed to get the work done, however I do feel that I put more time into the project than the other members of the group; this is however probably mainly to do with the fact I am somewhat of a perfectionist. I feel that this project has been a great learning experience, as I have learned what I am good at, and what I need to improve at.”

### 1.5.3 Oscar

“In my opinion, despite any differences or difficulties we might have came across, the value of this project as a learning tool is of great proportions. I value and thank those moments were the stress was obvious amongst every team member, those moments when I had to stay up late or working way to many hours on row in an effort to catch up where we fall behind, when we thought we were in trouble; I treasure all of those because I know as a fact that tomorrow, when I face a similar real life situation, when I start working on a project with a new team and we are designing a project plan, when I have free time and I do not feel like doing some work; it is in those situations when I will put to use what I’ve learnt today.

I believe that all of us should feel the same way, and I will finish this project happy and pleased.”

### 1.5.4 Kristian

“I am Kris and I love working really hard”

### 1.5.5 Kelvin

“I am Kelvin and I love working really hard”

# 2 Project Management Account

Introductory spiel.

## 2.1 Gantt Charts

Chat about what gantt charts are and why they are helpful?

### 2.1.1 Start Gantt

Date Created, The chart itself. One side A4

### 2.1.2 End Gantt

Date created, the chart itself . One side A4

### 2.1.3 Reflection

reflection on the transition from start-midway-end, i.e.,

how estimates of cost and timescale for each activity changed over time.   
What did you learn about planning these activities?   
Did your estimation accuracy improve?   
How far out were your initial estimates?   
How well did you manage the project and what, with hindsight, would you differently?

# 3 Summary of Time Records

(2 sides A4 max) Summarize as a table, with group members on one axis and week number on the other, the time spent per activity/ member/ week. Don't worry if this report doesn't agree exactly with the end-project Gantt chart, although the two shouldn't be TOO different.

Not sure if this section needs an narrative since that seems to be covered in the next section.

# 4 Time Management Account

## 4.1 Changes in Estimates

Based on the summary report. A clear account of the initial activity/ time estimates and how these changed over time (over to you to avoid overlap with 2. above).

## 4.2 Analysis

Graphs as floating figures, discuss what they mean in the narrative.

Analyses (graphic and narrative) of time spend by week/ activity/ member. Discussion of any time management decisions requiring departure from the 100 hours budget, e.g. choice to overspend to add value to the product, or the choice to spend the "ghost" budget of an absent (ill or dropped out) group member.

## 4.3 Reflection

Reflection on the effectiveness of time usage and management.

# 5 Conclusions

Again a non-specified section, but RJ will probably not mind if we summarise everything here ;)

1. Peopleware: Productive Projects and teams. [↑](#footnote-ref-1)