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

1.4 Evaluation 9

1.5 Individual Views 10

1.5.1 Brian 11

1.5.2 Edward 11

1.5.3 Oscar 11

1.5.4 Kristian 11

1.5.5 Kelvin 11

2 Project Management Account 12

2.1 Gantt Charts 12

2.1.1 Start Gantt 12

2.1.2 End Gantt 12

2.1.3 Reflection 12

3 Summary of Time Records 13

4 Time Management Account 14

4.1 Changes in Estimates 14

4.2 Analysis 14

4.3 Reflection 14

5 Conclusions 15

# Introduction

This project was the first software group project which the team members have taken part in. We made use of the full software development process through requirement analysis, design, coding and testing to the final deployment of the software.

For members where this is a first large group software project, it will prove interesting and different from the individual efforts that is typical of most university computer science coursework. In this report we look at the how the project went from the perspective of the team as a whole and each individual member’s perspective.

This document narrates, concludes and evaluates our project. It contains a full history of the project along with details of the groups formation and time expenditures.

This report contains individual team member’s thoughts on how the project went and how the group got on and, most importantly, the management aspect of the project. This is where the group looks back on the process of software development, reflect on the goods and bads and what could be done differently in the next group software project each one of us will do – whether this is in a university environment or as a professional in industry. By reflecting back, we can learn what we can do better next time.

# 1 Project History

A complete narrative account of the project history

## 1.1 Formation and Dynamics

Group nine was formed on October 2011 to work on COMP2007’s (Software Analysis and Design) group project were the task was to develop requirements and design documents for a fictitious Global Positioning System based device. Considering the four stages of Tuckman’s team development cycle (Forming, Storming, Norming and Performing), this team began COMP2012’s (Software Engineering Group Project) project with the belief that ‘Forming’ stages were definitively behind. Each team member have had opportunity to show his strengths and weaknesses, and to realise about each others. That previous project was also very helpful in terms of solving possible differences residing within working styles, personal drive and expectations. Therefore the ‘Storming’ stage was already visited on a few occasions before even getting started with the ‘Runway Redeclaration’ design.

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. Every member showed interest in taking part of every stage and area of the process. In that way, everyone took part in the documentation, development and testing of the final product. As Java was the programming language of choice, Model View Controller the architectural pattern, Eclipse the Integrated Development Tool, and every member involved had previous contact with those technologies over the course of the degree, every team member, from the most experienced programmers to the least experienced ones were ready to take part on every area of work. The drive and enthusiasm of everyone lead to a working environment were communication was the main ingredient; were everyone felt confident to speak his voice; were everyone was heard and taken into account; were everyone was working together towards a common objective and goal as a single entity. Effectively a team, as opposite to a set of individuals working together.

Reflecting about conflicts or tensions arisen, it is difficult to pinpoint a particular example or occasion were something like that happened. This does not indicate frequent episodes of that nature, in fact it means exactly the opposite. Of course there were times were decisions had to be made and the pool of opinions and ideas was very heterogeneous, but that did not mean tension or problems at any time; if anything, slight delays. Even when differences of opinion did not disappear with time, that only meant that eventual chats about it were recurrent but in the end, everyone accepted that it would be impossible to go both ways. An example of that would be the structure of classes within each module of the system (Model, View and Controller). Even if the general class structure and relations of the system were decided and properly documented at the design stage, minor changes had to be made on the go (being faithful to agile and iterative development); and in the end, the sub team in charge of the module had the last word on how to structure the classes when an agreement could not be reached.

Towards the end of the project, every member became extremely busy with other course works and their deadlines. This situation of course triggered stress and anxiety amongst all team members. 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 specially when reflecting about the project, this team has the capability of self evaluation. Planning, in particular, is a skill that was developed and improved. The disadvantages of procrastination were, once again proven true. But 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

At the time of handing in this report, this team can say that the number of hours and effort invested into the project by each member is similar. Each individual team member fall behind on expected worked hours at least once during the project, but each member was very committed to the project and managed to do the extra work necessary to bring himself up to the expected worked hours again.

Every team member had contributed their skills in Java but different skills: the members in charge of Model (Oscar and Kelvin) used the core Java skills – creating classes for utility – setters, getters, iterator patterns etc. which was different to the skills used by members in charge of the View (Kristian and Edward) – creating the runway visualisation and the GUI – using the Java Swing API which was also different to the skills used by Brian (who was in charge of programming the Controller aspect) – creating listeners – using Java Events and Listeners API and needed to adapt to applying these APIs to link the Model and View.

Each individual member has put 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 scene with Edward to ensure the GUI is well designed and polished. Near the end of the project, was sharing the Listeners and GUI work with Edward and produced a large part of this report.

### 1.2.3 Kristian

Effort and commitment was all dedicated to the visualisation aspect of View. Work on this has begun quite early but effort on this dropped in favour of other courseworks 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. 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.

### 1.2.4 Kelvin

Worked in tandem with Oscar to produce great quality Model. Lots of meetings with Oscar ensured work was done on time and to a standard everyone else expected of Model. At the later phase of GUI programming, 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. This reflects the amount of effort Kelvin puts into the project: it was constant throughout.

### 1.2.5 Oscar

Worked in tandem with Kelvin to produce great quality Model. Lots of meetings with Kelvin ensured work was done on time and to a standard everyone else expected of Model. At the later phase of GUI programming, helped out by changing and adapting the saving and loading to XML file aspect of Model to a good standard and communicated the changes clearly. This reflects the amount of effort Oscar puts into the project: it was constant throughout.

### 1.2.6 Edward

Is like a chocolate darling.

## 1.3 Chronological Account

Chronological account – the numbers represent the week number

1. First formal meeting, first informal meeting. Plans for project plan discussed and jobs allocated. A lot of time was spent discussing hour allocation in meetings. Decisions to do with work and hour allocations were generally agreed by everyone or adjusted slightly. A lot of time was spent on this section as a group because the project had to be broken down into small chunks of manageable tasks and assigned to member(s) who were the most suitable.
2. Deadline for project plan (D1). Appointed time manager as a solution to making sure deadlines are being stuck to and hours worked is being recorded in a universal format – this format was presented in the Project Plan. Appointed document manager to ensure work that we hand in as a group reads consistently and uses a uniform house style. The Project Plan was handed in promptly on the morning of the deadline day. It was proof read in whole by three members of the group the day before the hand in and printed by the document producer as agreed. The hand in was not a rush; as a group, we felt like it was a polished document and were satisfied with the process we went through in producing the work for this hand in.
3. Established a proper folder structure for SVN. Working pairs was appointed: Kelvin and Oscar on Model, Kristian and Edward on View and Brian on Controller. Architectural design and HCI design was discussed in formal and informal meeting as a group. UML work begun. Working pairs meeting up separately 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 was happening 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.
4. Coding begins on Model and View by each working pair separately. Communication and discussion between the pairs about the design of their parts is still small. 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 discuss/design/code separately. Brian the Controller person helps out with View offering design consultancy as View is highly tied with Controller and not much can be done in terms of writing Controller code can be done at this stage as interfaces were not defined between Model, View and Controller.
5. Discussion of ideas for D2 begun. Prepared code and presentation for D2. UML diagrams 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 will be presented by the whole group and needs to be dynamic and flows 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. Agreed to freeze code after this week in preparation for the application demo for D2.
6. 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.
7. 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.
   1. 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.

\*\* Easter vacation: no work was planned to be done and no work was actually done. \*\*

1. 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.
   1. Lesson learnt: be more specific at the planning/design stage: define folder structures for paper work and also define packages for coding.
   2. Lesson learnt: discuss plans for looming deadlines earlier to prevent coding or documenting rush.
2. 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.
   1. Lesson learnt: be more specific in the Design stage: define interfaces well before coding begins.
   2. 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.
3. Hand in for this document.

## 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 ;)