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) |

# Document Control

|  |  |  |
| --- | --- | --- |
| **Version** | **Author** | **Changes** |
| 1 | ejfs1g10 | Initial document |
| 2 | onme1g10 | Formation and Dynamics |
| 3 | by2g10 | Added introduction |
| 4 | onme1g10 | Evaluation |
| 5 | by2g10 | Chronological narrative |
| 6 | ejfs1g10 | Reviewed and rewrote document |

# Table of Contents

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 6

1.2.4 Kelvin 7

1.2.5 Oscar 7

1.2.6 Edward 7

1.3 Chronological Account 8

1.3.1 Week One 8

1.3.2 Week Two 8

1.3.3 Week Three 8

1.3.4 Week Four 9

1.3.5 Week Five 9

1.3.6 Week Six 9

1.3.7 Week Seven 10

1.3.8 Easter Vacation 10

1.3.9 Week Eight 10

1.3.10 Week Nine 10

1.3.11 Week Ten 11

1.3.12 Week Eleven 11

1.4 Evaluation 11

1.5 Individual Views 13

1.5.1 Brian 13

1.5.2 Edward 13

1.5.3 Oscar 13

1.5.4 Kristian 13

1.5.5 Kelvin 14

2 Project Management Account 15

2.1 Gantt Charts 15

2.1.3 Reflection 15

2.1.1 Start Gantt 16

2.1.2 End Gantt 17

3 Summary of Time Records 18

4 Time Management Account 19

4.1 Changes in Estimates 19

4.3 Reflection 24

# Introduction

For all the members of the group, this was the first full software engineering group project taken part in, all the way 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 than being able to work well individually. Weekly formal group meetings and more regular informal meetings enabled the team members to help each other and produce something far greater than could have been produced if they remained a mere collection of people.

# 1 Project History

## 1.1 Formation and Dynamics

Group Nine was originally formed 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 professionally and ensure nothing was taken too personally. Although there were no major conflicts surrounding it, the group did have to decide how the different Java 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 very 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 visual 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 Kristian 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 better visualisations reflected by Kristian’s skill and effort. Kristian 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.

Kelvin also took on the role of time manager, collating, analysing and producing graphs of the time spent by each member of the group. He spent a long time producing the Time Managers report for the End of Project Report. 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. He also wrote a large proportion of the End of Project Report.

### 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 decreased to very low 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 harder 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

### 1.3.1 Week One

During the first week the group held both the first formal meeting and the first informal meeting; the first formal meeting was the first time the group met its supervisor, during this meeting it was decided that the weekly meetings would take place every Friday at 15:00 as long as everyone was available. The group chose to rotate the chair and recorder each week to allow each member to get an equal experience in these roles.

The project plan was discussed and jobs were allocated to every member of the group. Predictions were made on the time each task would take to complete. The team held a casual skills audit to decide which tasks should be assigned to which member - this generally agreed upon by all members, in the cases where this was not true, compromises that were deemed satisfactory by the group were made.

Decisions were also made about which tools to use. These choices can be seen in the Analysis, Design and Testing Document. The group also made ‘high level’ plans about how the application would be implemented.

### 1.3.2 Week Two

Kelvin was appointed as the time manager, this role required him to ensure time is logged correctly by each member of the group, and kept up to date. He is also responsible for analysing the timesheet data, and producing reports, charts and graphs which can be seen in other sections of this document.

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 group found that creating one document that was only edited by the Document Manager ensured a consistent style and tone.

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 there was no frantic rush or panic to hand the document in.

The use cases of the software were identified, and the most important classes of the application were specified.

### 1.3.3 Week Three

The group decided upon a folder structure for the SVN repository – it was intended to keep the repository organised ensuring that the files team members were looking for could be easily found.

Working pairs were appointed: Kelvin and Oscar were assigned the Model, they were concerned with the classes that model the problem domain. Kristian and Edward were given the task of implementing the View, both the visualisations of the runway on the screen, and the Graphical User Interface. Brian was assigned the Controller, mainly implemented using Java Action Listeners.

Architectural design and HCI design was discussed in the formal and informal meeting as a group. UML diagrams were used extensively to help communicate the various aspects of the project. The group learned how valuable UML diagrams can be for conveying ideas.

The working pairs held frequent meetings to discuss and design their parts to encourage progress. But there was a very limited amount of communication between the pairs. The group was kept updated about the progress each pair was making, but the inner workings were not communicated openly to the rest of the group.

### 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. The formal group meeting was used to review progress on the code.

The View was at first implemented by hand, but then after to a suggestion from the group’s supervisor, Google Window Builder Pro was used to create the GUI instead. The menu items that would appear in the finished product were agreed upon during an informal meeting.

A small demonstration of what the Model and View code was capable of was presented in the formal group meeting. Brian, who was unable to begin implementing the Controller before the Model and View code had 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). An all-day session was booked in a library room to allow the group to agree on the content of the presentation, produce the slides and rehearse the material. Neat versions of UML diagrams were produced to be included in the slides for the presentation.

The atmosphere of the group during week five was very relaxed; the members were all getting on well together and being very productive. A middle Gantt chart was also produced to help understand how well the budget had been stuck to, and allow a new estimate of how long each task would take to be produced.

The group rehearsed the presentation a few times, but in hindsight, more rehearsals would have been better. A rehearsal in front of a small audience might have helped some of the team members with their presentation skills.

The group agreed to freeze the code after this week in preparation for the application demo for D2. The amount of non-presentation work that was contributed during the fifth week was quite low due to conflicting coursework deadlines.

### 1.3.6 Week Six

The agreed code freeze for the demo was put into action the day before the presentation. On the day of presentation a final rehearsal took place before the presentation of D2. The final rehearsal was held during an informal meeting an hour before the presentation to ensure everyone knew what they are presenting and the material was fresh in their minds.

Everyone was satisfied with how the presentation went in terms of the performance and the immediate feedback from RJW. The mark for the project plan was also received and all members of the group were reasonably satisfied with the grade.

After the presentation a small amount of coding took place within the pairs.

### 1.3.7 Week Seven

The progress of the coding was reviewed and the group realised it was a little behind where it should ideally be with respect to the Project Plan. The group blamed this on conflicting coursework. Since the View has still not been completed, Brian began helping by producing some dialog boxes. From this the group learned that it needed to be more specific about specifying tasks that needed to be completed as some dialog boxes were not thought of but was needed.

The group began work on deliverable 3, the Analysis, Design and Testing document; a template was created for the document and small amounts of notes were added to it. Only a small amount of progress was made on this document before the Easter break.

Marks for Deliverable 2 were received and the members of the group were generally satisfied with the marks. The group entered the Easter vacation possibly a little bit over confident about the project.

### 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. Fortunately the group had not allocated large amounts of work to be done during the Easter break, so the impact of the lack of contributions was not severe; finishing the other coursework during the holidays also gave the group more time to be spent on the project once they returned to Southampton.

### 1.3.9 Week Eight

During the first week back, very small amount of work was carried out by the group on the project overall. This was due to more urgent (but not necessarily more important) coursework deadlines from other modules. From this the group realised that the members need to learn to plan their time more effectively: important tasks should have been dealt with of before they became urgent tasks.

The file structure of the SVN repository was altered to help facilitate the integration of the View and the Model code via the creation of the Controller code. A few Action Listeners were implemented to act as the Controller.

At the end of the week Edward performed a code review on the Model, and produced Javadoc style comments for the classes.

### 1.3.10 Week Nine

As the final deadlines came closer, and noticing that he was way behind in terms of hours Edward began putting a lot more effort into the project. He attempted to motivate the group into putting more hours in, but they were hard at work dealing with large amount of other courseworks.

The group found that with the integration (as they had been warned), the interfaces has not been thoroughly thought out, and so were slightly clumsy to use. Together, Brian and Edward came up with the solution of using the Observer Pattern[[2]](#footnote-2) to ensure that the required functionality was implemented in the neatest way possible.

The group learned from this that they need to spend more time specifying the programs design and ensuring that the various interfaces are published, and will all have the desired functionality.

Edward continued to work hard on the project throughout the week, and by the end of the week was joined by the other group members in a final sprint towards finishing the project.

### 1.3.11 Week Ten

Every evening of Week 10 Edward and Kris met in the undergraduate labs and worked hard to finish the coding. The GUI code was completed along with the model. By the end of the week the group was left with a product that could pass the User Acceptance Test and also implemented a few extension features. By this point Edward had added a lot more hours to his timesheet than the other member of the group.

Kelvin, Brian and Oscar had focused more on the End of Project report which was due to be handed in on Thursday, coding only when bugs were found, or features that had been omitted were requested. Kelvin and Oscar also spent a large proportion of the week testing their code.

The group was very glad that an extension was issued on this deliverable as it enabled a far more polished and complete document, that really reflects our ability, to be produced. The group did however decide that to prevent the reports from interfering with revision, the two deliverables, D3 and D4, would be submitted on the same day as the presentation.

### 1.3.12 Week Eleven

During week eleven the team spent most of their time testing, fixing bugs and preparing for the presentation. Some time was also spent on polishing the documents.

The group generally felt that the presentation went well, and as the project came to a close, everyone was satisfied with the progress that had been made.

## 1.4 Evaluation

As in every formal and professional project, evaluation is a very important process. Keeping an objective point of view, being able to recognize mistakes and suggesting new ways to approach previous issues is a vital part of learning. The team believes evaluation is a very positive and necessary task, therefore, when the project is close to reaching its end, the team members of this group self-evaluated and peer-reviewed their work and participation. Each team member reached similar conclusions about the project.

Firstly, as the project progressed the group began to realise the importance of thorough, methodical planning. A better planning process at the beginning of the project would have helped the group avoid stressful situations. The team’s project planning tended to be less than accurate in many areas; this was more due to inexperience in estimating time cost, than it was to do with a disregard, or unwillingness to create a plan. The miscalculations in the initial project plan substantial – For example, too many hours dedicated to programming, and not enough were allocated to testing.

Fortunately once the project had begun, the group realised these issues and set out to resolve them. New Gantt charts were produced at the middle of the project to reflect the current estimations for the time required to complete each task. These charts did not completely resolve the problem, and the estimations were still not perfect. Unforeseen circumstances such as very heavy load of coursework from other modules ended up getting in the way.

If the group repeated the project, it is agreed that they would put more effort into planning their time. It is believed that this, in combination with the experience gained from this project would have helped the entire process become more organised.

The other big aspect of the project that the team realised could have been done better is the initial design of the code. The group did plan at a high level how each subsystem would function, but not enough planning was done to fully understand how the various parts of the program would work together. The group found that when it came to joining the code together, it didn’t quite fit, and as such modifications had to be made to both parts of the system.

It was until these problems arose that the team fully understood the importance of good design. The group agrees that the interfaces for each class should have been thoroughly discussed, and then defined. The interfaces should have been designed to offer all the functionality that that classes need to offer, so would never need altering. Good interfaces represented on a good class diagram which has a good level of detail is very necessary in order to improve the (later) integration of modules of any project.

Had the design been performed to a lower level, more detailed design; for example having each method written in pseudo code before any coding began; the whole process would have been much simpler and less painful for each member of the group. The group believed that the work generated by not being specific enough in their design lead to inefficiencies in the programming stage, and therefore more time was spent than necessary.

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 understanding of what problem the team is dealing with is higher than what the team had initially thought. It is important that every team member is comfortable with every aspect of the problem so that during meetings and discussions every team member’s contribution becomes a solid, informed opinion about the matter.

In conclusion, this team believes that, despite any mistakes or errors, despite overlooked areas of work, despite any differences within the group, and despite any stressful situations that could have been generated; this project taught each of the members many important lessons. All member of this team will face similar situations throughout their career.

Each member learned the importance of many areas and stages in the development of a serious software product. The team knows that the experience gained through taking part in the project will be of great benefit in later life. The experience will help not only with future university projects, but also it will have a deep and beneficial impact on each team member’s careers as professional and proficient computer scientists or software engineers.

The team also made use of various software metrics including source lines of code (SLOC) and cyclomatic complexity to help understand their progress.

## 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

“The project was interesting. It allowed me to discover what to expect when working in a team, an experience not provided by other modules in the course so far. The task was an interesting one that allowed me to challenge myself, taking the visualisations and graphics as my role, an area that I felt weak in, so that I could develop that with the help of my other team members.

The project could have run smoother with more preparation and in hindsight we needed to do much more work on the design to allow our individual contributions to the programming to work more more seamlessly. There was a lot of need to ask other members on how they had implemented certain aspects as it was not readily available from the design.”

### 1.5.5 Kelvin

“I would say our teamwork and commitment is brilliant as our attendant rate of meetings is good, no one will miss any formal or even informal meetings unless they have got some urgent events to attend. What problems we faced in the progress of the project is we were too optimistic on our progress in the early stage like before the Easter. As after the Easter we discovered loads of misunderstanding, misinterpreting or inefficient codes that we thought the number of problems should be far less than we have got in the last two to three weeks of the project schedule. Despite of the misunderstanding, we work efficiently and with good communication in both facebook group discussion and meetings.

What I have learned from this module is to ensure any changes, even the minor one, should be reported to the group members afterwards or gain some comments from them before changing the codes as this may affect the efficiency of the code or even the structure of the system. Moreover is we should discuss more details on the model implementation before actually implementing the classes like the attributes of each class and the method headers as this could minimize the chance of misunderstanding and misinterpretations.”

# 2 Project Management Account

This section details how the Project was broken down into tasks, and how much time was given to each of these tasks. It then looks at how these estimates changed by the end of the project and the reasons for the changes before reviewing the whole process.

## 2.1 Gantt Charts

Gantt charts are useful for Project Management, they allow a team of people to organise and define tasks that a project can be broken down into. This is very helpful in managing a project and tracking progress and expected progress of the project.

## 2.1.3 Reflection

As can be seen from the two Gantt charts, start and end there was quite a change from the initial estimates to how the Project was actually completed. The initial Gantt chart was created by splitting the Project up into large tasks, the deliverables and then breaking these up into tasks that one or two people could tackle. The time estimations on the hand in dates for those deliverables and how hard it was expected each task to be. The final Gantt chart was based on the dates team members started on their tasks and when the final addition to that task was completed, these dates were obtained from the Time Expenditure Report.

The first thing to note as different was the final report. In the initial Gantt chart it was not split up into tasks, this was just due to the fact that it was so far into the project that it was not felt necessary to break up into tasks until the later stages, which was done as can be seen from the end Gantt chart.

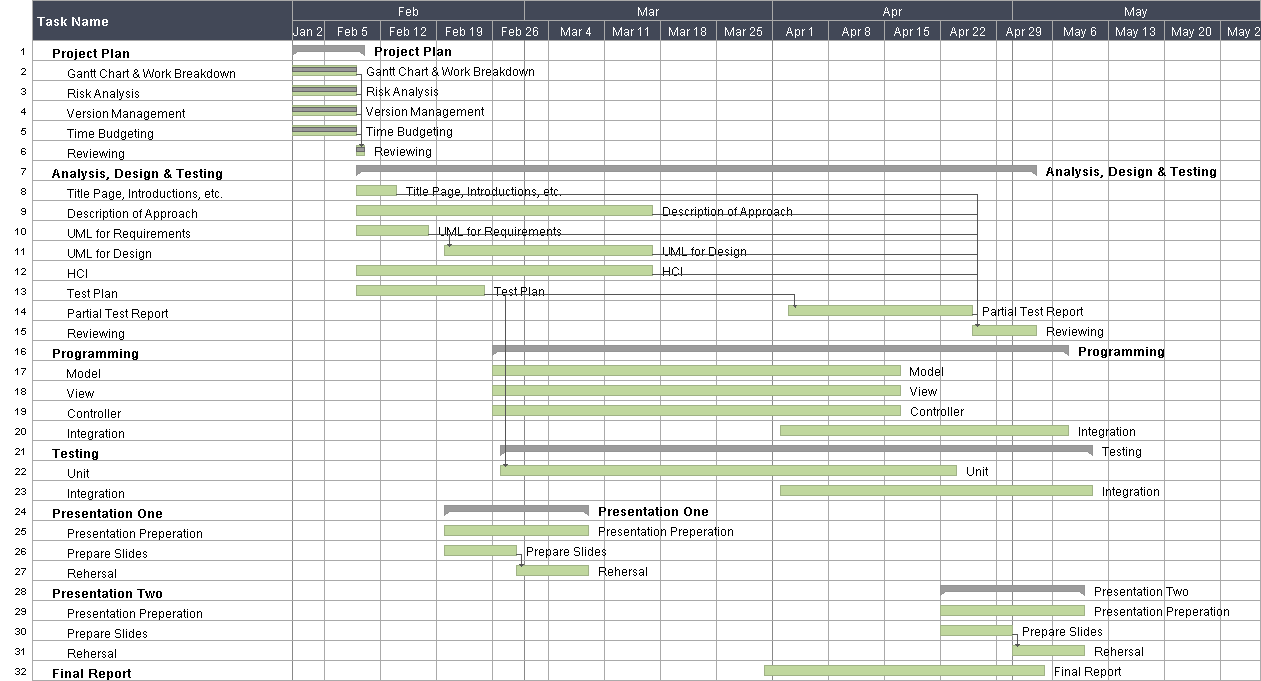
Another big difference is the lack of thought about Easter holidays. From March 16th to April 16th all members were on Easter break so no meetings were held and each member spent more time working on other projects than this one. In the initial Gantt chart this was not taken into consideration. This led to a lot of work in the programming and testing being scheduled for the Easter break, when very little work was actually done. The overall effect of this overestimation was that a lot of the programming and testing was pushed back until after Easter, causing all other tasks to lose time that was original allocated to them.

In hindsight the team may have been able to benefit from using a Gantt chart more effectively. If more time was spent allocating timeframes, taking into account little work would be done over Easter break then it would have been more evident that more work should have been done before Easter when the team had more time, so as to avoid the rush needed to catch up after Easter to meet deadlines.

The team member’s skills at estimating time expenditure has been improved due to the experience gained by collaborating on a large project and observing those areas of the project that are usually left to the last minute, such as testing, and which areas take a bit longer than expected due to unforeseen issues, such as implementation. Each member of the group now values the importance of Gantt Charts based and will be likely to use them in future projects.

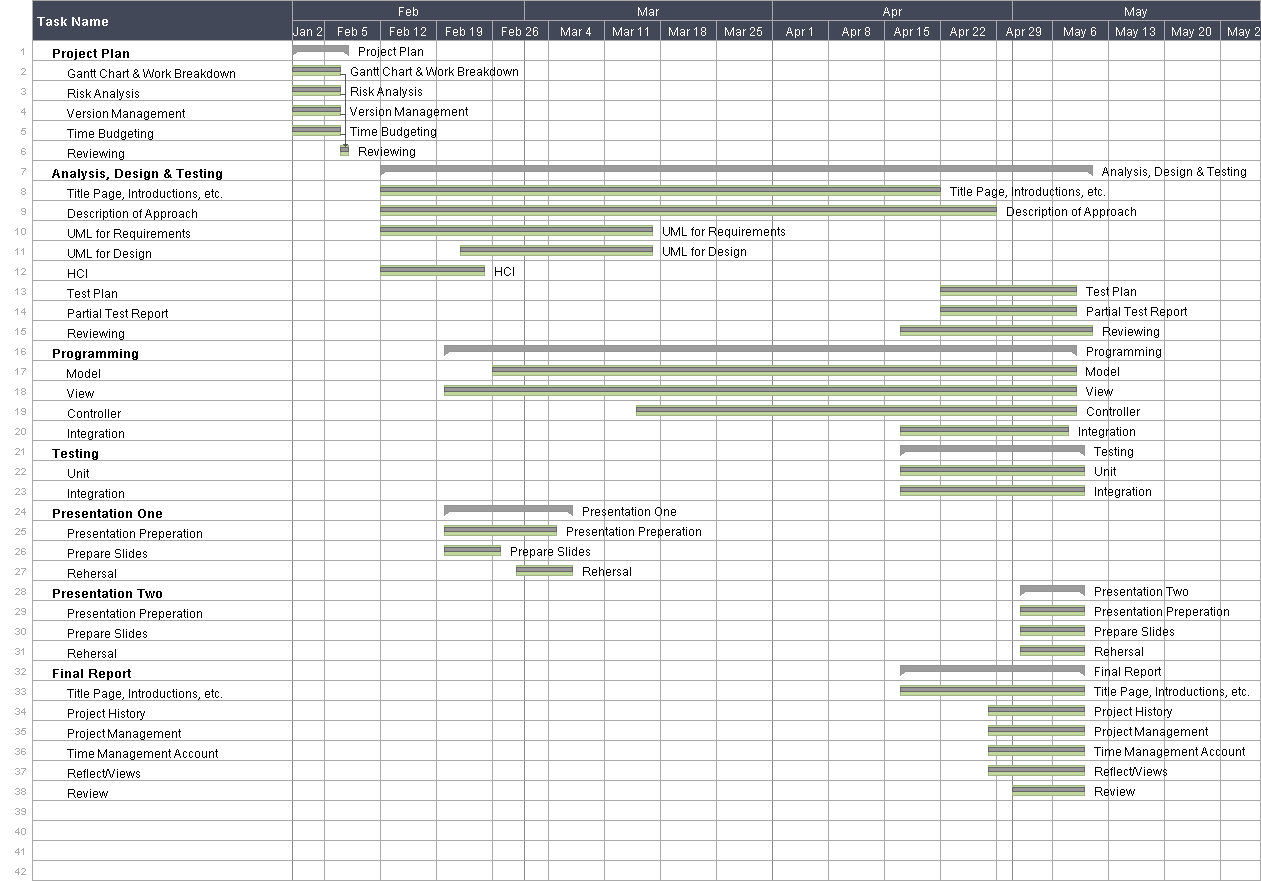
Overall the initial Gantt chart was too optimistic; the group planned to get things done a lot earlier than was actually possible. This is partly due to overlooking clashing dates and party due to underestimating time frames that certain tasks would take. There is no doubt that in future projects more time will be spent planning the project management to ensure a smoother process, however the project was well managed despite the underestimates.

### **2.1.1 Start Gantt**



Date Created 16/02/2012 for Project Plan

### **2.1.2 End Gantt**

  
Date Created 08/05/2012 for End of Project Report

# 3 Summary of Time Records

This table summarised the time spent on the project by each member of the group on a week by week basis. The tasks have been group together into their categories to save space and allow the table to fit within the page limit.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Process / Member | Edward | Kristian | Brian | Oscar | Kelvin |
|  | **Week 1 : 19.5 hours** | **5** | **5** | **3.5** | **3** | **3** |
| 1 | Project Plan : 19.5 hours | 5 | 5 | 3.5 | 3 | 3 |
|  | **Week 2 : 6.5 hours** | **1** | **1.5** | **1** | **2** | **1** |
| 7 | Analysis, Design and Testing : 6.5 hours | 1 | 1.5 | 1 | 2 | 1 |
|  | **Week 3 : 27.5 hours** | **7** | **3** | **2** | **10.25** | **5.25** |
| 7 | Analysis, Design and Testing : 17.75 hours | 5 | 3 | 2 | 5.5 | 2.25 |
| 16 | Programming : 9.75 hours | 2 |  |  | 4.75 | 3 |
|  | **Week 4 : 15.5 hours** | **1** | **7** | **1.5** | **2.5** | **3.5** |
| 7 | Analysis, Design and Testing : 2.5 hours | 1 |  | 1.5 |  |  |
| 16 | Programming : 13 hours |  | 7 |  | 2.5 | 3.5 |
|  | **Week 5 : 38.25 hours** | **11** | **7** | **6** | **7.75** | **6.5** |
| 7 | Analysis, Design and Testing : 2.5 hours | 2.5 |  |  |  |  |
| 16 | Programming : 5.75 hours | 2.5 | 1 |  | 1.75 | 0.5 |
| 24 | Presentation : 30 hours | 6 | 6 | 6 | 6 | 6 |
|  | **Week 6 : 6 hours** | **0** | **0** | **6** | **0** | **0** |
| 16 | Programming : 6 hours |  |  | 6 |  |  |
|  | **Week 7 : 9.5 hours** | **0** | **0** | **6** | **3.5** | **1** |
| 16 | Programming : 9.5 hours |  |  | 6 | 3.5 | 1 |
|  | **Week 8 : 32.25 hours** | **14.25** | **1** | **3** | **6** | **9** |
| 7 | Analysis, Design and Testing : 5 hours | 3 |  | 2 |  |  |
| 16 | Programming : 15.75 hours | 9.25 |  |  | 2.5 | 4 |
| 21 | Testing : 7 hours | 2 | 1 | 1 | 1 | 3 |
| 32 | Final Report : 6.5 hours |  |  |  | 2.5 | 4 |
|  | **Week 9 : 75.75 hours** | **22** | **16** | **12.5** | **9** | **16.25** |
| 7 | Analysis, Design and Testing : 7 hours | 5.5 |  | 1.5 |  |  |
| 16 | Programming : 43 hours | 15.25 | 15 | 5.5 | 2 | 5.25 |
| 21 | Testing : 6 hours |  |  |  | 2 | 4 |
| 32 | Final Report : 18.75 hours | 1.25 | 1 | 5.5 | 5 | 7 |
|  | Process / Member | Edward | Kristian | Brian | Oscar | Kelvin |
|  | **Week 10 : 110.75 hours** | **30.25** | **17.5** | **20.5** | **30.75** | **15.25** |
| 7 | Analysis, Design and Testing : 4.25 hours | 1 | 2.5 | 0.75 | 1 |  |
| 16 | Programming : 19 hours | 7.25 | 1 | 7 | 2.5 | 1.25 |
| 21 | Testing : 22.75 hours | 5 | 6 | 3.75 | 6 | 2 |
| 28 | Presentation : 30 hours | 6 | 6 | 6 | 6 | 6 |
| 32 | Final Report : 27.75 hours | 11 | 1 | 3 | 15.25 | 7 |
|  | **Subtotal** | 88.5 | 58 | 62 | 71.75 | 61.75 |
|  | Formal & informal meetings | 14 | 14 | 14 | 14 | 14 |
|  | **Total** | 105.5 | 72 | 76 | 88.75 | 75.75 |
|  | Breakdown | | | | | |
| **1** | **Project Plan : 19.5 h** | 5 | 5 | 3.5 | 3 | 3 |
| **7** | **Analysis, Design & Testing : 45.5 h** | 18 | 7 | 8.75 | 8.5 | 3.25 |
| **16** | **Programming : 122.75 hours** | 36.25 | 24 | 24.5 | 19.5 | 18.5 |
| **21** | **Testing : 35.75 h** | 7 | 7 | 4.75 | 9 | 9 |
| **24** | **First Presentation : 30 h** | 6 | 6 | 6 | 6 | 6 |
| **28** | **Second Presentation : 30 h** | 6 | 6 | 6 | 6 | 6 |
| **32** | **Final Report : 53 h** | 12.25 | 3 | 8.5 | 22.75 | 18 |

4 Time Management Account

4.1 Changes in Estimates

This table shows the difference between the estimate time expenditure and the actual hours spent on each category of tasks. It can be seen clearly that there are some areas where the actual time spent is very different from the estimate.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Activity Name | Estimation | Actual Spent | Edward | Kristian | Brian | Oscar | Kelvin |
| 1 | Project Plan | 18 | 19.5 | 5 | 5 | 3.5 | 3 | 3 |
| 7 | Analysis, Design and Testing | 38 | 46.5 | 19 | 7 | 8.75 | 8.5 | 3.25 |
| 16 | Programming | 180 | 122.75 | 36.25 | 24 | 24.5 | 19.5 | 18.5 |
| 21 | Testing | 40 | 36.75 | 7 | 7 | 4.75 | 9 | 9 |
| 24 | First Presentation | 30 | 30 | 6 | 6 | 6 | 6 | 6 |
| 28 | Second Presentation | 30 | 30 | 6 | 6 | 6 | 6 | 6 |
| 32 | Final Report | 50 | 64 | 12.75 | 3 | 8.5 | 22.75 | 17 |
|  | Formal & informal meeting | 0 | 70 | 5 | 5 | 3.5 | 3 | 3 |

At the beginning of the project more time was spent on the initial project plan than had been anticipated. This was because it was decided that the project needed thorough planning to ensure that it proceeded smoothly. The initial estimate for the time to complete the project plan was created before a substantial amount of time had been spent on the project, this meant that the team member’s lack of experience lead to and inaccurate estimation.

The team ended up spending a larger amount of time on the Analysis, Design and Testing document than had been planned in the initial estimates. One reason for the over spend in the area of the project, is that the original estimate was too low; the group didn’t possess enough experience to successfully estimate the amount of work that this would take.

The other main reason for this is that in the early stages of the project the group was eager to begin the programming stage, so tended to rush this document. When the group came to review the document later, it was felt that the quality of the document did not reflect fully the ability of the group. This lead to a lot of time being spent reviewing, and rewriting the document. The team is glad that this extra time was spent improving the document as it increased the understanding of the problem and the program’s design, making certain tasks easier in the long run.

The amount of time required for the programming was overestimated. Of the 180 hours allocated to programming, the group used just 127.75, only 68.2% of the original estimate. The group feels that there is a whole host of reasons for this underspend. Firstly the hours were allocated at a point in time when the group didn’t fully understand the project definition; the group didn’t fully understand the scope of the project, and as such decided to err on the side of caution when distributing the hours.

Secondly the group lacked experience in predicting how long large programming tasks would take to complete. The programming was completed at a much faster rate than the team had anticipated. The core functionality of the application was finished quickly as working code was needed for the presentation shortly before the Easter Break. The use of a GUI builder also helped the group cut down on the number of hours required for the programming. In general the hours saved were reallocated to other areas of the project.

Finally the group had been warned of issues that would arise during integration and planned for this accordingly. A small number of problems did occur at this stage, but far less time that had been planned was actually needed to resolve these issues.

Testing was another area where the team ended up under spending, although the underspend was not dramatic – 92% of the allocated time was used. The main reason for this underspend it not that time was not spent on testing, but instead that the time spent was not formalised, and as such many hours that the team members had spent were simply not recorded as testing.

Many of the hours that were recorded for the testing were actually spent documenting the testing. Since this task took almost as long as we had allocated for the testing, it could be argued that the estimate for testing is in fact slightly lower than it should be. Finally it is worth mentioning that regression testing times were almost completely removed since JUnit was used to automatically run regression tests.

The estimations for the End of Project Report were slightly under; this is for the same reasons as for the project plan. The group feels that the End of Project Report that it has produced is of a very high quality.

The group had managed to omit meetings, both formal and informal, from the initial time estimations. It turned out that a large amount of work relating to the planning of the project and the distribution of work took place in these meetings. The meetings allowed the group to track progress, make decisions and help each other.

It should be noted that the hours spent in formal meetings is the same among each member of the group. Since the presentations were both planned, and rehearsed during these group meetings - the hours spent by each member are identical.

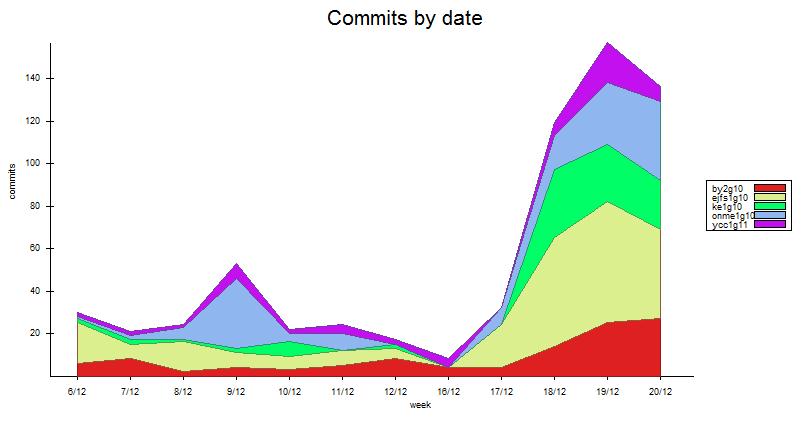
Overall, the group tended to over allocate time for the practical tasks, underestimating the amount of that the administrative and management tasks would take. Thanks to this project, each member now has a far better understanding of the importance of planning, documenting and evaluating the work that they do.

**4.2 Analysis**

Throughout the whole project the team members logged their time spent on each different tasks. The following section discusses and analyses the time spent by the group.

### 4.2.1 Distribution of work per tasks per member

From chart 4.2.1, you can see the time spent for most of the task are fairly distributed among the group members except for Edward who have paid a bit more than the others that is why he deserved to get 24% of mark distribution on this project. While Kelvin pay slightly less than the others in some tasks as he is the time manager while the time for managing the time expenditure form did not shown in the graph but you can see the time he paid in the final report to summarise the time expenditure and time management accounting. For testing as Kelvin and Oscar are responsible for the model classes which require more unit testing than the other classes so they paid most of the effort on testing. For Kristian, he paid a lot of effort on visualization of the system so he did not have much time to write for the final report and we decided to distribute most of the task in final project to the other group members. Brian provided some critical comments and worked for most of the task as he responsible for the controller classes which requires him to understand both model and view classes of how they works and integrate both of them into a single system.



**4.2.2**

By comparing the charts above, chart 4.2.2 and chart 4.2.3, the number of commits to the SVN is in proportion with working hours as the shape of the charts are quite similar which is quite delightful that the whole group pay around the same range of time before each commits. From the first few weeks we work on a average time of five hours and do six commits per week and we reach to a small peak on the fourth and fifth week as we are going to do the first demonstration so we make frequent changes and more effort within week four and five. After the motivation for the first demonstration we fail to keep it as Easter comes the work rate of the whole group drops gradually until the Easter finishes. After Easter finishes we try to get back on track with a slow start then a blast to the maximum of work rate in the last two weeks of the project schedule. But then the project deadline is postponed so we can still make some fine tunes on both our systems and documents so the work rate drops gradually in the last week of the schedule.

Again, obviously Edward has paid more effort and time on the project that is why his lines on both graphs are on top of other member’s line for most of the time. In the chart 4.2.2 shows Brian and Kelvin do not commit as much as the other members but when compare to chart 4.2.3 they actually pay decent time on each tasks which shows they try to do a bit more work before doing a commit to the SVN.

The last graph, chart 4.2.3, is showing the comparison between the actual time the whole group have spent throughout the project and the initial estimation of the time spending. Our group have tried to work above the expectation in order to finish as much as we can before Easter holiday. But after week 5, which is the start of Easter, we have stopped working at all and focus on other course works. We have nearly spent zero effort during Easter which cost us the result of the second half of the graph which is we cannot catch up with the schedule and the time we should pay for the project. As we did not spent any time for the project in the Easter holiday, the week 7 in the graph is actually week 11 which is the first week after Easter holiday. After coming back from the Easter holiday we start getting back on track and make our last effort on the project in the last two weeks of the project schedule so we have spent nearly 80 hours in week 9, actually week 13, to get our system ready for the demonstration and well prepared for the documents that we are going to hand in.

## 4.3 Reflection

When the project is close to be concluded, and the team is writing reports, evaluating its own work and seeking past mistakes or miscalculations, it is only logical and expected to reflect upon how the time was spent and managed. In fact, time is money and a project’s cost is usually directly proportional to the number of hours spent in it by each team member. Therefore, paying special attention to how the time was used in the process of planning, developing and testing this software product can be considered as paying attention to how the project’s cost and value was managed. To how efficiently the money was spent.

At an early stage of planning, this team implemented a system where a time manager was assigned to the project. One of the team members would be in charge of keeping track of how time was used. This facilitated reaching accurate conclusions. This team would like to conclude by stating that, this project, a learning project, was very beneficial in terms of experience. Each team member now understands how important it is to keep track of how time is being spent, how many hours are being dedicated to what and how that maps to what was planned. This team has now seen a clearer picture of the consequences of inaccurate time management and the consequences of getting away from the planned strategy for time expenditure. Every time anything like that happened, the pressure increased on every team member at a later stage. Luckily, this team is confident that in general, good track of the time spent was kept at all times. But most importantly, a very valuable lesson was learnt, to be thorough, efficient and strict when it comes to managing the time spent or to be spent on a project is one of the most important skills that need to be learnt and put into practice as soon as possible by any Computer Scientist or Software Engineer that intends to be a top professional in his area.

1. Peopleware: Productive Projects and teams. [↑](#footnote-ref-1)
2. Design Patterns: Elements of Reusable Object-Oriented Software [↑](#footnote-ref-2)