|  |  |  |
| --- | --- | --- |
| CS3343-LA3 2014/15 Sem A December 5, 2014 | B.T.  Self-Assessment Report | |
| [Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.] | | Prepared by  B.T. |

Contents

[Summary 2](#_Toc405488495)

[CHENG Suk Yin 3](#_Toc405488496)

[HUI Kwun Tung 4](#_Toc405488497)

[KWAN Tsz Ho 5](#_Toc405488498)

[LO Weng Kuai 6](#_Toc405488499)

[TUNG Leong Kok 7](#_Toc405488500)

[YAU Yet Chi 8](#_Toc405488501)

# Summary

Overall achievement

At first our project idea is to develop a program that can generate the best scheduled timetable for AIMS users. However, due to limited time we have to reduce the size of our project scope. Finally, we created the program that can automatically generate a random timetable instead, based on the preferences set by user. In release 3, it allows the user to configure some useful constraints – building, ‘pre-reg’ course and time constraint. We would like to implement more constraints for a more accurate timetable such as credit and optional courses at first. However we decided to implement the current constraints first as these can help to generate the usual timetable a student wanted – with half/full day off for their part-time job, avoiding classes with big time gap and classes with distanced classrooms (especially CMC classes for CS students). We also decide to provide a visualize timetable output for the user, as currently we are not able to view a tentative timetable with modification wanted by the user in AIMS. This feature will help the user to have a greater understanding of their choices.

Contribution of the team

From the project effort tracking worksheet, we can see that Donald our configuration manager, has a relatively higher time effort than other members. The reason is that he is good at programming and he contributed most of his time in our java program. However, this doesn’t mean that other of our members did not contribute to the project, they did spent their time on writing different reports (bug report, refactoring report, test report etc.), test code and do different testing. In further reading the tracking worksheet, some of the weeks are doing similar tasks. It is because all of our team members are having the placement job at the same time we are studying some courses in the school, we might not have enough time to just focus on the project when we have quizzes or assignments. We decided to distribute the tasks into weeks (for example week 7-9) as solution for creating a better product.

After week 9, we focused on doing the documentations rather than developing the program, as most of the functionalities have been developed already. As there are so many documentations and source code, our configuration manager has done his best to do the version control to have a better source control.

# CHENG Suk Yin

In this project, I mainly participate on the data work, such as prepare the data file, user interface and the report. My role is a normal member, helping to develop the main program and the report. Practically used the skill I learned in these course such as writing the test case, refractor the code, identify the bug, etc. Besides that, meetings are held frequently to keep track on the project, not just face to face but also the meetings in chat group.

The details of my contribution are shown as follow:

|  |  |
| --- | --- |
| Week 1 | Prepare course data.txt (1 hr)  Getting familiarized with the java (2 hrs) |
| Week 2 | Prepare the user interface (2 hrs)  Prepare the expected result output format (0.5 hrs)  Vertify the course data.txt (0.5 hrs) |
| Week 3 | Console output function (2.5 hrs)  Vertify the course data.txt (0.5 hrs) |
| Week 4 | Test code development (2 hrs)  Prepare the expected result of the test case (1 hrs) |
| Week 5 | Study the overall program (2 hrs) |
| Week 6 | Console the output function (2 hrs)  Test code development (1hrs) |
| Week 7 | Develop the main program  Prepare the readme.txt |
| Week 8 | Prepare the release summary (2 hrs)  Develop the test code (2 hrs) |
| Week 9 | Prepare the release summary (2 hrs)  Develop the test code (2 hrs) |
| Week 10 | Prepare the release summary (3 hrs)  Prepare the bug report (1 hrs) |
| Week 11 | Prepare the bug report (2 hrs)  Prepare the refractor report (2 hrs) |
| Week 12 | Prepare the self- sccessment report (3 hrs)  Prepare the readme.txt (1 hrs) |
| Week 13 | Presentation (5 hrs) |

# HUI Kwun Tung

In this project, my position is programmer. My duty is to design the code in order to achieve the functionality of user requirement. As a programmer, I have to design to most convenient way for user to assess this program. Therefore, my first step is to think about how to build the main function code and constraint function.

In code developing, I have used system development life cycle (SDLC) approach to make my work more efficient. I have generate some test cases to test reliability of the code. It helps to find out the bugs of the code so that the bugs can be fixed early. Once the new function was added into the program, some new test case was created to maintain the reliability of the whole program.

After all the functions is working well, I have helping to refactor the code in better way. As the code may not be object oriented, refactoring code is an essential action to increase the quality of the code. It allows the code to communicate its purpose and intention without the use of commenting.

As a programmer, I am good at coding. However, it is easy for me to waste too much time in coding many useless function. It caused by the miss understand of requirement. It is very important to declare the requirement first than starting the coding.

Moreover, my coding is very messy so that other programmers are difficult to follow up my coding. Therefore, it is very important for me to refactor my coding every time I finished one new function.

# KWAN Tsz Ho

# LO Weng Kuai

# TUNG Leong Kok

# YAU Yet Chi

In this project, I am the project manager and also contributed as the secondary programmer of the project. In teamwork terms, I helped to elaborate the main topic that raised by Donald (timetable scheduling system), provide more ideas on the program features like the constraint options and the visualized timetable, and completed the project plan. I did tried to divide the tasks to all the members to avoid unbalance workload on any single member, sadly I don’t think I did well in this issue.

I would consider myself as a normal member rather than a project manager in the project and actually it’s not good for the project management. In terms of software engineering skills, I helped to develop the main program at first, and also the helper functions like timetable output and validation of input data later. I also contributed in refactoring the program code in different stages. I tried my best to develop the code in good practices – with good use of indentation and comments for the ease of later modification and debugging.

For documentation, not only project plan but I also helped to write the release summary of the project. I also contributed to the bug report and refactor report. I would consider these documents as precise documentations, because our team members have good understanding of our project.

Here is my use of time spent throughout the project:

Week 1 file read IO function (1hr) project plan (3hr)

Week 2 Change 2Darray -> Map (2.5hr) Documentation version control (0.5hr)

Week 3 Console output function (4hr)

Week 4 helper functions (1hr) project plan (2hr) release summary (1hr)

Week 5 study the overall program (2hr) Console output function (2hr)

Week 6 Helper functions (2hr) project plan (2hr)

Week 7 main program (2hr) analysis and design report (2hr)

Week 8 refractoring the code (2hr) analysis and design report (2hr) project plan (3hr)

Week 9 refractoring the code (2hr) analysis and design report (2hr) project plan (3hr)

Week 10 bug report (1hr) refractoring the code (1hr) constraint function development (1hr)

analysis and design report (1hr) program input validation (3hr)

Week 11 refractor report (2hr) analysis and design report (1hr) program input validation (3hr)

Week 12 Self-assessment report (3hr) analysis and design report (1hr) bug report (2hr)

Week 13 Presentation (5hr)