City University of Hong Kong

Department of Computer Science

CS3343 (A) Software Engineering Practice

2014/15

Project Plan

Student Names	Student ID
HO, Wai Kit	53144248
WONG, Chung Man	53145233
YIU, Yiu Yeung	53144144
Kong , Tsz Kit	53143798
Lau, Kam Yu	53144170
So, Chun Hei	53144525

Table of Contents

1. Summary of the Project

1.1. Project description

We will apply Agile software development methodology in this project, this methodology will start the test stage before coding implementation on each phase to reduce the workload when the requirement is changing during the development phase. We will use this methodology to develop a system for calculating Bonus Miles (Frequent-flyer Program) for different airline company and different credit card. Frequent-flyer program is a type of virtual currency to purchase ticket, cabin upgrades, hotel stays or car rentals etc. We will introduce more about our project in this Project Plan.

1.2. Stakeholders

Stakeholder	Project Manager	Project Developer	Scrum Master
	Make sure the project team can meet the schedule and finish the project before deadline	To develop the project with their professional skill	To turn the requirement into the project, and make sure the requirement is implemented
Stakeholder	**Credit Card Company**	**Airline Company**	**Main User (customer)**
	Promote their credit card through the system and let customer gain bonus miles by using the credit card to buy the airline ticket	Let customer buy airline ticket or exchange the bonus miles to different services or airline ticket	Use the system to buy airline ticket and gain bonus miles

1.3. Objectives of the project

➢ To purchase airline tickets:

As an airline tickets purchasing system, a function for purchasing airline tickets is a general and mandatory function. The system read the file which included the flight (including direct and indirect of one way and round-trip flight) and the fee for doing to different destinations. Based on the requirements of the user, the system will filter the flights. After that, the system will read the quality of the tickets and the information of clients. Finally, the system will calculate the final amount of airline tickets.

➢ To calculate and use the bonus miles:

"Bonus miles" is a general bonus in the aviation industry. It is used to exchange free airline tickets, gifts or have a discount on purchasing airline tickets. It should be implemented in the system. The system will read the credit card information of the clients and what airline tickets they buy. After that, the system can calculate the bonus miles that will be earned in this flight based on different aviation company.

➢ To find out the shortest path from source to destination

"Time is money". Every customer wants to go to their destinations as soon as possible. So that our system can calculate the shortest path and the shortest time from source to destination. It is a useful function which all customers can be benefited.

➢ To find out the suitable flight for check-in baggage

Extra or special baggage is charged such as large baggage, pet and so on. A baggage fee calculator will be implemented. In addition, part of routes may not support some special types of baggage. All available routes which are able to transfer these baggage should be found by a function. It is also a pre-report function to get token for special baggage.

➢ To use credit card discount for purchasing airline tickets

In the system, the customers can use their credit card to get a discount for purchasing tickets. However, there are several type of credit cards and several airline companies. Therefore, each type of credit card has its own discount for each airline company. It is an attractive function of the system since customers can use the saved money to purchase more airline tickets.

2. Software Development Methodology

Agile software development methodology will be applied in this project. Waterfall development methodology is outdated and takes high risk for the software development since all the requirements must be collected and software design needs to be confirmed before the coding implementation. The biggest problem is that the client can't change their requirement during the development or software design can't be easily to change if there is problem existed.

The key of agile software development methodology is the "iteration" and "incremental". Agile promotes early delivery, continuous improvement and encourages rapid and flexible response to change. For each phase (iteration), our team will only focus some functions by the priority. And then our team will present it to the client, so agile can increase the communication chance with client. Any new or change of user requirements can be easy to achieve.

The 4 Agile manifestos that our team need to follow (left more than right):
Individuals and interactions over Processes and tools
Working software over Comprehensive documentation
Customer collaboration over Contract negotiation
Responding to change over following a plan

2.1. Project Team Organization

With applying the agile development methodology, the team should be self-organizing and cross-functional. Scrum is agile software development framework that our team can be used for managing the product development.

- **Scrum**

Scrum is a framework for our team to follow. The roles of the scrum team are scrum master, product owner and development team. Project manager and configuration manager will be assigned in this team.

Roles	Student Names
Project Manager, Development Team (Programmer)	**SO, Chun Hei**
Assistant Manager, Project Planning, Development Team (Programmer)	**WONG, Chung Man**
Scrum Master, Development Team (Programmer)	**HO, Wai Kit**
Configuration Manager, Development Team (Programmer)	**YIU, Yiu Yeung**
Development Team (Programmer)	**KONG, Tsz Kit**
Development Team (Programmer)	**LAU, Kam Yu**

- **Test-Driven Development (TDD)**

Our development team will work with test driven development (TDD) that the programmer will start the test cases before coding. The few steps should be following: write test cases, write code, debugging, conduct code and refactoring. It can make sure all the classes and methods are passed with a small piece of test cases. It's better than traditional testing method after the development, lower risk for the TDD since early testing.

- **Pair Programming**

In our coding development, our team uses pair programming that two programmers work together. One will be the code writer and the other once acts as the observer and switching the role after a period of time. It can increase the quality of the code and program structure with the idea coming up during the pair programming. Programmers can share the experience and promote some new methods to handle the case.

3. Development Tools

Software	Version	Description
Eclipse	Standard 4.4	Integrated development environment for Java. There are many plugin, which provide very rich function for eclipse.
GIT (GitHub)	N/A	GIT is a distributed version control system and source code management system to help in project development with speed and efficiency.
EGit	3.4.1.201406201815	EGit is an Eclipse plugin. Developer can use this plugin as a GIT client.
SourceTree	1.9.8	SourceTree is an Independent GIT Client for Windows and Mac. It is a very powerful GIT desktop client for developer.
JUnit	4.11.x	JAVA framework for unit testing
Mockito	1.9.5	Testing framework. By using this framework you can mock some components of the application.

4. Configuration Management Approach

In the project, Git, a distributed version control system and source code management system, is recommended to assist in our configuration management approach with speed and efficiency. In addition, GitHub will provide services for storing Git repositories.

4.1. URL of GitHub Repository

https://github.com/kitho/CS3343

4.2. Git User Account

Each team members will have their own Git account which stores their information to identify their identity, role and actions. This factor is useful for tracking change of the project and limiting access right.

4.3. Basic Flow of Commit and Retrieve

All the source code and the documents of the project which called software configuration items (SCI) are stored in a named repository (e.g. team name with project name) of Git with complete history and version-tracking function. Our members can check-out the SCI from the repository to their local workspaces to develop. The modified SCI can be committed to the targeted repository. In this process, the SCI will be given a new version number with the editor and record this process as a part of history for tracking and rollback. Thus that, we can track "who modified which version" easily.

4.4. Review Before Committing

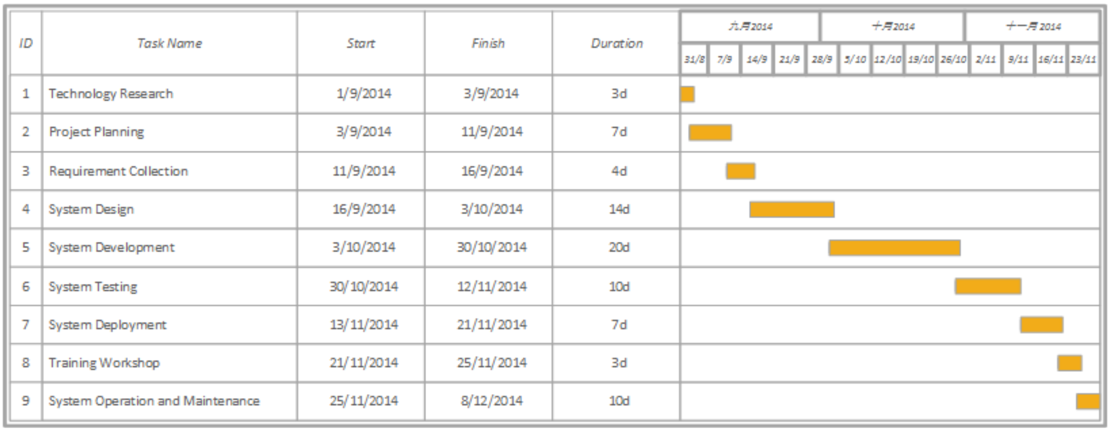
The SCI should be reviewed by Test Team to ensure that the modification is correct and grant authorization of Project Manager before committing. "Staging Area" is one of the ways to achieve it that an intermediate area where SCI can be reviewed before committing. The members can stage only part of a modified SCI. "Branching and Merging" is another way to achieve it that the team members have multiple local branches that can be work independently with creation, merging and deletion. By applying this feature, "Role-Based Codelines" will be performed. A branch that contains the SCI which will be launch on production while another branch will be created for testing and reviewing. Therefore, the Test Team and Project Manager can work in this two areas to confirm all changes of the project.

4.5. Further Functions of Git and GitHub

More functions can be supported by Git and GitHub to make our configuration management approach more convenient and efficiency. Multiple backups and different workflows will be experienced in this project development.

5. Work Breakdown Structure

5.1. Project Schedule



ID	Task Name	Start	Finish	Duration
1	Technology Research	1/9/2014	3/9/2014	3d
2	Project Planning	3/9/2014	11/9/2014	7d
3	Requirement Collection	11/9/2014	16/9/2014	4d
4	System Design	16/9/2014	3/10/2014	14d
5	System Development	3/10/2014	30/10/2014	20d
6	System Testing	30/10/2014	12/11/2014	10d
7	System Deployment	13/11/2014	21/11/2014	7d
8	Training Workshop	21/11/2014	25/11/2014	3d
9	System Operation and Maintenance	25/11/2014	8/12/2014	10d

5.2. Deliverables and their Date

Task No.	Deliverable(s)	Expected Completion Date
1	Detailed project plan	12-Sep-14
2	Prototype	30-Oct-14
3	Functional test report for prototype	12-Nov-14
4	Design review	14-Nov-14
5	Formal release for production	25-Nov-14

6. References

http://scrumreferencecard.com/scrum-reference-card/
http://www.sourcetreeapp.com/
http://www.eclipse.org/egit/
http://en.wikipedia.org/wiki/JUnit
http://en.wikipedia.org/wiki/Mockito
http://en.wikipedia.org/wiki/Mock_object
http://en.wikipedia.org/wiki/Cucumber_(software)

<<End>>