

# Post-Mortem Report

## 1. Project

### A. Description

- i. Project Name: Appointment Booking System
- ii. Client: RMIT-SEPT Assignment
- iii. Project Manager: Ran Lu
- iv. Solutions Architects: Yang Ding, Ziqi Liu
- v. Start Date: March 10, 2017
- vi. Completion Date: May 26, 2017

### B. Project Overview

- i. Discuss the project charter

The aim of this project is to provide a software that our client can use it to manage services, employees and creates bookings. It contains three kinds of accounts: A business owner can add services, customize layout, add employee, update business hours, view bookings and create bookings. A customer can view booking availability, create bookings and view employees' availabilities. A system super account can create business owner accounts.

- ii. What was the project success criterion?

This project need to implement all functionalities listed above and the program should be free of errors or bugs.

## 2. Performance

### A. Key Accomplishments

#### i. What went right?

All functionalities have been completed before the demo date and the program is robust and have no errors or bugs.

#### ii. What worked well?

All team members are cooperative and respond quickly.

The program has a nice graphic user interface.

The program is humanized and logical.

Acceptance and unit tests have shown that there're no compile, logic or runtime errors.

All the works are distributed well; each team member is able to do some parts of the work by their different abilities.

The weekly scrum meeting runs well, good communications between project owner and project members.

#### iii. What was found to be particularly useful?

Using Scene Builder to create fxml pages for building GUI saves lots of time.

Using Sqlite for database provides easy management for all information.

Advices from project owner is useful which can make sure the whole project in on track.

#### iv. Project highlights

The program allows co-existing businesses.

Functions used to check availability divide 24 hours into 48 time slots, so there will be no logic errors.

New functionality can be implemented easily by adding new controllers.

## B. Key Problem Areas

### i. What went wrong?

Nothing was going wrong about the project, we have met problems in coding, writing documentation etc. but everything was on track.

### ii. What project processes didn't work well?

Time management need to be improved, since sometimes we need to catch up with the schedule, such as completing new functionalities before next scrum meeting, finishing all GUI pages before the demo date.

Need more time to get familiar with the GUI and build a better interface.

### iii. What specific processes caused problems?

Spend lots of time on building GUI stuff.

### iv. What were the effects of key problems areas?

Sometimes need to catch up with the schedule.

### v. Technical challenges

Using Maven to include dependencies. In the beginning the program cannot be run in command line tool, but we managed to add plugins for executable jar and finally solved this problem.

This is the first time combine a database with Java, so we spent sometime to learn Sqlite in Java.

### C. Risk Management

- i. Project risks that have been mitigated: None
- ii. Outstanding project risks that need to be managed: None

### D. Overall Project Assessment

Performance against project goals/objectives	10
Performance against planned schedule	8
Performance against quality goals	10
Performance against planned budget	10
Adherence to scope	10
Project planning	8
Resource management	10
Project management	10
Development	10
Communication	10
Team cooperation	10
Project deliverable(s)	10

### E. Additional Comments:

- i. Other general comments about the project, project progress, etc.

### 3. Key Lessons Learned

#### A. Lessons Learned

Learn to set scrum meetings and record everything, using Github as source management tool and communicate with the product owner to make rapid changes about the project.

We also learned to record test cases using Lean Testing, which was efficient tools for bug reports and test suites.

Through weekly scrum meetings, we are able to complete a project on time and make sure the quality of project is high level.

#### B. Post Project Tasks/Future Considerations

##### i. Ongoing development and maintenance considerations

A better time management need to be considered and need to become more efficient with solving the problems in the product, e.g. when there is a problem in writing the code or other stuff, try to find out the solutions more efficiently instead of keep working on it for several hours, get some relax may be helpful instead of consistent work.

##### ii. What actions have yet to be completed and who is responsible for them?

None

##### iii. Is there anything still outstanding or that will take time to realize?

The whole project was completed with all functionalities and documentation.