COSC2299/2428 Software Engineering: Process and Tools
RMIT University

Major Assignment, Part B – Semester 1 2017
Due: 9AM Monday May 1 2017

Assignment Overview

This assignment continues building the Appointment Booking System application and turns it into a **Minimal Viable Product**. In particular, it continues the development of Part A.

This is a team assignment; Part B is worth 10% towards your final grade.
Note: there is a Part C to come: the original assignment outline stated that it would consist of Parts A and B, it has been split into 3 parts to facilitate management and more feedback.

In the Part B submission, team members will again be required to indicate how and how much each team member contributed to the project.
Marks for individual team members may be adjusted to reflect level and quality of contribution, as indicated by peer assessments and logs from collaboration tools.

Marks will be awarded for quality of the submitted system, quality of associated artefacts (e.g., test plans, test reports), and the proper use of the SE process and tools: a guide to the progressive weekly marks is below; a marking guide for the final solution will be published during semester.

Teamwork

You should continue in the same team as for Part A. If your team has an untenable working relationship, talk to your tutor ASAP.

Each team should discuss how to distribute tasks. **Different team members may contribute in different ways**: e.g., writing user stories; design and implementation; designing and running tests and bug reporting; writing documentation; etc; however, **all team members must make some contribution to coding/implementation**.
The tutors will facilitate weekly Scrum meetings in the weekly tute/lab sessions but it is recommended that each team appoint a Scrum Master who will be responsible for leading Scrum meetings and updating task plans in Trello or JIRA.

Academic Integrity

The submitted assignment must be your own team's work. No marks will be awarded for any parts which are not created by your team.
Plagiarism is treated very seriously at RMIT. Plagiarism includes copying code directly from other students (other than those in your team), internet or other resources ***without proper reference***. Sometimes, students study and work on assignments together and submit similar files which may be regarded as plagiarism. Please note that your team should always create its own assignment even if you have very similar ideas to other teams. Harsh penalties may be applied in cases of plagiarism.

What to do

Part B continues development of your existing system, resulting in a minimal viable product. During Part A, different teams have implemented different aspects of functionality, as directed by their Product Owner. For the Minimal Viable Product, it is expected that **all** the following functionalities are implemented; each team likely has at least one of the following already implemented:

1. The Business Owner can **add Employees and their working days/times**; an Employee can be added and their days/times be added or updated later. Employees and their days/times can also be loaded from a configuration file or database.
   As a **BONUS**, some Employees could perform only certain activities (see next spec).
2. The Business Owner OR a configuration file/database can be used to **define activities that take different times**. For example, a Hairdresser may allow bookings for Men's Haircut for 30 minutes or Hair Colouring for 1 hour. For simplicity, all activities can be multiples of (any) basic booking unit, e.g., multiples of 30 min slots, or even 1 hour slots if that's what you implemented in Part A. Adding such an option requires specifying activity name, duration, and other relevant info about it.
3. **A Customer must be able to create a booking; a Business Owner should be able to create a Booking for a Customer** (e.g., if the Customer phones in). A Booking should allow a specific Employee to be selected (if available at that time) and should be a for a particular activity if the full time required is available.

If you haven't already done so, **you must implement a Graphical User Interface**. If you are implementing a GUI during this phase, then you may need to postpone one of the above requirements until Part C: **this can be negotiated with your Product Owner**.
You should discuss the priority ordering of implementation with your Product Owner.

As in Part A, you must **provide User Stories and follow proper Scrum process** during Part B. Each of your sprints should produce a functioning extension to the system, as well as appropriate testing (including Unit Testing).

You should **checkpoint your Repository** to indicate that Part A is Version 0.1. The Minimal Viable Product (i.e., on completion of Part B) will be Version 1.0.

There will not be progress marks in this phase of the assignments but all t**eams are expected to demonstrate progress weekly and engage their tutor as Product Owner**. If you do not attend your weekly tutorial sessions then this will be taken into account if you encounter problems.

You should also update all documentation in Part B, including your Class Diagram and other design., installation and User Manuals, etc. **Code quality and clarity will again be marked** so make sure you properly document and lay out your code, avoid hardcoding, etc.
For Part B, **you must include a logging framework** and appropriate logging statements throughout your code.

More details and clarifications of Requirements will be posted to Blackboard.

Final Submission of Part B --- 10%

Part B of the assignment is to be submitted by **9AM Monday May 1 2017**.

Include a README file in your submission, containing:

- Names and student number for all team members;
- A short description of the contributions of each team member to the submission, including a % contribution of each team member (these should add up to 100%) and a statement of ***what*** part of the submission each team member was responsible for.

Other submission requirements will be as for Part A; details will be posted to Blackboard.

Marking criteria for the assignment will be discussed in class and published on Blackboard, but will include:

- how well your software meets the requirements at the demonstration;
- quality of the software produced, including quality of documentation/comments and adherence to programming standards;
- adherence to process and use of the planning and collaboration tools;
- how consistently you work throughout the project, based on weekly meetings/demos and details found in your code repository.

Marks for individual team members may be adjusted to reflect level and quality of contribution, as indicated by peer assessments and logs from collaboration tools.

All assignment **submission is electronic** – submission instructions, including **what to submit,** will be posted to Blackboard.
Assignments must be submitted by due date/time—**late assignments** may not be accepted at all.

You are required to manage your time, and required to work consistently until submission.
If you are ill, and are unable to progress on your work, you should submit a request for special consideration. We will then calculate the amount of time you were unable to progress on your work as a proportion of the time available for the assignment, as well as assessing the work you have submitted. In general teams should be able to cope with the temporary loss of a team member for a short time. If a team member is absent for any length of time, you should consult your lecturer or head tutor. We may then make an appropriate adjustment to your team's result. Illnesses of 1 or 2 days are unlikely to affect your work greatly, and are thus unlikely to be considered justifiable reasons for your work not progressing.
If you are ill and unable to work for either a major period (>50%) or the entire period of the assignment part, you should contact your student advisor to discuss your situation.

A consequence of not submitting an assignment/part on time is that your team may get 0 marks for that assignment/part.