# COSC2299/2428 Software Engineering: Process and Tools
# RMIT University
## Major Assignment, Part C – Semester 1 2017
## Final System Due: 9AM Monday May 22 2017
## Project-Completion Documents / Video due: 9AM Monday May 29 2017

## Assignment Overview
This assignment continues completes the Appointment Booking System application and produces some end-of-project artefacts. This builds on Parts A and B.

**This is a team assignment; Part C is worth 15% towards your final grade.**

In the Part C 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. Details of project post-mortem report and video presentation will be provided later. A marking guide for the final solution will be published during semester.

## Teamwork
You should continue in the same team as for Parts A and B. 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 C continues development of your existing system, and also produces project wrap-up documents.

As well as functionalities described in Parts A and B, add the following for Part C. **If you feel that this is not possible,** then discuss with your Product Owner: The more substantial marks are for **quality** of product and process, as well as for project-completion artefacts. You can release a high-quality MVP but NOT a poor-quality fully-functional implementation!

1. A Business Owner can create a new instance of the system for a business; they can:
   a) Provide the Business Name and details
   b) Provide Business Hours (e.g., Mon-Fri 9am-5pm; Sat 9am-1pm). Employee work times and Booking availabilities must then be within the Business Hours
   c) Define the services, either via a GUI page or uploading a file (**if not done in B**)
   d) Allow some customisation of the GUI pages: e.g. provide an image or at least the name/header on the page
   e) **It is not required,** but the challenge would be to allow multiple co-existing businesses (but note: this is not a requirement).

2. You should use a clear complex Design Pattern, either Factory or Facade (Singleton and Iterator don't count!). Indicate in your design document where this is used.

3. You must use Maven (for Java) or a similar tool for PHP to manage dependencies and provide automated builds. This must be able to build on another machine (windows, linux or unix) with minimal intervention. It must include running compiling, running tests, creating any artefact or installing into a folder as required.

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.

Other:

1. Checkpoint/tag your repository with the Part B version.
2. Your tutor will check your **repository use** in tutorials to **ensure proper workflow,** e.g., use of Branching.
3. More comprehensive Design/Architecture doc that includes some description beyond the class diagram
4. Complete Unit and especially Acceptance Testing, including history of running Acceptance Tests and any bug reports.
5. Clear and complete Installation/Build documentation, that lists all components and frameworks that need to installed.
6. Basic but clear User Manual (can be to same standard as Part A)
7. Release Notes (details to be discussed)
8. Project Post-Mortem (details to be discussed)
9. Project pitch video (details to be discussed)

**Some items will be marked during tutorials**, e.g., the repository setup and use of branching. You should update all documentation in Part C. **Code quality and clarity will again be marked** so make sure you properly document and lay out your code, avoid hardcoding, etc.

## Final Submission of Part C --- 15%

The System for Part C of the assignment is to be submitted by **9AM Monday May 22 2017**. Submission of other project artefacts (final documents etc) is due **9AM Monday May 29 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 posted to Blackboard, but will be similar to Parts A and B, along with the project-completion documents / artefacts.

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

- how well your software meets the requirements, as agreed with your Product Owner;
- quality of the software produced, including quality of documentation/comments and adherence to programming standards;
- the quality of product-associated artefacts (documentation, build scripts, project documents and video);
- adherence to process and use of the planning and collaboration tools.

**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.**