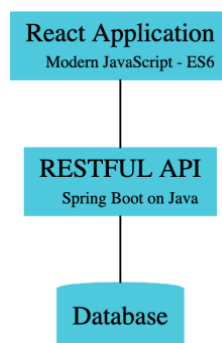


Software Engineering: Process and Tools  
COSC 2299 / 2428  
Major Assignment

<b>Assessment Type</b>	Group assignment. Groups will be created within tutorial groups and registered to Canvas, by teaching team. Submit online via Canvas → Assignments → Milestone 1/2/3. Marks awarded as per rubrics on Canvas. Clarifications/updates may be made via announcements/relevant discussion forums.
<b>Due Dates</b>	<b>Milestone 1:</b> End of Week 4, 8% <b>Milestones 2:</b> End of Week 8, 15% <b>Milestones 3:</b> End of Week 12, 13% <b>Progress Updates:</b> week 3, 5,6,7,9,10 and 11 (Each week 2%, in total 14%)
<b>Total Marks</b>	50

## 1 Overview

You are required to develop a full stack application matching the specification below, using **appropriate technologies, process and tools**, in particular those supporting collaborative project and code development. The program may be implemented in **Java** for the back-end using the **SprintBoot** framework and **JavaScript** for front-end using React. Guides on Getting Started using these frameworks will be provided for students not familiar with them.



(a) Full Stack Architecture

## **1.1 Purpose of the Assignment:**

You will simulate a Scrum software development team that has a contract with Bookeroo as a client. You will follow all SE principles throughout the production.

## **1.2 Project Description:**

BOOKEROO company requires a system to provide online book selling and sharing services for their users. BOOKEROO is not yet clear on all required features of the application; however, you can read their basic requirements below. An BOOKEROO Representative (your tutor act as product owner) will work with the team to check tasks priorities and review your promised work by the end of each sprint.

You are required to design, develop, test and deploy an application named "Online Book Sharing/Selling App". This is a web/mobile friendly application e-commerce that can be used by anyone, allowing users and publishers to sell new or used books. The application provides an open API for public websites, so the other websites can integrate it into their public website. Your application should have two sets of functionality for managing users: admin users and customer(buyer/seller or publisher) users, with different authorisation settings. Each type of user can see their own dashboard and functions.

This application would have common and main features and bonus features. You can discuss features with your PO (tutor).

For example, if a customer successfully logs in then the customer's profile dashboard will be displayed, allowing them to provide address, contact and other details. If the customer has previously had a transaction and their information has been stored, the application will allow them to retrieve that information. A customer should also be able to track their order status. Besides providing the required functionalities, your program should incorporate appropriate error handling, e.g., empty card, or if a book is not available.

You will produce this application over 3 milestones: Milestone 1, 2 and 3. You will apply Agile Scrum for your process management and at the end of each sprint you must have a ship-able functional product. A ship-able product is code that has been built, tested and deployed on a cloud server.

In each Milestone, you can implement as many features you promise/plan for each Sprint based on your Definition of Done. Definition of Done can be negotiated with your BOOKEROOrepresentative for each User-story per sprint separately and the rep will help prioritise your user-stories on your product backlog.

You will be assessed based on your process quality, your communication and teamwork internally and with BOOKEROO Representative (i.e., your tutor).

Implementation of your application should be incremental and via multiple iterations.

### 1.3 Group Work

This assignment is to be performed in groups of 3-5; these will be created by the teaching team. You will work as a Scrum team. Your group must be registered on Canvas, by end of Week 2. You must assign a Scrum master, Product Owner and other Scrum team roles. All members must be self organised and self-managed according to the Scrum process and all members must contribute equally throughout the project. Progress marking and Peer Reviews will be used to ensure equal contribution; Github and other logs will be used to resolve disputes.

If at any point you have problems working with your group, please inform your tutor or lecturer as soon as possible, so that any issues may be resolved. Regular progress updates should assist in communicating the progress of your group work.

Individual evaluations to the group project and to teamwork will be assessed by stating your members' contributions in a Peer Review.

Note also that there will be **Individual Progress Marks** assigned to team members for progress each week.

All team members must provide equal contribution towards the project in every possible aspects, such as documentation, implementation, integration and testing.

**Group meetings are compulsory.** To help manage your group work, and demonstrate that you are consistently contributing to your groups, you are required to use the following tools:

- Github (for source code sharing and management)
- Jira (for task tracking)
- Slack or MS Teams (for group discussion) (Messenger may also be used, so long as a log of communications can be captured and provided in milestone reports).

As well as the progress marks in labs, Individual Contributions will be included in a Peer Evaluation in each Milestone submission. Logs / analysis of tools use (e.g. Github contribution logs; Jira task logs) will be used to resolve disputes on Contribution, and to determine team progress and performance.

### 1.4 Relevant Lecture/Lab Material

To complete this assignment you will require skills and knowledge from lecturer videos and tutorial materials that would be shared as course material or some public programming tutorials on the web. You may find that you are unable to complete some of the activities until you have covered the relevant materials. However, you will be able to commence work on some sections. Note that the grade for your group work requires **consistent** work throughout all sessions. Thus, do the work you can initially, and continue to build in new features as you learn the relevant skills.

There will be materials/guide and help for PASS, Credit and Distinction grades of each milestone, however, there will be no help for the HD part of each milestone and it needs extra research, self-study and critical thinking.

## 1.5 Tools and Technologies

You will use multiple tools and technologies including:

- Code Repository: Github (each group will be assigned to their own private Github repository after forming)
- Task Management: Jira
- Group Communication: Microsoft Teams or Slack
- Documentation: Office 365 online
- CI/CD tool: CircleCI or Github Action
- Recommended hosting environment: Amazon AWS (VPS, EC2, RDS, S3, Cloud-watch)
- Java version: 1.8 or later
- Back-end framework: Spring Boot
- Front-end framework: ReactJS, latest version; talk to your tutor
- Build tool: Maven, most recent reliable version
- Data-store: RDS MySQL
- Unit testing framework: JUnit5 for Java, Jest or Mocha for Front end testing
- Other tools (more advanced) include Ansible and Docker for configuration management and Kubernetes for clustering

## 1.6 Progress/weekly assessment

An important aspect of this course is the learning and practice of modern Software Development techniques. Agile Software Engineering requires regular meetings, short sprints of development, regular commits (multiple per day) and maintaining a working product in the main branch, as well as testing-focused processes (e.g., Test Driven Development).

This assignment requires you to use an Agile development methodology (Scrum) and associated tools for each aspect of the process. You may use different tools in each milestone and gradually build and deploy your application on a cloud pipeline.

You must have frequent commits during each sprint, or whenever you add or change the code. There will be no progress mark if you add the whole code once before the due date in one large commit or push. You will receive Fail or minimum mark for the whole assignment if you won't follow the scrum implementation process (incremental and through multiple iterations).

Each week, your tutorial/lab session will involve a Scrum meeting, including a short (10-12 min) presentation to the “client” (i.e., your tutor). From Week 3, individual and group progress marks will be awarded. Only team members present at the team presentation qualify for the weekly progress mark(s).

On Github you will find startup code and a read-me file to help you get started with the assignment through Github classroom. You must clone that project and make your own project and upload that to your Github repository. This code includes:

- Base Structure of the full-stack framework
- read-me.

The use of the **start-up code** will be described in the Week 2 tutorial.

## 2 Learning Outcomes

This assessment relates to the following learning outcomes of the course:

- utilize various software engineering processes and their tools as required for best-practice development of software systems.
- cooperate in and contribute to a team environment, develop team dynamics, work according to an agreed team protocol, and resolve/manage conflict issues.
- run and document meetings.
- plan, identify and apply processes, standards and tools for phases of a software engineering life cycle for a substantial software development project.

## 3 Plagiarism

***NOTE: Plagiarism is a very serious offence.***

Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to exclusion from the University. Plagiarised material can be drawn from, and presented in, written, graphic and visual form, including electronic data and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited. Plagiarism includes the act of assisting or allowing another person to plagiarise or to copy my work.

The penalty for plagiarised assignment work may include reduction of marks awarded for that assignment, including to zero, or even failure of the course for a particularly serious case, for a first offence. Subsequent offences may involve heavier penalties, including the potential for expulsion from the university for repeat offences. Please bear in mind that RMIT University uses plagiarism detection software to detect plagiarism and that all assignments will be tested using this software.

If you have commit the whole assignment once and in a large commit that would be reported automatically for plagiarism and you will be interviewed by academic registrar.

See the RMIT website for more information about the university policies on Plagiarism and Academic Misconduct.

## 4 Project Details

### 4.1 Functional Requirements and the Main Scenario

The BOOKEROO Booking Application is an online booking web application; it includes multiple modules providing APIs (Micro Services) to allow integration into the overall application. You must describe functionality via user stories based on the main scenario.

Main Functional requirements:

1. The system should display a home page with functionality such as login/sign-up, about and contact-us. The program must authenticate and authorise users based on user type/role, i.e., admin or customer. You have three types of users:

admin user (business owner); publishers and shop owners, who can sell their own books, new books and used books; public users, who can buy, share or sell only used books.

User registration for Publishers/Shop owners and Customer: name, address, phone, username and password, ABN if it's a business and relevant information. Users can register themselves and the data must be saved to the database, however if the user is representing a business, the registration would be pending until admin user approve it. There might be some business rules for each user, (talk to your PO)

2. Admin user can login, and upon successful login, the admin is able to add/approve/block/edit a new users, add/edit books, look at the summary of past and current transactions (sorted date), and more, you may have specific admin features depends on your PO.
3. A Customer can search for books by name, author, ISBN and category.
4. Customers add review for a specific book that has been bought or specific user that had interaction with.
5. Customers/buyers must be able to pay for books or receive the fee through PayPal. For implementing this part, your application must interact with PayPal API.
6. Customers/Buyers must be able to cancel an orders until 2 hours after item has been ordered. In that case, a refund request must have been sent.
7. Customer/Shop owners must be able to see a history of their transaction and the status of current orders.
8. Users should be able to see the book cover image and table of content of the book(pdf), you can talk to your PO about the GUI of the pages/items.
9. Admin user should be able to download different sort of reports about books and user transactions in CSV format.

Your job will be to gather requirement, write user stories add all of them to your project backlog, break user stories into tasks, and add tasks to your sprint backlogs. You should plan for each sprint separately with your PO and implement highest priority user stories first. For any API you choose to implement, you must discuss proposed functionality and confirm your backlog with your product owner/tutor. (So different groups may implement different specific functionalities for the same module.)

## 4.2 Git workflow model

All groups must use Gitflow for project Git workflow as this assignment is scheduled as a release based project. For more information, see the Git lecture notes on Canvas.

## 5 Deliverables

### 5.1 Mandatory Requirements

As part of your project, you must:

- Use Scrum Agile as your project process methodology and sprint activities
- Attend daily Scrum standup meetings (at least two per week, one in your class and one on another day, this one can be online), also constant communication through slack or Teams.
- All meetings must be documented and submitted as part of Milestone submissions
- Include a refactoring period through your project (last sprint)
- You must contribute in all parts of the project as a scrum team member, including design, coding, documentation, management and so on.
- Implement based on standard coding style; your code must be efficient.
- Apply coherent and low coupled design and follow the MVC design pattern, using REST APIs and a micro-service design.
- Use Github for code management and follow the Gitflow organisation for your repository
- Have a shippable product at the end of each Sprint
- Use Jira for task management and always update your task board for task management and in-progress work. No task, no mark!

**If you fail to comply with these mandatory requirements, marks will be deducted.**

### 5.2 Milestone 1: Project Setup, User Stories, Product Backlog

For Milestone 1, you should:

- Write as many user stories as required by PO for all main features.
- Prepare your product backlog, including performing possible estimation, and prioritise your backlog based on meetings with the client/product owner
- Write test cases and Definition of Done (DOD) for each Product Backlog Item (PBI).
- Write your Sprint Planning notes for the first sprint (that will start at week 5)
- Breakdown user stories into tasks for the first sprint into sub-tasks and give them effort number
- Prepare the Sprint 1 backlog (tasks)
- Write Sprint Retro notes for Sprint 0
- Setup and start using Jira and add all product back log user stories there



- Setup Github for use for the rest of the project using base-code.
- Run the application for on every group member machine.
- Write unit tests for existing base-code .
- Write Acceptance Test based for all User Stories in the backlog.

### 5.3 Milestone 2: Main Construction Sprints, Initial Deployment

For Milestone 2, you should:

- Write any further user stories required
- Update your product backlog and prioritise your backlog based on meetings with the client/product owner
- Write all your test cases and DoD for each PBI.
- Continue performing Scrum meetings and documenting them, including Planning, Review, Retro meetings
- Write your Sprint Planning notes for all sprints.
- Write Sprint Retro notes including burn down charts
- Keep updating your Jira board for task management
- Continue using Github for the rest of the project.
- Implement functions that you have planned for sprint 2 and/or 3.
- Write unit tests for your main functions. Run all your Unit Acceptance tests and document results
- Add Automated Build and Continuous Integration using CircleCI (and possibly Ansible)
- Deploy your application to AWS
- Release a version on Github with Release Notes

Note that not ALL above tasks are absolute requirements for Milestone 3 but will be used to determine grade level for submissions: specific expectations for each grade are detailed in the rubric on Canvas

### 5.4 Milestone 3: Construction Completion, Deployment, Monitoring

For Milestone 3, you need to develop one or more enhancements to your program.

To obtain a higher grade you will need to implement one or more minor or major enhancements. Enhancements only count towards this Milestone if they are fully functional and error-free.

For Milestone 3, you should:

- Write all user stories for remaining sprints
- Continue performing Scrum meetings and documenting them, including Planning,



Review, Retro meetings

- Update your product backlog.
- Prioritise your backlog based on meetings with the client/product owner
- Write all your test cases and DOD for each PBI.
- Add all validation tests and document results of running them
- Implement your enhancement features
- Refactor your code and rerun all tests
- Add CD, using Jira
- Optional: use Kubernetes for clustering and add monitoring tools

### **Bonus Tasks:**

- User can swap books with each other, only if both have item that is searched by both parties. ( Match users that are looking for swaps)
- Integrate logging to the solution so any logs from the Kubernetes clusters are automatically stored in AWS CloudWatch for the future. Provide a screenshot of the BOOKEROO app logs being displayed in CloudWatch (Hint: use the querying function in Log Insights), as well as the code used to deploy the logging solution to the cluster. Add the screenshot to your Readme file.

Note that not ALL above tasks are absolute requirements for Milestone 3 but will be used to determine grade level for submissions: specific expectations for each grade are detailed in the rubric on Canvas

## **5.5 Presentation and Final Interview**

At the end of the project, your group will create a presentation that discusses your Application program and the process and tools used to develop it. In your presentation you should:

- Demonstrate your Application implementation and enhancements
- Discuss the project process model, Retro and the learning outcomes
- Discuss the tools that you have used

It is up to your group to decide how to best conduct this presentation. The purpose of the presentation is to demonstrate and convince the assessor of the quality of your group's software and overall work.

## **5.6 Weekly Progress Updates (from Week 3)**

Every weekly lab session, you will meet your tutor and provide a brief update to them (including the Scrum stand-up meetings that you have with them):

1. You can discuss your group's progress
2. Ask questions about your software design and implementation

3. Raise issues
4. Demonstrate your consistent and regular contribution to the project.

You will be assessed during the meeting by your tutor based on Scrum values and your contribution. Progress marks will be awarded from Week 3 onwards for Individual Progress, i.e., based on individual contributions from the previous week (based on the assigned task), your communication and involvement in meeting and as Scrum team member.

## 6 Submission Instructions

**Note that detailed Submission Instructions will be provided via Canvas prior to each due date.**

The assignment submission is divided into two components:

1. Your group's code submission & testing to the Assignment module (in <group\_id>.zip>)
2. Your group's report, as a PDF, to the separate Assignment report module (in <group id>.pdf)

Only **ONE** member of your group needs to submit the files. Canvas will automatically associate the submission with all members of your group.

If you are unable to upload your submission because you are not correctly associated with a group, please contact ***your tutor ASAP***. They will need to properly configure your group.

You may re-submit multiple times, however, only your latest submission will be marked. Be careful, re-submissions *after* the due-date are subject to late penalties.

### 6.1 Code & Testing submission

You must add your project private Github for the project folder, make a release for each milestone and rename it based on the instruction, then upload the URL through the Canvas submission for relevant milestone.

### 6.2 Report submission (.pdf)

Submit your group's report as a PDF file, using the separate Assignment Report submission module. **ONLY** a PDF submission format is permitted. Your PDF will be checked against the Turnitin software.

The PDF file must be named by your group id:

<group id>.pdf

**ONE** member of your group should submit this file via Canvas. This report must include all meeting minutes, user stories, UAT tests and test report, retro and planning notes, presentation, product backlog, screen shot of working application, Jira and group daily communication application, a link of your server and link of your Github.

Each member must report what they have done and address their work through tasks, code through Github (committed) and documents.

A detailed template has been provided via Canvas for the required contents of your report.

Rubric for each milestone will be published prior to the due date.