COSC2299
Software Engineering Process and Tools

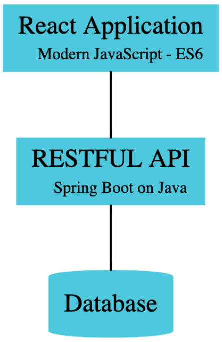
Assignment (v1.1)
RMIT Social Application

Weight: 40% of the final course mark
Group Registration: Week 1 and Week 2
Milestone 1 Due Date: End of Week 5 12.5%
Milestones 2 Due Date: End of Week 9 12.5%
Milestones 3 Due Date: End of Week 12 15% including 3% presentation
Progress Updates: During Sessions
Presentation & Marking: End of each milestone,
Learning Outcomes: This assignment contributes to CLOs: 1, 2, 3, 4, 5

1 Introduction

1.1 Summary

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 Sprint Boot framework and ***JavaScript*** for front-end using React. Guides on Getting Started using these frameworks will be provided for students not familiar with them. Variations of these frameworks are possible but MUST be discussed and agreed to by your tutor. In particular, note that some alternative approaches or frameworks will have technical limitations, so ensure you discuss and obtain agreement before making any change.



(a) Full Stack Architecture

Purpose of the application:

You are required to design, develop and test an api with front end of "RMIT Social Application". This is a social network application contains various components. Each component can work independently but must follow the design principles. Suggested APIs are available in Section 2.2 that as group you can pick an API and confirm the functionality of that API with your tutor.

Note that you are NOT expected to implement all components in the course of this Assignment!! Priority of components to implement is to be individually negotiated and agreed with your Client (your tutor), for each individual team. Each group should pick one component and start by writing a User Story and then working up a design. These must be discussed and agreed to with your tutor/client.

This assignment is divided into four delivery Milestones:

- **Milestone 1:** User Stories, Test Cases,Sprint planning, Product Backlog, Sprint 1 retro and burn-down chart, Shippable product with Milestone one required functions, Automatic build.
- **Milestone 2:** Unit Tests, Sprint planning, Sprint retro and burn-down chart, Shippable product with Milestone 2 requirements, CI.
- **Milestone 3:** Project burn-down chart, Improve UX, Refactoring and adding design patterns, Implement Enhancement features, Final product, CD.

In additional, you will provide regular updates on your progress in this assignment to your tutor during your classes.

1.2 Group Work

This assignment is to be performed in groups of 4 or 5. You will work as a Scrum team. Your group must be registered on Canvas, by Week 1 and Week 2. You must assign a Scrum master, Product Owner and other Scrum team roles; you must assign roles by Week 2. It would be advantageous to have a multi-skilled team. All members must be self organised and self managed according to Scrum process and all members must contribute equally through the project.

If you are unable to find a group, please discuss this with your tutor **as soon as possible and before** the group registration deadline. 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. The daily progress updates should assist in communicating the progress of your group work.

Individual evaluations to the group project and to teamwork will be be assessed using SparkPLUS, which assesses both contribution to outcome and to teamwork process (e.g., meeting attendance). **Individual marks may be adjusted upwards or downwards from the team mark based on Peer Evaluations entered using SparkPLUS.** Failure to use SparkPLUS may result in loss of marks for the team.

1.3 Relevant Lecture/Lab Material

To complete this assignment you will requires skills and knowledge from lecture and lab materials for Weeks 1 to 4 (inclusive). You may find that you will be unable to complete some of the activities until you have completed the relevant lab work. 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.

1.4 Tools and Technologies

You will use different tools and technologies including:

- Code Repository: Github (each group will be assigned to their own private Github repository in First Week)
- Task management: Trello
- Group Communication: Microsoft Teams
- Documenation: Office 365 online
- CI tool: Travis
- CD tool: Google Cloud
- Java version 1.8 and Spring Boot
- React newest version
- Build tool: Maven last reliable version
- Data-store: Google data-store
- Unit test: JUnit 5.

1.5 Start-Up Code

On Github you will find start-up code to help you get started with the assignment. You must clone that project and make your own project and upload that to your Github repository. This code includes:

- Base Structure of the application
- Examples of Unit Tests.

The use of the start-up code is described in the tutorial exercise for Week 2.

1.6 Plagiarism

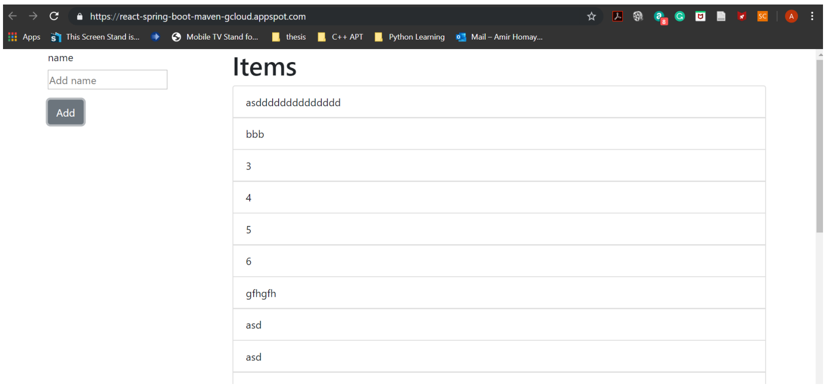
! ***Plagiarism is a very serious offence.***

A core learning outcome for this course is for you to: *Demonstrate and Adhere to the standards and practice of Professionalism and Ethics, such as described in the ACS Core Body of Knowledge (CBOK) for ICT Professionals.*

The penalty for plagiarised assignment work may include zero marks awarded for that assignment, or failure of the course, 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. See the RMIT website for more information about the university policies on Plagiarism and Academic Misconduct.

2 Startup Application and Functionality

The Startup `RMIT Social Application` program implements a basic Spring Boot / React sample application that includes a form to add items dynamically. This is a multi-module Spring Boot / React / Apache / Maven starter app with appropriate default settings. The React app is built using the command create-react-app.



name

Add name

Add

Items

asdddddddddddddd

bbb

3

4

5

6

gfhgfh

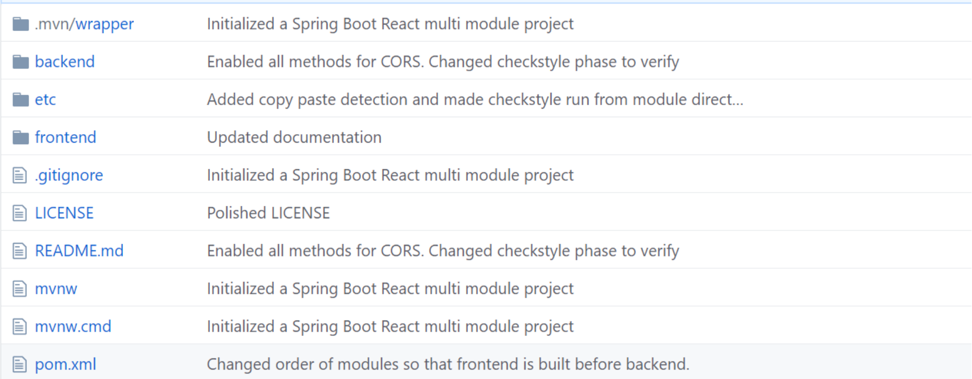
asd

asd

(a) Base Application

This project provides a setup for building Spring Boot / React applications. The application is divided into two Maven modules:

- Backend: This contains Java code of the application.
- Frontend: This contains all react JavaScript code of the application.



.mvn/wrapper	Initialized a Spring Boot React multi module project
backend	Enabled all methods for CORS. Changed checkstyle phase to verify
etc	Added copy paste detection and made checkstyle run from module direct...
frontend	Updated documentation
.gitignore	Initialized a Spring Boot React multi module project
LICENSE	Polished LICENSE
README.md	Enabled all methods for CORS. Changed checkstyle phase to verify
mvnw	Initialized a Spring Boot React multi module project
mvnw.cmd	Initialized a Spring Boot React multi module project
pom.xml	Changed order of modules so that frontend is built before backend.

(a) Base Application

> ! Aspects of this specification are flexible and open to your interpretation. In general, where there is flexibility, it is up to you to determine the best course of action. You may ask questions on the forum for clarity. You may also discuss with your tutor oMake sure that your tests are written to ensure your program works correctly based on any decisions you make.

2.1 Launch

- cd to project folder
- mvn clean install
- Change ITEMS_URL in ui/client/src/items/index.js to 'http://localhost:8080/items'
- install datastore emulator: Running the Cloud Datastore Emulator

```
$ gcloud components install cloud-datastore-emulator
```

- Command 'gcloud beta emulators datastore env-init' only print the variables to console. You need to manually execute them.

- Run 'gcloud beta emulators datastore start' in terminal to start the datastore emulator install datastore emulator: Running the Cloud Datastore Emulator

2.1.1 Manual Deployment

N/A, will be updated soon.

2.2 Main Functionality

2.2.1 Functional Requirements and the Main Scenario

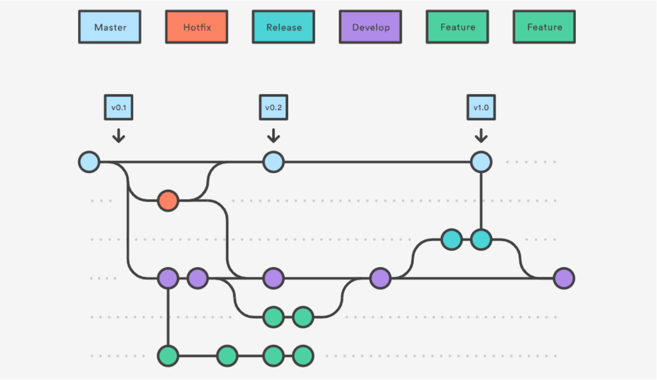
RMIT Social Application is a social network application; it includes multiple modules providing APIs to allow integration into the overal application. You must describe functionality via user stories based on the main scenario.

1. Central authentication API, and Search for Classmate API, Classmates list
2. RMITChat API and RMITWall API
3. Mentoring API
4. Buying and Selling API
5. Events API, Calendar API and Admin API
6. RMITWallet API
7. Student Live Stream API
8. Group Study API and Class/Course API
9. RM Startup API
10. RMIT Campus Map API
11. RMIT Until, QRCode and Help, Share location Student

You job will be to choose and prioritise APIs, write user stories, and add tasks to your backlogs. For any API you choose to implement, you must discuss proposed functionality and confirm your backlog with your product owner/tutor. (This means that different groups may implement different specific functionalities for the same module.)

2.3 Gitflow 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 in Canvas Module 1.



(a) Gitflow workflow

The overall flow of Gitflow is:

1. A develop branch is created from master
2. A release branch is created from develop
3. Feature branches are created from develop When a feature is complete it is merged into the develop branch
4. When the release branch is done it is merged into develop and master

5. If an issue in master is detected a hotfix branch is created from master Once the hotfix is complete it is merged to both develop and master

3 Deliverables

3.1 Mandatory Requirements

As part of your project, you must:

- Use Scrum Agile as your project process methodology
- Attend daily Scrum standup meetings (at least two per week, one in your class and one on another day)
- All meetings must be documented
- Have a refactoring period through your project
- Implement based on standard coding style; your code must be efficient. (An appendix will be provided soon for the standard coding and the style)
- Have coherent and low coupled design and follow the MVC design pattern
- Use Github for code management and follow the Gitflow organisation
- Have a shippable product at the end of each Sprint
- Use Trello and always update your Trello board for task management and in-progress work

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

3.2 Milestone 1:

For Milestone 1, you must:

- Write all user stories for the first sprint
- Prepare your product backlog.
- Prioritise your backlog based your meetings with the client/product owner
- Write all your test cases and Definition of Done (DOD) for each Product Backlog Item (PBI).
- Write your Sprint Planning notes for all Sprints.
- Write Sprint Retro notes including burndown charts
- Setup and start using Trello for task management
- Setup Github for use for the rest of the project.
- Implement functions that you have planned for Sprint 1.
- Write unit tests for your main function Run all your Unit Acceptance Test tests and document results
- Release a version on Github with Release Notes

3.3 Milestone 2:

For Milestone 2, you must:

- Write all user stories for the second sprint
- Update your product backlog.
- Prioritise your backlog based your meetings with the client/product owner
- Write all your test cases and DOD for each PBI.
- Write your Sprint Planning notes for all sprints.
- Write Sprint Retro notes including burn down charts
- Keep updating your Trello board for task management
- Continue working Github for the rest of the project.
- Implement functions that you have planned for sprint 2.
- Write unit tests for your main functions. Run all your Unit Acceptance tests and document results
- Add Automated Build and Continuous Integration
- Release a version on Github with Release Notes

3.4 Milestone 3:

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 must:

- Write all user stories for the second sprint
- 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.
- Write your Sprint Planning notes for all Sprints.
- Write Sprint Retro notes including burn down charts
- Add all validation tests and document results of running them
- Implement your enhancement features
- Refactor your code
- Add CD

3.5 Presentation

At the end of each milestone, your group will present and discuss your Application program with your tutor. In your presentation you should:

- Demonstrate your Application implementation and enhancements
- Discuss about the project process model, Retro and the learning outcomes, and your project burndown chart
- 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.

3.6 Class Progress Updates

Every session in the class, you will meet your tutor and provide a brief update to them (including the stand up meetings that you have with them). You can discuss:

1. 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 group

> ! One of the criteria in the marking rubric is for consistent and regular group work. These lab updates will help inform your final grade for your group work. **Individual Contributions will also be assessed using SparkPLUS, which may result in differences between team mark and individual marks.**

4 Enhancement Suggestions

You must implement at least two Minor and one Major enhancements.

4.1 Minor Enhancement

4.1.1 Simple suggestion

Whenever there is a user mouse over on an input, the program could display some text to help the user determine what commands they may perform.

4.1.2 Improved Feedback for Invalid Input

Whenever the user enters invalid input at a user prompt, the program should display a useful error message to explain why the input was invalid.

4.2 Major Enhancements

4.2.1 Docker

Integrate Docker for your development environment and deployment

4.2.2 Puppet

Add Puppet as your configuration management tool to eliminate manual works

4.2.3 Design Pattern

Adding an appropriate design pattern relevant to your UI design and/or implementation (other than MVC)

5 Getting Started

5.1 Managing Group Work and Communication

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, we recommend that you use the following tools:

- Git (for source code sharing and management)
- Trello (for allocating tasks)
- Slack/Facebook/WhatsApp (for group discussion)

These tools are not mandatory, but are highly recommended.

5.2 Designing your Software

This assignment requires you and your group to *design the ADTs and Software* to complete your implementation. It is up to your group to determine the "best" way to implementing the program.

6 Submission Instructions

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 (.zip)

You must zip your project folder, compressed it as zip file and rename it based on the instruction, then upload 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 to the ***Assignment 2 REPORT module***. 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, Trello and group daily communication application, a link of your server and link of your Github.