**Test plan for**

**<<project name>>**

***Note that you can refine your testing plan as the project development goes. Keep the change log as follow:***

*ChangeLog*

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| version number | Date of Change | Name of person who made changes | Description of the changes made |
| 1 | 11/3/2022 | Aaron Nguyen |  |
|  |  |  |  |

# Introduction

## Scope

Scope defines the features, functional or non-functional requirements of the software that **will be** tested.

As a group, we plan to gradually increase the scope over the span of the project. At first, wesimply do unit and integration test at the start. The target will be backend code and some of the frontend code. For backend code, we will test the API, token, database and app. For frontend code, we will test if it render correctly or is there any bug. After finishing unit and integration test, we will tackle acceptance testing.

## Roles and Responsibilities

Detail description of the Roles and responsibilities of different team members like. Note you only need to list the role you have in your team. There are some example roles.

* QA Analyst
* Test Manager
* Configuration Manager
* Developers
* Installation Team

Amongst others.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Net ID** | **GitHub username** | **Role** |
| Abu Sayeed Khan |  | sayeedkhannabil | Team leader + Integration backend-frontend |
| Aaron Nguyen |  | prosaox | Configruation + Test |
| Thomas Livingstone |  | thomaslivingstone | Frontend Developer |
| Hasin Ishrak |  | ishrakHP | Backend Developer |

# Test Methodology

## Test Levels

**Test Levels define the Types of Testing to be executed on the Application Under Test (AUT**). In this course, **unit testing, integraiton tesing, acceptance testing, regression testing, and laod testing** are mandatory. Please describe how will you do these testings. You may skipp load testing at this moment. Please revisit it after the related lecture is given.

Requirements:

* List the class/method/core feature you plan to test and how you could like test them and its acceptance criteria.
* For unit testing, at least 10 unit tests for each core feature to cover the code related to each core feature
* For integration testing, at leat 10 in total to cover core features.
* Acceptance testing for each core feature (if it is manual, need to list the steps)
* For regression testing, need to execute all above unit tests + integration tests you have for each commit pushed to main branch.

Unit testing and Integration testing:

* Connection to MongoDB
* Database functionality
* Server
* API routes
* Front end

Regression testing:

* Use Jenkins to execute all the tests when there is a change in main branch

Acceptance testing:

## Test Completeness

Criterias for test completeness:

* Whenever there is a pull request or push on the main branch, Jenkins should be called automately to start up, build and test the system. This means the unit and integration tests will be checked regularly to determine the functionality of the project as a whole. Moreover, this means bugs are more easily recognized.
* All the bugs and failed test should be checked and fixed regularly
* The test should cover 80%-90% of the codebase

# Resource & Environment Needs

## Testing Tools

The following tools are being used in this project for testing:

* Jest: Frontend and backend test
* Enzyme: React
* Chai: Backend

## Test Environment

It mentions the minimum **hardware** requirements that will be used to test the Application.

Example, following **software’s** are required in addition to client-specific software.

1. Windows 7 and above
2. Jenkins
3. Putty
4. AWS

# Terms/Acronyms

Make a mention of any terms or acronyms used in the project

| TERM/ACRONYM | DEFINITION |
| --- | --- |
| API | Application Program Interface |
| AUT | Application Under Test |