Our test plan is designed to allow developer detect defects and errors in order to determine the effort needed to validate the quality of the application under test.

Test Strategies

A. Testing Scope

We test our application during our development process by detecting issues in both small code segment level and functionality level. All of the code and features will be tested during this process.

B. Testing Approach

Our testing includes low-level tests and high-level tests. These tests are designed to detect error and defects due to bad logic, incorrect formula, typo and other incorrect behavior in development process.

1. Low-level tests

This strategy allows developers verify a small source code segment. Since the way our application designed is in a simple code and structure manner, our developers could inspect, maintain and test the small segment of code easily.

Developers may print out the result or the information they would like to monitor on the console to determine the operability of this segment of code.

2. High-level tests

High-level tests are conducted based on each features of our application. They will provide a guidance for the developers and a set of milestones for the manager. Our application is decomposable, which enables the developers to control the scope of the testing and detect problems

effectively. After developer complete the code for each feature, developer will test it on the following ways.

2.1 Backend developers

Backend developers are able to detect errors in data structures or external databases access or performance errors as below:

a. Test it via console by print the output or the error message
b. The functionality could be tested with the frontend code and test it on user interface. The interface error will tell the developer the related logic error in their code segment.

2.2 Frontend developer

a. Frontend developers could test the functionality on the browser. They can adjust the design of the user interface as needed.
b. They can also test the buttons or other click events with backend developer to check whether the correct requests are send to and handled properly.

3. Error handling

Our application is highly observable and easy for the user and developer to detect the error in our product.

3.1 Error handing message

Our developer will provide error messages if the users are not able to provide sufficient information during user interaction. This type of message will display on user's browser. For example, the error message will pop up if the user doesn't provide correct user name or password for login process.

3.2 System error handling message

If there is some system or processing issue occurred during the user interaction, the user will notice the error message on the screen. At this

circumstances, user might contact the administrator for further assistance. At this circumstances, an issue will be opened for the developer to fix. This part will be covered in our next part (error and defect management) of this test plan.

Error and Defect management

Our group uses GitHub to enable tracking a defect or error in development, testing and deployment phases of our product. Our defect lifecycle includes in the following stages.

1. A new defect or error is detected.
2. Tester can post this defect or error on GitHub as a new issue. Enough information should be covered in the issue description for the developer.
3. A triage meeting will be held at this moment to assign this issue to the proper developer. Furthermore, the priority (high, medium, low), severity(critical, major, minor, trivial) and risk(high, medium, low) of the fix will be determined based on the business perspective.
4. Assign this issue to the developer and choose the severity, risk and priority in Github issues.
5. Set the milestone on this issue.
6. Fixed:

Developer can mark this issue as fixed after the code review from other developers.

7. Retest:

Tester can retest the code and check whether it is fixed.

8. Reopen:

If the error still exist, tester can reopen this issue and the defect or error will go through the life cycle once again.