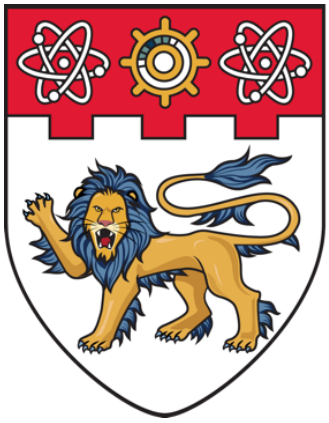

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Test Plan for TimeWise

Team 0
Lab group : SSP7

Group members	**Roles**
Mantri Raghav	Project Manager
Dwivedee Lakshyajeet	DEV Lead
Harding James	DEV Back-End
Alex Bernini	DEV Front-End
Mittal Madhav	Release Manager
Xue Xueting	QA Engineer
Koh Hui Ling	QA Engineer
Lek Jie Ling	QA Manager

Table of Contents

Test Plan Identifier - TW 1.1

This is the test plan on TimeWise, developed by Team 0. This is the first version of the test plan and it is at level 1 at this moment because when new features are included in it, the developers will enrich the test plan according to changes. This is a master plan for the entirety of the TimeWise project.

Revision	Description of Change	Change By	Date
1.0	Change formatting	Dwivedee Lakshyajeet	13/3/2020
1.1	Introduction to Approach	Hui Ling	15/3/2020
1.2	Pass/Fail Criteria to Testing Tasks	Xue Ting	16/3/2020
1.3	Small changes in introduction	Hui Ling	19/3/2020
1.4	Environmental Needs onwards	Raghav Mantri	19/3/2020

Introduction

The Test Plan is designed to prescribe the scope, approach, resources, and schedule of all testing activities of the project TimeWise. The plan identifies the items to be tested, the features to be tested, the types of testing to be performed, the personnel responsible for testing, the resources and schedule required to complete testing, and the risks associated with the plan.

This plan will cover the master testing plans. The test carried out will be Unit Testing, Integration Testing, System Testing and User Acceptance Testing. More specifically, we will be using a blax-box approach.

1. Unit Testing: Test cases created by the Quality Management team will be used to test each individual unit.
2. Integration Testing: Units which have been tested will be tested in a combined manner to expose faults in integration.
3. System Test: Independent external testers will be asked to test the entire system as a whole and their feedback will be taken.
4. User Acceptance Test: Prospective end-users will be asked to use the application and their feedback will be taken.

Test Items

1. For unit testing, the following black-box cases will be taken into account:
    a. AddButton.js
    b. MenuButton.js
    c. CompleteTask.js
    d. DeleteTask.js

The input and output will be clearly given to the testers. The test results will be analysed by both the QA as well as the development team.

2. For integration testing, the following black-box cases will be taken into account:
    a. Login.js
    b. TaskDetail.js
    c. AddTask.js
    d. EditTask.js
    e. DeleteTask.js
    f. RefreshSchedule.js
    g. Logout.js
    h. Notification.js

3. For system testing, the TimeWise APK will be given to external testers who will be asked to identify the following:
    a. Incorrect and erroneous functions
    b. Errors in the UI
    c. Behaviour or performance issues
    d. User Experience of the app

4. For the User Acceptance Testing phase, a beta version of the app will be launched and the users will be asked to give their feedback.

Features To Be Tested

Features will be rated based on their risk into three levels: **High, Medium, and Low**. High level risks have the highest importance and must be tested and debugged as soon as possible. Medium level features are to be tested only after high level bugs. Low level features are not as important and will only be tested once all other features are working properly

Item	Risk
Importing timetable upon login	High
Recommending time slots for tasks	Low
Creating a task	Medium
Edit a task	Medium
Delete a task	Medium
View task details	Low
Login/Logout	Medium
Refresh Schedule	Low

Features Not To Be Tested

All features will be tested.

Approach

Tests will be conducted per the documented test cases stored in the Test Summary Report. The QA engineers will create test runs. The tester will execute the tests in the Test Summary Report and mark each case as Pass/Fail. The tester should record down the actual results on the Test Summary Report.

When tests are marked as Fail, developers that are incharge of that particular feature should look into the errors and correct the system.

The QA Engineers will review the test runs in the Test Summary Report and report back to the QA Manager accordingly.

The test cases will be presented to the QA Manager before they are finally marked as passed.

When testing is deemed to be complete by the QA Manager, a test report will be submitted to the project manager who will mark the testing as complete if the application works as it should.

Item Pass/Fail Criteria

For each test case, the proper input and the proper output will be given to the tester. The test case will only be deemed to have passed if the result matches **exactly** with the ideal output defined in the test case. If the output is not exactly the same, the test case will be deemed to have failed.

If it is not clear whether the output matches the ideal result, the QA Manager will be consulted who will consult the Project Manager as necessary.

The completion criteria for this plan are:

1. 100% of unit testing cases are complete
2. 90% of integration testing cases are complete and 10% cases have minor defects
3. System Testing is done with 20 external testers
4. Atleast 100 users have given their feedback for User Acceptance Testing

Suspension Criteria And Resumption Requirements

1. Suspension Criteria:

If the team members report that there are 40% of test cases failed, suspend testing until the development team fixes all the failed case. The Lead Developer will be informed about the test cases which have failed.

2. Exit Criteria:

Specifies the criteria that denote a successful completion of a test phase - Run rate is mandatory to be 100% unless a clear reason is given. - Pass rate is 80%, achieving the pass rate is mandatory.

The testing will also need to be stopped in case Heroku or MLab server are down.

Test Deliverables

The deliverables included in this Test Plan are:

1. Test Cases
2. Test report
3. Revision Logs
4. Defect logs with solutions implemented

Testing Tasks

The following activities must be completed:

1. Test plan prepared
2. Items to be tested are identified
3. Identify method of conducting tests
4. Personnel assigned to each test case
5. Conduct testing
6. Fix bugs and errors which are discovered
7. Create defect logs
8. Create test report

Environmental Needs

The following environmental needs must be satisfied for testing:

- Test data populated using either the tester's NTU email or using special credentials provided by NTU
- Mocha and Chai installed on the tester's machine
- NPM installed on the user's machine
- Heroku and MLab deployments must be fully functioning

Responsibilities

Role	Responsibilities
Project Manager	Deciding which features should be tested Monitoring the errors found Collecting and analysing final testing report
DEV Lead	Providing required training in code details. Providing solutions for running the code Conducting white-box testing on the entire system
DEV Back-End	Conducting white-box testing on the back-end components
DEV Front-End	Conducting white-box testing on the front-end components
Release Manager	Collecting details of test runs to make sure software meets requirements before deployment
QA Manager	Stating the risk and contingency plan for the different phases of the test. Monitoring testing activities and ensuring all testing resources are available Setting overall testing strategy
QA Engineer	Conducting black-box testing Reporting test logs to the QA manager Generating additional test cases as needed

Staffing And Training Needs

Training will be required to setup the application on the testers' machines as well as to get acquainted with the JavaScript testing frameworks

The Lead Developer will be repsonsible for providing training in how to run the application locally and the structure of the code. The QA Engineers can consult the front-end and back-end developers if they run into any problems.

The QA Manager will be responsible for finding methods to train the testers on the JavaScript testing frameworks such as Mocha and Chai.

Schedule

The testing phase for TimeWise will last for upto 1 month, the details of which are given in the Project Plan. Our planned activities will be as follows:

- Number of days assigned to each test case will depend on their risk level. High risk will get 5 days, medium risk will get 4 days and low risk will get 3 days. This is so that not too much time is wasted on testing low risk items.
- A daily backlog will be maintained to keep track of items which are yet to be completed. The QA Manager will continuously monitor this backlog and will help the QA Engineers if the backlog gets too large.
- For any bugs that are discovered, the result will be immediately updated on the GitHub Projects Board and the Lead Developer will be notified of the same immediately.
- A one week buffer is taken in the plan in case there aren't enough users for User Acceptance Testing

Risks And Contingencies

The risks and methods to mitigate them are as follows:

Risk	Contingency
Shortage of testers	Members of the development team as well as management will pitch in to help with the testing if the QA team gets overwhelmed
Improper communication amongst testers and developers	The daily testing backlog will ensure testing is on track. Testers will be required to report any bugs they discover immediately to the Lead Developer
Improper training for testers	The Lead Developer and QA Manager will prvide all required training

Approvals

For each testing type, different personnel will be respnsible for providing approval:

Role	Test Type	Approval Criteria
Lead Developer	White Box	System performs in accordance with functional requirements
Release Manager	Black Box	User Acceptance Testing meets acceptance criteria
QA Manager	Black Box	All test cases are covered
Project Manager	Overall	The application works as required and user acceptance testing meets client needs