Test Plan Document
Team 3.02

1 Introduction

Do-D-Due is an application that allows a user to manage their daily tasks. This application allows the user to create a task list, edit and delete it via the Android application or the web interface. The tasks will have a name, description, due date and priority. The user will be able to check off items in the list and will be able to hide them. The application will support multiple users, each one with their own lists.

2 Quality Control

2.1 Test Plan Quality

The quality of the testing process will be ensured by independent testing from the testers. The results will be consolidated at the end and re-testing will be done for conflicting results. The selection of our test cases are driven from the requirements as well as cases that the developers and testers find most likely to cause a problem. This will ensure that the requirements are met and that the system will perform as expected. The value of the test plan can be evaluated by measuring the number of defects found at the end that were not covered by the test plan. In addition, it can be further measured by the number of defects prevented by our test cases.

2.2 Adequacy Criterion

Testing will conclude when all the requirements are satisfied and the appropriate bug fixes for are completed. Testing may be extended when more features or requirements are desired during customer feedback.

2.3 Bug Tracking

A shared Google Doc Spreadsheet will be used to track bug id, information, severity, and status. Team members will reach a consensus before finalizing the status of the bug.

3 Test Strategy

The Android code and web application code developed will be subject to a number of rigorous testing strategies involving unit, integration, system and regression testing.

3.1 Testing Process

The testing processes mentioned above will be performed collaboratively by the team members:

Jack Suen(JS), Niranjanaa Ragupathy(NR), Archana Shree Vaithiya Santharaman (AS), Sanchita Vijayvargiya (SV).

Testing Activity	Performer
Unit Testing	JS, NR, AS, SV
Integration Testing	JS, NR, AS
System Testing	JS, AS
Regression Testing	SV, NR

3.2 Technology

For unit testing of the Android code, JUnit testing technology will be utilized.

4 MTM and WTM Test Cases

Test ID	Test Case	Test Medium	Expected Result	Actual Result	Pass/ Fail
1	Ensure user registration completes	Both	User details stored in database		P/P
2	Ensure functionality of user authentication	Both	Log-in access only on credential verification		P/P
3	Ensure task creation	Both	Creation of a new task without errors		P/P
4	Check if default priority, due date and time values are enforced	Both	Default values assigned		P/P
5	Ensure user is logged in until he/she logs out	Both	User session maintained		P/P
6	Creation of more than one user account	Both	Multiple accounts registered		P/P
7	Session termination from any screen	Both	Successful log-out from any screen		P/P
8	Modify existing tasks	Both	Successful change in task name, due date, time, priority or note		P/P
9	Ensure user is able to exit from app	MTM	App exits to the Android home screen		P/P

10	Ensure app starts on the most recently viewed screen when reopened	MTM	Returns to screen user was viewing when app exited; remains logged-in		P/P
11	Ensure app automatically saves lists after creation/ modification	MTM	After adding/modifying an item, even if the app closes, the list should show modified results correctly		P/P
12	Ensure the user will have to confirm all deletions	Both	When the user clicks the Delete button, a pop up should prompt the user to confirm		P/P
13	Ensure user is able to delete task	Both	Correctly deletes task after confirmation		P/P
14	Ensure user is able to filter out completed tasks	Both	When user clicks the “Hide completed” button, all completed tasks should be removed from view		P/P
15	Ensure user is able to edit/delete task screen	Both	After a long press/right click, task edit/delete menu should appear		P/P
16	Ensure the app automatically syncs to CDB upon logging in	MTM	If there are new items on LDB, CDB should updat after logging in. Nothing should happen otherwise		P/P
17	Ensure user can explicitly sync	MTM	If there are new items on LDB, CDB should update after logging in. Nothing should happen otherwise		P/P
18	User presses save after modifying list	WTM	List is properly saved into the CDB		P/P

19	Put in past date	Both	Should not allow		F/F

5 Integrated Test Cases

TestID	Test Case	Expected Result	Actual Result	Pass/ Fail
1	No new updates	Neither WTM or MTM should have any updates to list		P
2	Update on MTM, auto-synced	WTM should receive updated lists		P
3	Update of MTM, explicitly synced	WTM should receive updated lists		P
4	Update of WTM, MTM logs in causing auto-sync	MTM should receive updated lists		P
5	Update of WTM, MTM explicitly syncs	MTM should receive updated lists		P
6	Update of WTM and MTM, MTM syncs (either explicit or automatic)	If updated lists are the same, do nothing. If not, save a copy of both versions “Conflict - ” prefixed.	No longer doing this, all conflicts will be resolved through time	F