# To-Do List

## Test Plan Document
### Project 2, Team 6

## Introduction

The software Product which we are working on is titled "To-Do List for Android".
The To-do List allows users to manage tasks that they have to accomplish. The user can add tasks, set the priorities of each task, set the due date for each task, check-off items in the list and hide/show the checked items. The application will support multiple users and will be developed using Android Developer Tools.
This document gives a brief about our testing strategy for the project. Its main objective is to set different quality standards for the unit, integration and system testing of the specified application.

## Quality Control

### Test Plan Quality

The main objective of this activity is to detect and fix as many bugs or defects in the code as possible. This will in turn help in improving the quality and reliability of the program code. The more rigorous the Test Strategy is, the more chances there are of getting a bug free program.
We will try to ensure the most thorough and efficient Test Plan for our project.

### Adequacy criterion

The Adequacy Criterion depends on the extent to which the test cases made are successful in debugging the program code. Hence we need to set the range of the test data in such a way that most of the bugs get detected and fixed in this phase.
Our tests are exhaustive and based both on extreme and common cases, that's why it will ensure an adequate software code delivered to the customer.

### Bug Tracking

The bugs found by implementing the test cases need to be documented and fixed properly. The Testing phase enables the testers to put in any random value and record the case if we get any erroneous or strange value as the output. This case is then recorded and reported back to the development team to get permanently fixed. The bugs found are stored for reporting and reference usage.

## Test Strategy

Test Strategy is the plan by which we will analyse our program code and detect the differences between existing and required conditions. This strategy should be made in a way so that the gap between what exists and what is desired is minimized. We are carrying out two basic Test activities for our project – To-Do List.

**Unit Testing**- Each software module is tested internally at the development phase itself and all possibilities of internal bugs at each unit are removed.

**Integration Testing**- Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests to those aggregates, and delivers as its output the integrated system ready for system testing.

**System Testing**- The system once integrated is tested as a whole and the output is checked for any anomalies.

**Regression Testing**- Once bugs are fixed, regression testing is conducted to verify whether fix is correct and if it causes new bugs.

## Testing Process

The testing process will consist of these procedural steps:

**Procedural Steps**

- As a part of **Unit testing**, we will test the individual modules of the program like Signup, Login, Add/Edit/Delete, Sort/View, Hide/Show modules.
- As a part of **Integration testing**, we plan to use the big bang approach to integrate all modules for testing as the application is not very complex.
- **System testing** will be conducted based on the system requirement to test the system as a whole.
- In **Regression testing**, we will execute the failed test cases to verify whether the fixes which have been implemented are correct

## Technology

Since it's a code being implemented using the Evolutionary Prototype Model, there's a specific testing tool that we are using for testing purposes named JUnit. The Test Cases are being implemented in the same way as any Customer would try to use the App in his Android phone and exercise its in-built features.

# Test Cases

The Test Cases that we formulate to test the code forms the basis of our testing process. It also shows us how strongly the Code can withstand against any arbitrary or unexpected input values without crashing. The Test Cases that we plan to execute include:

| Test Case# | Purpose | Steps | Expected Result | Actual Result | Pass/ Fail |
|---|---|---|---|---|---|
| 01 | Test whether the app can identify a pair of username / password which is already signed up on the app. | Try to login using a valid username / password pair | Should login correctly. | | |
| 02 | Test whether the program identify an incorrect username / password pair | Try to login using an incorrect username/ password pair. | Should not login and prompt the user for correct entry. | | |
| 03 | Test whether the program can identify an empty field in the username / password field. | Try to login using an empty field in either username / password. | Should prompt the user for an empty field. | | |
| 04 | Test whether the user is able to check / uncheck the check box in the list view | Try checking / unchecking the checkbox in the main list view. | Should check / uncheck correctly. | | |

| | | | | |
|---|---|---|---|---|
| **05** | Test whether the user is able to scroll down the list without any errors. | Try scrolling down in the task list view. | Should scroll without any errors. | |
| **06** | Test whether the user is able to add a task. | Try adding a task from the main page. | Should add a task in the tasks list view. | |
| **07** | Test whether the user is able to change the details for a particular task. | Go to the task details of a particular task, and then try changing the details. | Should be able to edit without any errors. | |
| **08** | Test whether the user is able to change the priority of a particular task | Click on edit task and try changing the priority. | Should be able to change the priority. | |
| **09** | Test that if the user gets a phone call while using the app, then app is restored after the phone call in the same state as before the call. | Try calling the cell phone on which the app is installed and running and check for the behaviour. | Should be running smoothly and in the same state after the call ends. | |
| **10** | Test whether the user is able to change the due date of a particular task. | Try changing the due date of a task. | Due date should change without any errors. | |
| **11** | Test whether new users can sign up correctly | Try signing up with expected format of username and password | Should sign up successfully | |

| 12 | Test whether the program validates the input for signing up | Try signing up with empty field in username/password | Should instruct users to give a correct username | | |
| --- | --- | --- | --- | --- | --- |