# To-Do List

## Requirements Document

Project 2, Team 6

## Introduction

This document provides a requirement description about the software product 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 off items. The application will support multiple users and will be developed using Android Developer Tools. This document will serve as a reference for both the customer and the developers.

## User Requirements

This specifies the requirements which the user expects from the software Product.

### Software Interfaces

The application will be working on an Android-powered mobile devices. (Smart phones or a tablet computers).

### User Interfaces

The application will provide a graphical user interface for users to interact with the system. Users must be able to do all operations with graphical user interface. The user interface is supposed to be intuitive and give useful instructions to make the system easy to use. The user interface must be responsive so that the user does not have to wait after performing some functionality.

### User Characteristics

The user should be able to use an Android device.

### Assumptions and Dependencies

The following are the assumptions and dependencies that affect the application

- Android system is properly installed on users’ device as this is an Android specific product.
- This product will only run on Android powered devices.

## System Requirements

This lists down both the functional and non-functional requirements for the Product.

### Functional Requirements

- The application must support multiple users, and each user must have their own To-Do list. User information can be stored in a built-in database.
- The application must allow users to add, delete and edit the items in the list.
- The application must allow users to specify item name, priority, and due date when users add an item to the list.

- There must be a default value for priority and due date, so that users won't have to necessarily specify them.
- The application must save lists automatically and immediately after they are modified.
- Users must be able to set the priority using dropdown lists with options of High, Medium and Low.
- Users must be able to set the due date by either selecting a date from a calendar or inputting a date in given format.
- The application must ask users to confirm when items are deleted from the list.
- The application must be able to check off an item from the list when users accomplish it without deleting it.
- The application must be able to show/hide checked-off items.
- The application must provide detailed information when users click on a certain item.
- The application must provide reminder some time before a task is due. The time can either be specified by users or by default.
- The application must remind users to check off an item or reset it if it is not checked of manually by the due date.
- The user must be able to sort the items in the list by priority or by due date.
- The application must display user-friendly error message when users give ill-formed input (e.g.: The wrong format of due date).

## Non-Functional Requirements

**Software Quality Attributes**

The Software Quality Attributes includes

- Robustness & Stability – the application must not fail when user gives an input with wrong input and the number of items reaches it maximum limits.
- Usability – the software must be easy to use. User manuals must be complete and understandable.
- Portability – the application must run on different versions of Android operating system.
- Testability – the application must be testable for all the test scenarios.
- Security – the application must be able to prevent a user’s list from being viewed and edited by another user.
- Maintainability – the program must be written in an understandable form with module description and comments and wherever necessary.
- The software must be compliant to the coding standards.

# Revision History

| Version | Date | Description | Author |
|---|---|---|---|
| 0.1 | 9/13/2012 | Initial Draft | Qianqian Wang |

| | | | |
|---|---|---|---|
| 0.2 | 9/16/2012 | Formatting | Arvinder&Akash |
| 1.0 | 9/20/2012 | Revised Version | Qianqian Wang |
| | | | |