

Project 2 - To Do List
CSCE 4623 – Mobile Programming – Fall 2019
Due Date – Friday, October 11th, 11:59 PM CT

Task: You have been tasked with developing a To Do List application for the Android mobile phone system. The To Do List will operate as a reminder to complete tasks that need to be finished.

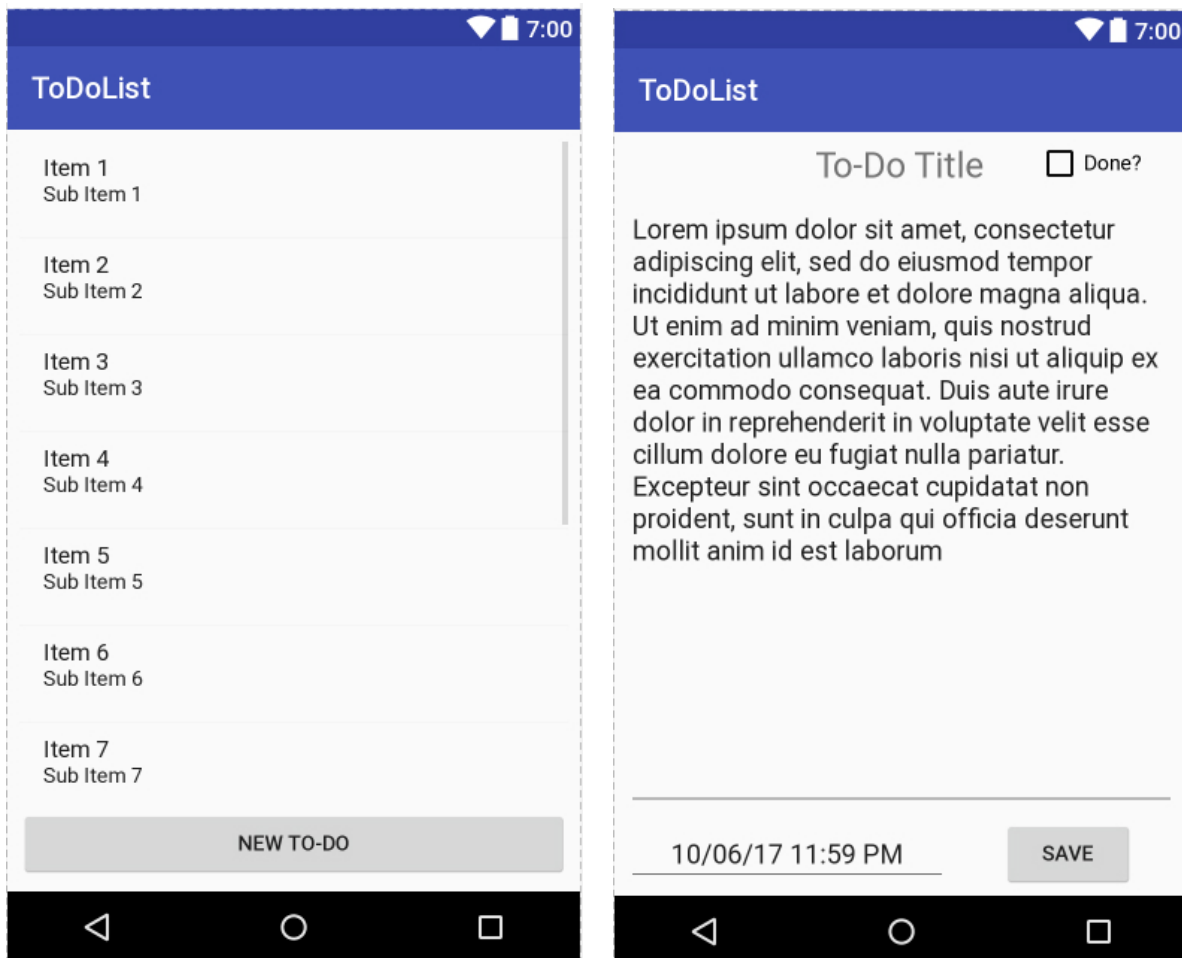


Figure 1: Google Nexus 6 Emulator showing a partially completed To Do List application

Expected Operation: The To Do List will be expected to allow a user to create, read, update, and delete tasks. A task will have a name, a text field for content, a boolean field for whether it is completed, and an optional field for a due date and time.

The To Do Application will consist of two activities: 1) The main screen which will show a list of all tasks to be completed, and 2) An information screen which displays a single task which is either being created, read, updated, or deleted. A ContentProvider is provided to you for the project, and is to be used to maintain the data for the tasks. It will need to be updated to allow fields for completion and due date. It uses a SQLite Database resident on

the phone, but all calls to the `ContentProvider` have a 3 second sleep in the field to emulate having to communicate with a remote server. For this reason, you should either use a service, or the `CursorLoader` class to perform all Database calls asynchronously.

A broadcast receiver will need to be implemented as a context-registered receiver which listens for connectivity changes. If the user no longer has access to the network, all changes to tasks should be saved to a file. When the connectivity is restored, a background task should read the changes file and synchronize with the `ContentProvider`. Because the connectivity receiver is context registered, the application should check connectivity at start-up, and if there exists previously unsynchronized data, a service should update the `ContentProvider`.

An Alarm Manager should be used to create notifications when an item has come due.

The application will be tested on the Google Pixel 3 Phone. If that does not work, I will attempt operation on the Pixel 3 emulator.

Rubric: The project will be graded according to the following rubric:

| Category | Description | Percentage |
|-------------------------|--|------------|
| Pass Given Tests | There are a set of example scenarios in this document which you must implement. | 50% |
| Pass Hidden Tests | I will run a set of hidden tests on your application to test operation. | 20% |
| User Experience | How your application looks and feels to the user is important. Consider how you want to implement the UX to appeal to your user. | 10% |
| Coding Comments & Style | Android uses XML and Java as its primary languages. Use the appropriate coding styles (e.g. lowerCamelCase for variables and methods). Comment functions with a description about their behavior, any parameters, any return values, and any shared variables which it manipulates. | 10% |
| Report | A simple, one- or two-page report. The report should have the project name (e.g. To-Do List), a picture of the application, a short description of what you did, and the outcomes (e.g. Did it pass all example tests, if not, why not, your UX design methodology, etc...). The report should use the template found on the website. Failure to do so will result in missed points. | 10% |

Table 1: Grading Rubric

Tests: The following tests will be run, and the expected output is shown:

1. Create a new task
2. Allow selection of a previously created task to be read
3. Allow update of a task's title and content
4. Allow deletion of a task
5. Allow update of task's completion
6. Allow update of task's due date
7. Notify user of task's due date through status bar notification
8. Use a broadcast receiver to note when changes to connectivity occur, and let user know
9. Store changes during connectivity loss to file

Submission: You should zip your project directory and submit it to Blackboard.