

NEAR ME – A LOCATION-AWARE TO-DO ANDROID APPLICATION

By

DEEPTI REDDY GARLAPATI

B.Tech, Jawaharlal Nehru Technological University, India, 2010

A REPORT

Submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2015

Approved by:

Major Professor
Dr. Daniel Andresen

Copyright

DEEPTI REDDY GARLAPATI

2015

Abstract

The growing needs of different products is only being increased from day to day and among these numerous products that each person plan to purchase, it has become a tedious task to keep track of all the products that should be purchased. One such important thing is that, everyone wish to keep track of an item when the location associated with the item is nearby. We have many To-Do applications where we can just note down our day to day needs and things to get. But we might face situations like forgetting to keep track of what we have in our To Do list related to buying an item and when the location associated to the item is nearby, there are high possibilities that we overlook and just forget about purchasing these items. These situations occurred most of the times when I did a small survey among my friends. This difficulty has lead me to rethink and find a solution.

The usage of smart phones has become very common these days. Android market which is an open source has helped many people to develop their own applications and these could easily be run on Android smart phones. I thought of developing an android application that helps in tracking not only the To-Do list of the items that a person tracks down to purchase but also stores the location where that item can be purchased. This To-Do app then provide notifications when a person is nearby the location associated with the item. It also triggers an alarm so that the user can easily remember what item he has planned to get in that particular location. The proposed app tries to solve most of the problems by providing an intuitive interface to the user where the user can note down all their planned purchases with location of the products and get reminders about it when passing through that location.

The Near Me application is about tracking the items to be purchased or the tasks that are to be done specific to the location. Each to-do item is associated with a date, location and notes. Storing the locations in the application helps the user with timely notifications and alarms according to the location the user is in and the tasks that should be done in that location. Also these To-Do items can also be synced with online storage application like Dropbox.

Table of Contents

List of Figures	vi
List of Tables	vii
Acknowledgements	viii
Chapter 1- Introduction.....	1
Chapter 2 –Motivation	2
Chapter 3 – Background	3
3.1 Android OS & Architecture	3
3.1.1 Linux Kernel:	4
3.1.2 Libraries:	4
3.1.3 Android Runtime:	4
3.1.4 Application framework:	4
3.1.5 Applications:	5
3.1.6 Data Storage:	5
Chapter 4 -Requirements	6
4.1 Requirement Analysis.....	6
4.2 Requirement Specification.....	7
4.2.1 Software Requirements	7
4.2.2 For running on Android device	7
4.2.3 Minimum Hardware Requirement	7
Chapter 5 - System Architecture.....	8
Chapter 6 – System Design.....	9
6.1 Use case diagram	9
6.2 Class diagrams	10
6.2.1 Class Diagram for Database Layer	10
6.2.2 Class Diagram for Services and Receivers	11
6.2.3 Class Diagram for UI Layer.....	12
Chapter 7 – Android Framework Components	13
7.1 AndroidManifest.xml.....	13
7.2 Activities.....	15

7.3 Intents.....	16
7.4 SQLite Database	17
Chapter 8 – Graphical User Interface	18
8.1 Screens	19
8.1.1 Splash Screen	19
8.1.2 User Sign In	20
8.1.3 Tasks View.....	21
8.1.4 Adding New Tasks.....	22
8.1.5 Adding Locations	25
8.1.6 Search Task	26
8.1.7 Sorting Tasks	27
8.1.8 Filtering Tasks	28
8.1.9 Dropbox Sync	29
8.1.10 Edit tasks	30
8.1.11 Deleting Tasks	31
8.1.12 Notifications and Alarms	32
Chapter 9 – Google Places Android API V2	33
Chapter 10 – Testing.....	34
10.1 Debugging.....	34
10.2 Unit Testing	34
10.3 Integration testing	35
10.4 Compatibility Testing	36
10.5 Usability testing	36
10.6 Performance testing	37
10.6.1 Navigation of Screens and its performance.	37
10.6.2 Analysis.....	37
Chapter 11 - What I have learned in developing this Application.....	38
Chapter 12 – Conclusion.....	39
Chapter 13 - Future Work.....	40
Chapter 14 - References.....	41
APPENDIX.....	42

List of Figures

Figure 3.1 Android Architecture.....	3
Figure 5.1 System Architecture	8
Figure 6.1 Use case diagram.....	9
Figure 6.2 Class Diagram for Database layer	10
Figure 6.3 Class Diagram for services and receivers.....	11
Figure 6.4 Class Diagram for UI Layer	12
Figure 7.1.1 Permissions.....	13
Figure 7.1.2 Activities.....	14
Figure 7.2 Activity Lifecycle.....	16
Figure 8.1.1 Splash Screen.....	19
Figure 8.1.2 User Sign In Screen	20
Figure 8.1.3 Tasks View Screen	21
Figure 8.1.4 Adding New Tasks	22
Figure 8.1.5 Add New Task- Add date	23
Figure 8.1.6 Add New Task- Add time	24
Figure 8.1.7 Adding locations.....	25
Figure 8.1.8 Search Task	26
Figure 8.1.9 Sorting Options.....	27
Figure 8.1.10 Filtering Options.....	28
Figure 8.1.11 Dropbox Sync.....	29
Figure 8.1.12 Edit Tasks	30
Figure 8.1.11 Delete Tasks	31
Figure 8.1.12 Notifications and Alarms.....	32
Figure 9.1 Google Places API.....	33

List of Tables

Table 10.2 Unit Test Case.....	34
Table 10.3 Response time of screens	37

Acknowledgements

I would like to thank major professor, Dr. Daniel Andresen, for his excellent guidance and encouragement throughout my project development. I would also like to thank my committee members, Dr. Mitch Neilsen and Dr. Torben Amtoft for serving on my committee.

I would like to acknowledge the academic support I received from the Professors of Computing and Information Sciences at Kansas State University and also for giving me the opportunity to take various courses under them that has helped me in this project. I would also like to acknowledge the academic and technical staff of the CIS department for the support and help in my graduate study.

I would like to thank my family- Shekar Reddy Garlapati, Aruna Reddy Garlapati, Rahul Reddy Garlapati and friends for their encouragement and tremendous support throughout my study period.

Finally, thank you Karthik Reddy Ginuga for your constant support and motivation and being there for me in my most difficult times.

Chapter 1- Introduction

Near Me is an Android application that helps in tracking down the to do list items and also provides push notifications of the item to purchase or action to be done in to do list when the location associated with is nearby. Most of the people have the tendency to forget their to-do tasks and things get even more difficult to remember when it comes to grocery, apparel, furniture and other kinds of shopping. Our application solves this kind of problems by providing the features of a To-Do app which is location aware.

In Near ME application, each To-Do item is associated with a date, location and notes which helps the user to set a date for the item and also provide the location by giving some notes on what action exactly that has to be done at that location related to the To-Do item. This helps in triggering an alarm when the date approaches or by providing push notifications when the location associated with the item is nearby. All these To-Do items can also be categorized into one of many categories available. Application also develops the user to add coupons.

The use of many online applications like Dropbox and Google Drive have become quite common these days and are much more efficient in accessing the files from anywhere else. Therefore in Near Me application, each To-Do item can be synced to these online storage applications for easy storage and accessibility.

Chapter 2 –Motivation

The purpose of this project is to create a location aware To-Do android application of the various To-Do items that of a user in day to day life. Keeping track of these To-Do lists timely has become a difficult task for many people these days as the growing needs in daily life only increased from day to day. Observing this, an idea of developing an Android application that is location aware of the To-Do items has come up and this Near Me application provides notifications and alarms that helps the user to get the To-Do things done on time and also these notifications helps the user to be aware of them when nearing its location. The online application synchronization of Dropbox and Google Drive will help the user to be updated and keep track of their history more appropriately.

Chapter 3 – Background

3.1 Android OS & Architecture

Android is based on a Linux operating system and also an open source. This creates a good platform for the developers to create and develop new applications as well as modify the applications that are already existing .It can be used on various tablets and smart phones available in the market. The developers can provide these android applications to the various users to download and they start on with them. Developing an android applications is not that hard as it seems to be. There are numerous resources and documentations available online which helps and guides everyone who are interested in developing an android app in Android OS.



Figure 3.1 Android Architecture [1]

Figure 2.1 above represents different layers of components of the Android OS. There are certain functionalities that each layer offers to the layers above it.

3.1.1 Linux Kernel:

It is the lower layer in the Android Architecture. It has different drivers and it interacts with the hardware of the device. This layer does not interact with the users or developers. It helps in memory, power and process management, security and networking. It also acts as an abstraction layer between hardware and software layers of the architecture.

3.1.2 Libraries:

The layer that is above Linux Kernel layer is 'Libraries' which helps in handling distinct data. The other open source libraries in C and C++ are SQL Lite, Surface Manager, SSL(internet Security), SGL (render 2D graphics), libc ,(system C Libraries) , OpenGL | ES, FreeType, Webkit that is a browser engine that displays HTML content

3.1.3 Android Runtime:

Android Runtime is also above the Linux Kernel layer and on with the same level of Libraries, It contains libraries that are native to virtual machine and android in order to run the native code.

3.1.4 Application framework:

It is a layer above the Libraries and Android Runtime. This layer communicates directly with the applications that are developed by the android developers. The services are provided in the form of classes to the applications that can be used by the developers. The blocks in this framework

are Activity Manager, Window Manager, Content Providers, View system, Package Manager, Telephony Manager, Resource Manager, Location Manager and Notification Manager.

3.1.5 Applications:

Applications is the top layer of the Android Architecture and here resides the Android application that is accessible to the users. In this layer, applications are written by the Android developers. There are few pre-installed applications like Home, Phone, Browser and Contacts.

3.1.6 Data Storage:

We have various options in android for saving persistent data. Private primitive data is stored in the form of a key-Value pairs when Shared Preferences are considered. Internal Storage, the private data is stored on the device directly. When external storage is considered, the public data is stored on shared external storage. SQLite Database will store the data in a structured form on a private database.

Chapter 4 -Requirements

4.1 Requirement Analysis

To build a useful, robust and user friendly applications, requirement analysis is the key to it. Gathering all the requirements is an important task. Thinking in every possible way and getting information about the current apps in the market will also help in being updated to the latest features and also the user interface. I have done some research finding out the apps in the market and also few difficulties my friends were facing in their day to day activities. I found that keeping track of things to be done in To-Do list timely and forgetting to get things done when the location associated with the item is near-by has become quite a tedious task. This interaction and research has also helped me what kind of features I should be implementing in my android application.

The main requirements I thought are important in the application are:

- An Easy and basic To-Do interface which allows the user to list down the activities
- Each of these To-Do items should be associated with a date, location and notes.
- A simple, compact and efficient way of displaying all the To-Do lists in detail.
- Easy navigation between the stored data and retrieving of the data that is stored without any loss.
- SQLite database is used as a storage for this app.
- A feature where each To-Do item can be synced to multiple online storage apps like Dropbox and Google drive
- Also to receive push notification when the To-Do is due or when the location associated with the item is near-by.

- The designed application has to support tablets and android phones along with varied sized screens and can be used on Android version 3.0 and other updated versions after that. Near Me android application was developed on Android 5.0

4.2 Requirement Specification

4.2.1 Software Requirements

Operating System: Linux/ Mac OS X mavericks or higher /Windows XP or higher

Technologies used: Java, SQLite, Android, Google Maps V2 API,

Platform: Android SDK Framework 14 or higher

Java Database: SQLite, Couch DBLite-NoSQL

Database Tools: Android studio

Android Emulator: SDK Version 4.0 or higher

Debugger: Android Dalvik Debug Monitor Service (DDMS)

4.2.2 For running on Android device

Operating System: Android 4.0 or higher versions

Network: Wi-Fi Internet or cellular Network

4.2.3 Minimum Hardware Requirement

Processor: Intel Pentium IV or higher RAM: 256 MB

Disk-Space: 250MB

Device: Android Tablet or Smart phone with Android version 4.0 or higher

Space to execute: 5.0MB

Chapter 5 - System Architecture

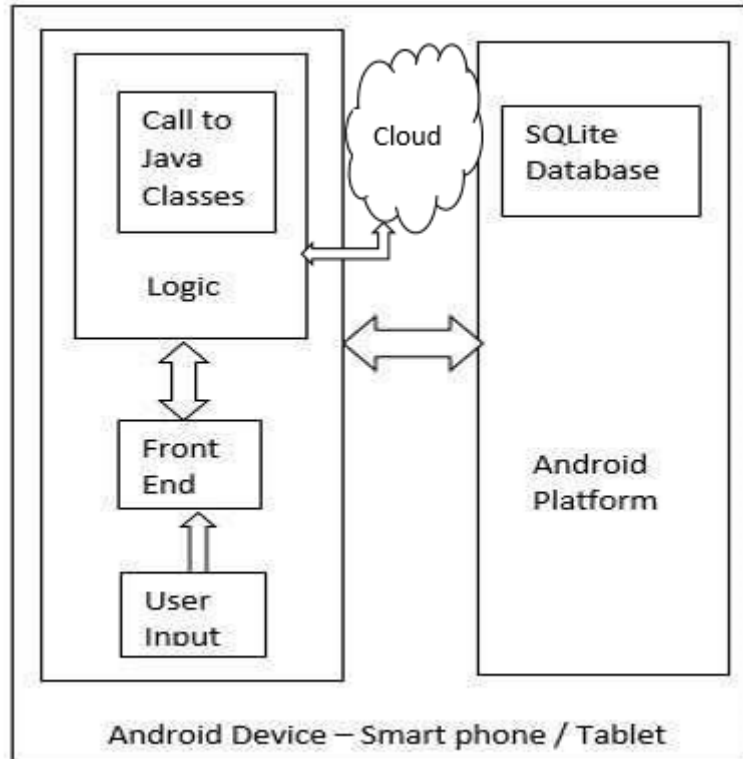


Figure 5.1 System Architecture

This is the system architecture of the Near Me Android application.

Logic: It defines the functionality of the application by wiring up the UI code and communication to android platform.

SQLite database: We also have SQLite database store on the disk which is accessed through logic.

Front end: This is where we define layouts, menus, themes and all other UL related part.

Cloud: Since we have features of uploading tasks to Dropbox, I have indicated that in the Architecture using a cloud symbol.

Chapter 6 – System Design

6.1 Use case diagram

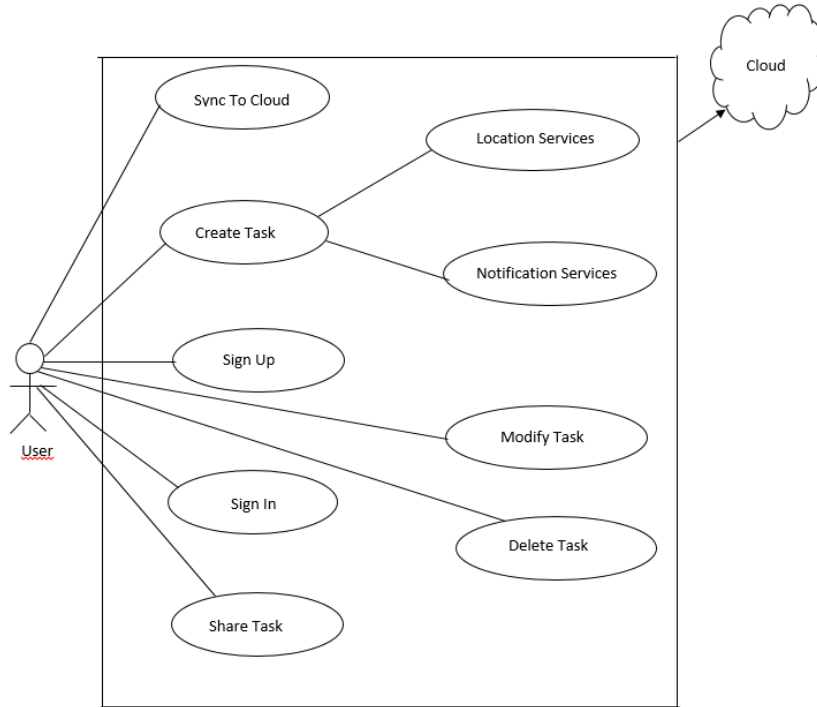


Figure 6.1 Use Case Diagram

In UML, the use case diagrams are used to describe the system’s functionality in a horizontal method. These diagrams help us to determine the available functionalities easily without going in detail of each individual features. It gives us the high level system view. Below explained are the elements of the Use Case Diagram.

- User shows what system the user interacts with and hence depicts the role of the application user.
- Use cases are the lists of tasks that allows us to understand the interactions between the user and the system.
- The relationship between the use cases and the users are represented by lines.

6.2 Class diagrams

Class diagrams gives us the overall structure of the system. They display the classes, attributes, associated methods and relationships among them. The above class diagrams illustrates the system designs of Near Me android application.

6.2.1 Class Diagram for Database Layer

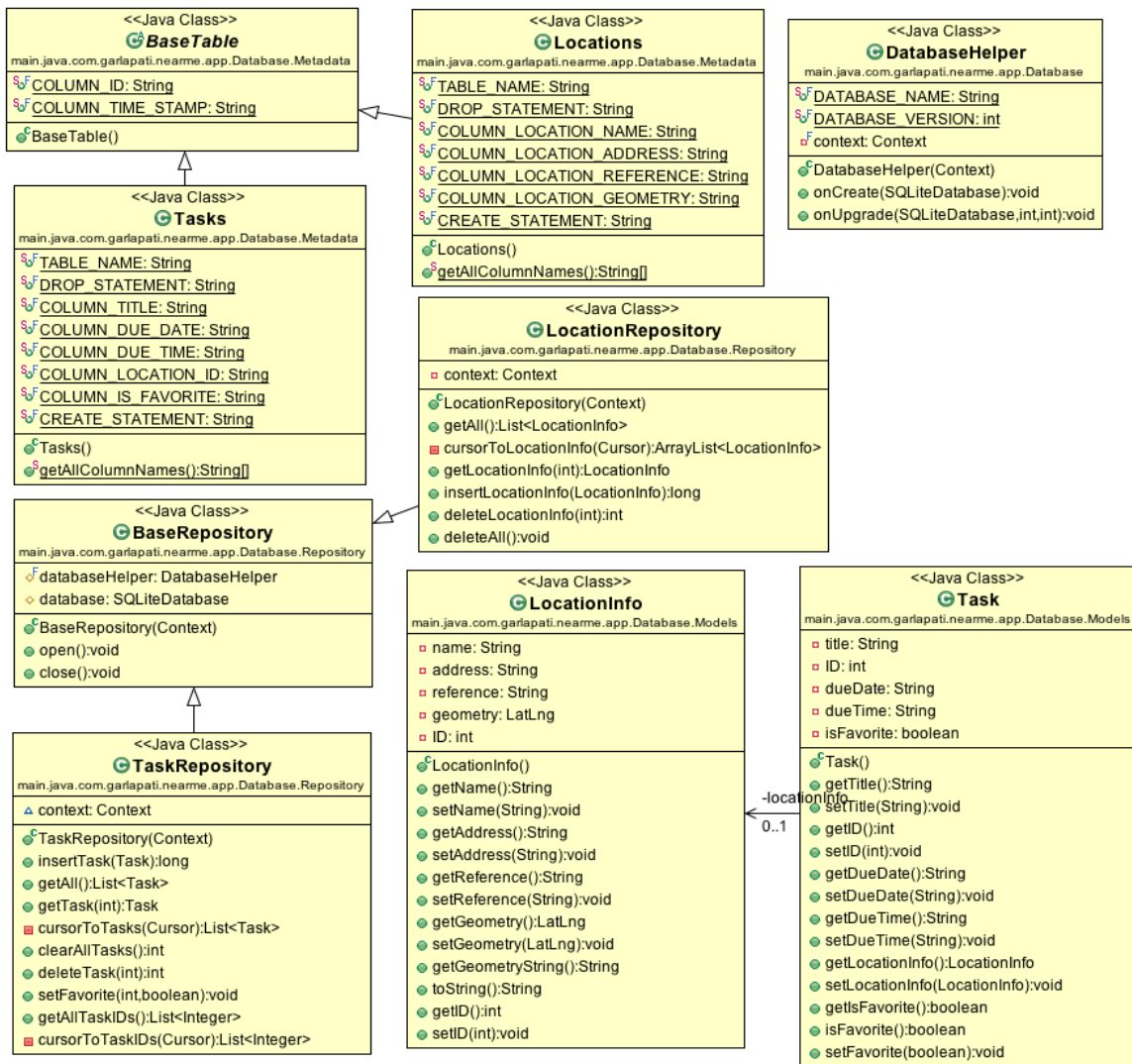


Figure 6.2 Class Diagram for Database Layer

6.2.2 Class Diagram for Services and Receivers

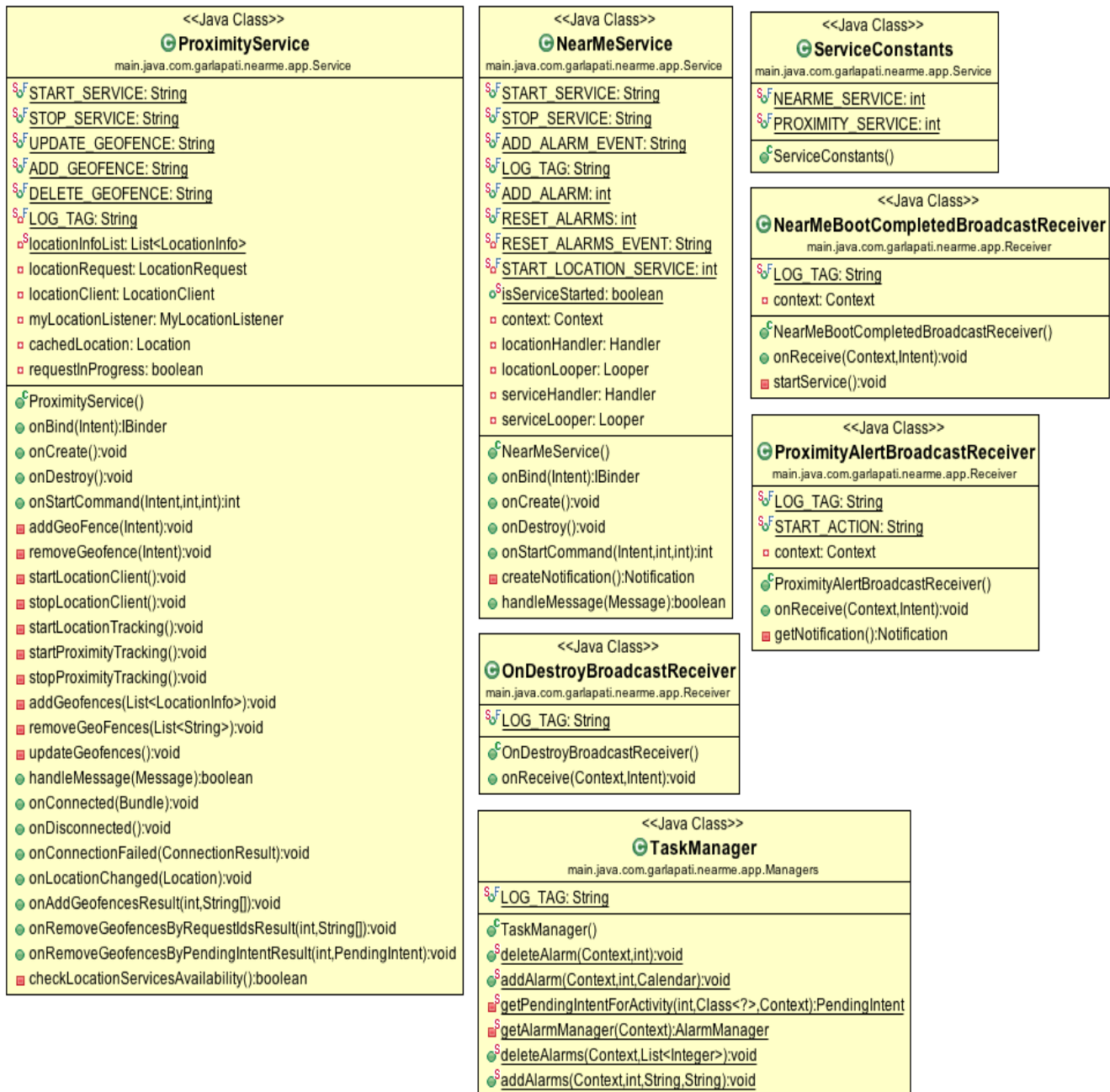


Figure 6.3 Class Diagram for Services and Receivers

6.2.3 Class Diagram for UI Layer

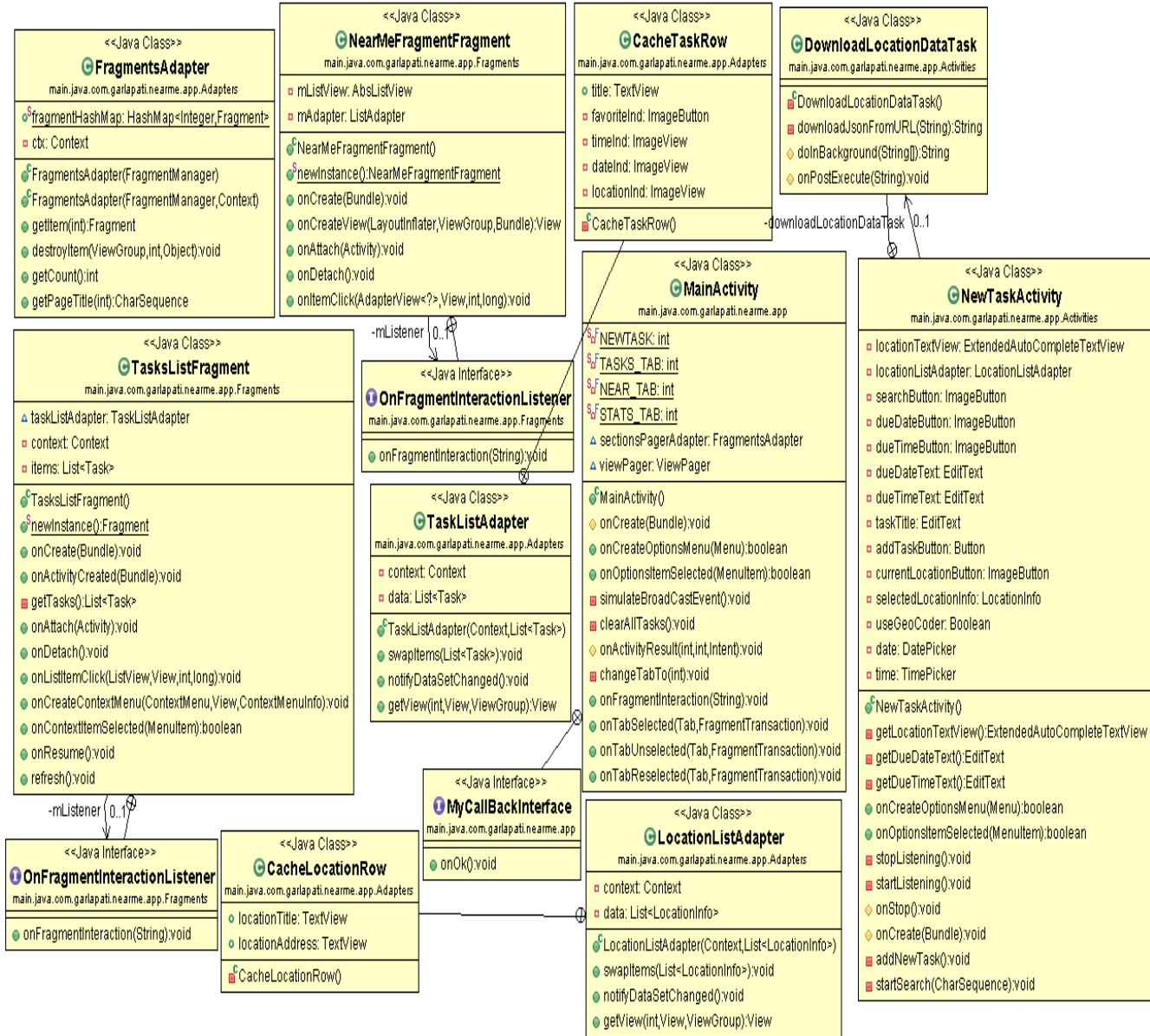


Figure 6.4 Class Diagram for UI Layer

Chapter 7 – Android Framework Components

Near Me Android application can be developed through many integrated development environments and the IDE that I used to develop is Android Studio. It is developed using Android SDK (Software development kit) in Java.

7.1 AndroidManifest.xml

The AndroidManifest.xml can found in the root directory of the android project and it consists of the information that is required by the android application to run the code. This file has different android components and their descriptions. The components include activities, content providers, services, intents and its filters. It also has permissions that are required for accessing various Application programming interfaces and also to access other applications on the device. This makes it possible to access gallery, camera, social media, maps, network and also the internet.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="apps.garlapati.nearmev2" >

    <permission
        android:name="apps.garlapati.nearmev2.permissions.MAPS_RECEIVE"|
        android:protectionLevel="signature" />
    <permission
        android:name="apps.garlapati.nearmev2.permission.C2D_MESSAGE"
        android:protectionLevel="signature" />

    <uses-permission android:name="apps.garlapati.nearmev2.permission.C2D_MESSAGE" />
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="com.garlapati.nearmev2.app.permissions.MAPS_RECEIVE" />
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

Figure 7.1.1 Permissions

The above Figure 7.1.1, lists all the permissions that are required by the Near Me Android application.

```
<activity
  android:name=".Activities.MainActivity"
  android:label="@string/title_activity_main" />
<activity
  android:name=".Activities.SplashActivity"
  android:label="@string/AppName"
  android:noHistory="true" >
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
<activity
  android:name=".Activities.NewTaskActivity"
  android:label="@string/title_activity_new_task"
  android:parentActivityName=".Activities.MainActivity" >
  <meta-data
    android:name="android.support.PARENT_ACTIVITY"
    android:value="apps.garlapati.nearmev2.Activities.MainActivity" />
</activity>
<activity
  android:name=".Activities.EditTaskActivity"
```

Figure 7.1.2 Activities

The above Figure 7.1.2, displays some of the activities that were created in the Android Manifest file.

For this application the minimum API version is 16 and we need the versions Android 4.0 (Icecream sandwich) and above to run this application due to the latest and updated features used to develop this application. Below is the detailed view of AndroidManifest.xml of Near Me android application.

7.2 Activities

This Activity component of the Android application helps interacting with the user and only those activities in the <application> tags can be recognized by the system and can be used. There are different android activities that user starts, creates and destroy. When a new activity is started, it will be the current activity. All other activities will be in the background. When an activity is running in the foreground, it is called an active state. An activity that is not in focus but can be visible is in a paused state. When we want to stop the activity or pause it then that can be done by killing the process. The user side interface components are associated with each activity in the manifest.xml and these user interface components can be anything from text boxes, stats, images, labels etc. The activities that Near Me application has are Create task, View task, Edit task, List of tasks and Sync task. On opening the application, we see a list of tasks that is managed through mainactivity. Here is a sample code that drives it.

```
@Override
protected void onCreate(Bundle savedInstanceState) {

    startService(new Intent(NearMeService.START_SERVICE));
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Set up the action bar.
    final ActionBar actionBar = getSupportActionBar();
    actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);
    actionBar.setHomeButtonEnabled(true);
    actionBar.setTitle("Near Me");

    sectionsPagerAdapter = new FragmentsAdapter(getSupportFragmentManager(), this);

    viewPager = (ViewPager) findViewById(R.id.pager);
    viewPager.setAdapter(sectionsPagerAdapter);

    viewPager.setOnPageChangeListener(new ViewPager.SimpleOnPageChangeListener() {
        @Override
        public void onPageSelected(int position) {
            actionBar.setSelectedNavigationItem(position);
        }
    });
}
```


Below is the diagram that displays all the state paths of an android activity.

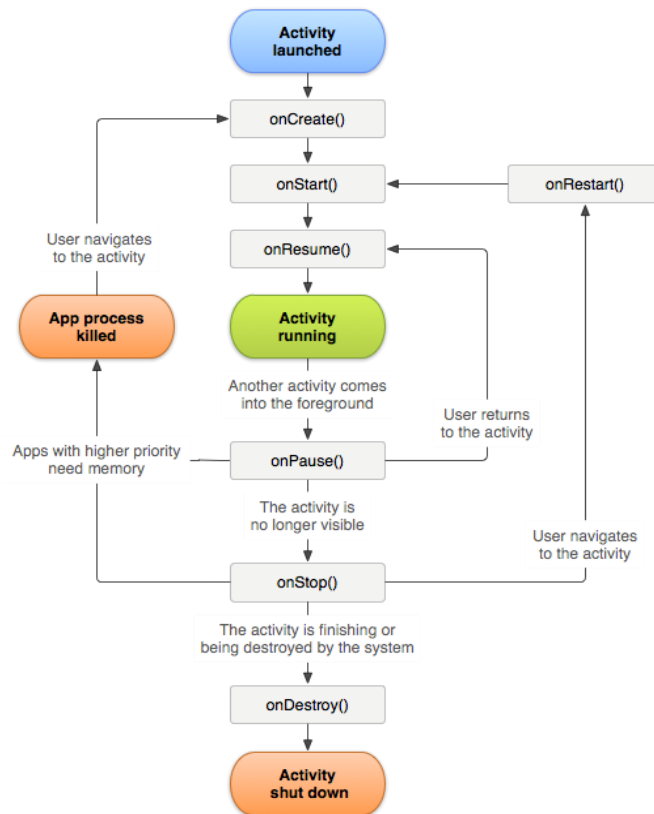


Figure 7.2 Activity Lifecycle [2]

7.3 Intents

An Intent is an abstract description of the operation to be performed by an activity.

Intents provide communication between different components like activities, services, content providers and other fragments of the application. Intent is mainly to launch activities and this can be done by using the function `startActivity()`. Intent also provides a late runtime binding between the codes in different activities in the application. Intent structure contains the information of the data and general actions to be performed like view edit and main. The other attributes that can

also be included in the intents are category, type, and component. Intent has two primary forms and they are explicit and implicit intents.

Explicit intents have a components that are specified and help in launching the internal activities when the user interacts. The below code is an explicit intent example that has been used to launch the Sign in Activity in Near Me Android application.

```
Intent intent = new Intent(context, SignInActivity.class);
intent.putExtra("isSignOut", true);
context.startActivity(intent);
```

Implicit intents are those that do not specify any component but they make sure that they have sufficient information to decide upon the best available components to run for that particular intent. We can start a component which is available in another application. Below is an example of implicit intent used to start the mailing services for Near Me Android application.

```
Intent emailIntent = new Intent(Intent.ACTION_SEND);
emailIntent.setData(Uri.parse("mailto:"));
emailIntent.setType("text/plain");
emailIntent.putExtra(Intent.EXTRA_EMAIL, TO);
emailIntent.putExtra(Intent.EXTRA_CC, CC);
emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Help with Nearme app");
emailIntent.putExtra(Intent.EXTRA_TEXT, "");
```

7.4 SQLite Database

Android applications which require relational database can use many different databases. When the data is specific to the user/device and needs to be stored on the Android device, we need a database which can manipulate the local memory. Android platform has a built-in SQLite database engine which is self-contained, server less and requires no configuration.

The database is created by extending SQLiteOpenHelper class and the created database is stored in device memory due to which the database is local to that application on the device. Database is created by overriding onCreate() function of the SQLiteOpenHelper class. The SQLite database objects can be used for transactions on the database.

Chapter 8 – Graphical User Interface

The Near Me Application is developed and designed in such a way that it helps in keeping track of the all the To-Do lists made and remind the user, when the To-Do list of the item expires by raising an alarm and also give a notification when the location associated with to-do item is nearby. This application design gives a good user interface and makes it very simple to track our needs on the go. The user can store the details on to Dropbox, google drive and also fetch them whenever it is required. Features of this application are the following:

- Create a New Task - The user can create any task that has to be done.. They can give the due date and time for the task, and location associated with the item for easy tracking.
- Near Me – All the tasks in the to-do list which are not completed are listed in this screen according to the nearer location associated with the tasks.
- Tasks- The user can view all the to-do tasks and by selecting each of the task, the user can edit the details of it that are stored.
- Sync the tasks – This is one of the cool feature of the application. When the user creates a new task, he can easily select the task and sync it through the available online options like dropbox and google drive. This makes it simple and easy for viewing the tasks details and not depending on mobile completely to access the to-do lists.
- Stats – It is a combined view of how many tasks have been completed and how many tasks are pending in total.

The Near me application is developed using Android Studio IDE on Android SDK 5. Dalvik Debug Monitor Server (DDMS) is used to debug the application, the application is developed using Java and XML and for the transactional data, the logic written is connected to SQLite database.

8.1 Screens

The main motive for the android developers is to keep the user interface for the user's simple, easy navigation and user friendly. It is developed using XML. Below are the screens for different functionalities.

8.1.1 Splash Screen



Figure 8.1.1 Splash screen

The above figure is the Splash screen that appears at the launch of Near Me application and it seen for 1500 milliseconds.

8.1.2 User Sign In

User Sign In screen will ask the user to sign in with their Facebook account.

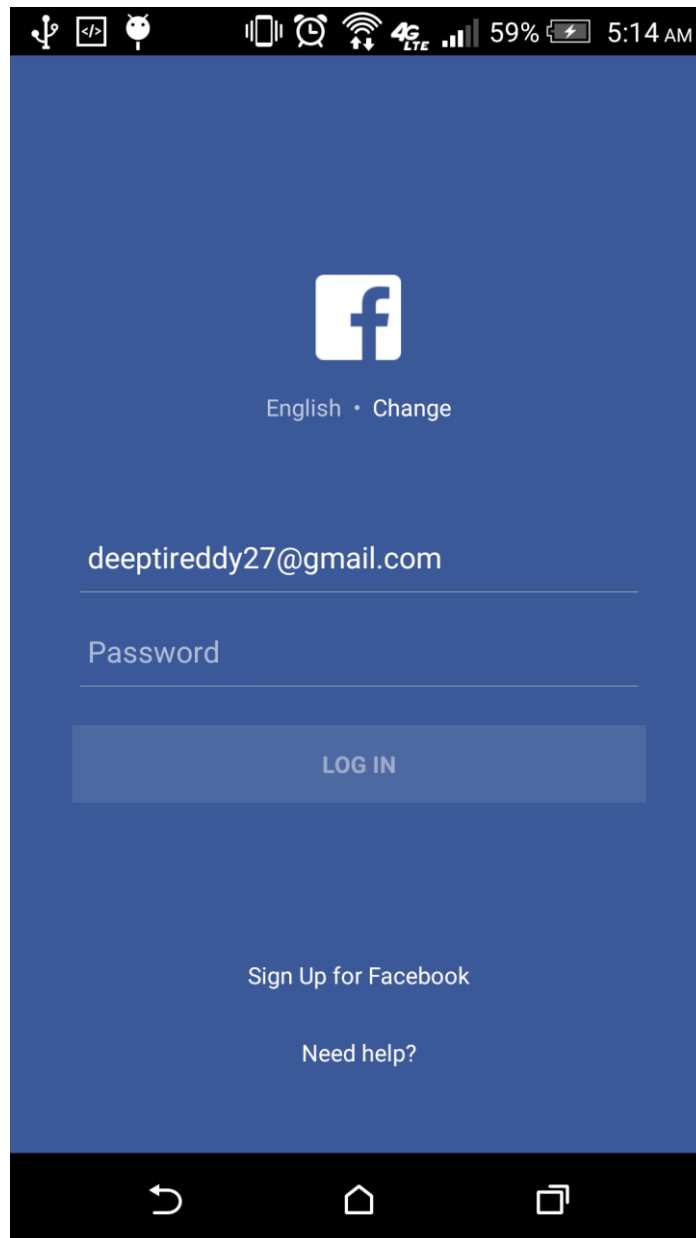


Figure 8.1.2 User Sign in Screen

8.1.3 Tasks View

The application has few options when the Tasks View screen appears where the user can perform those actions when required. They are:

- Search task
- Sort all tasks
- Filter tasks
- Upload to Dropbox
- Edit tasks
- Delete tasks

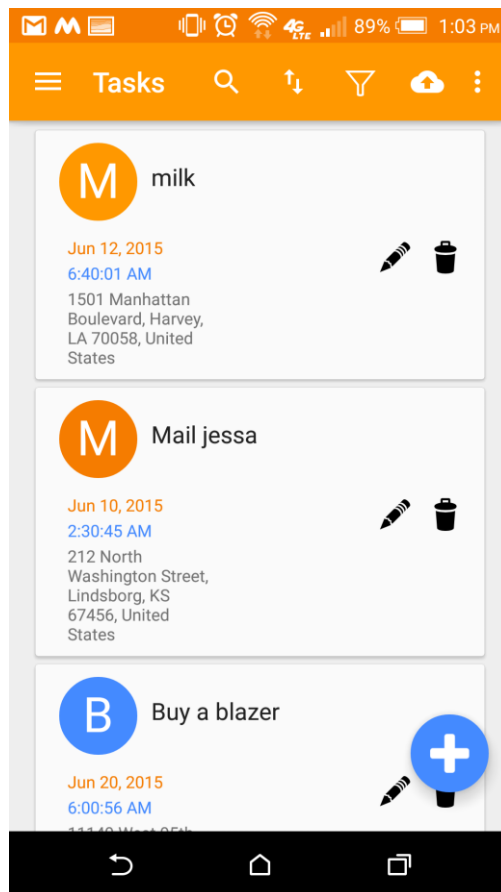


Figure 8.1.3 Tasks View

8.1.4 Adding New Tasks

This screen allows the user to create a task by asking to enter about what the application should remind the user about, a due date when the task should be done and also the due date. It also asks the user about the location where this task should be completed.

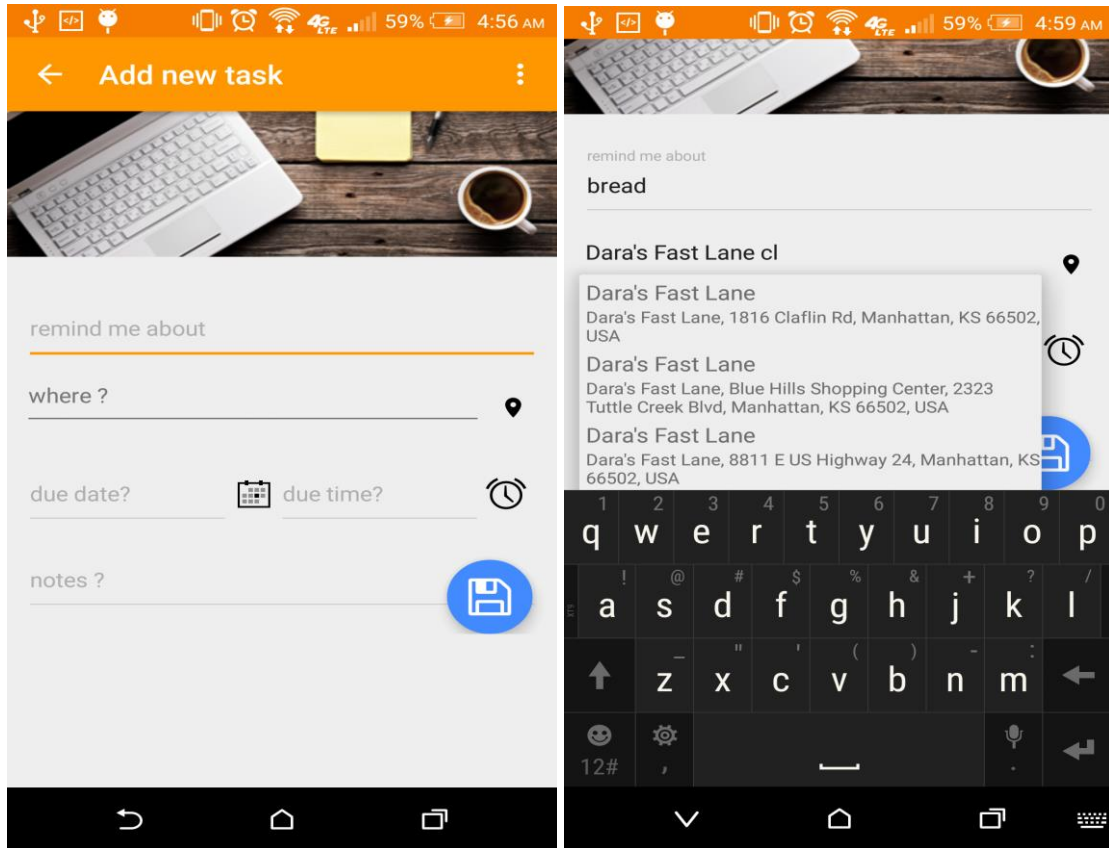


Figure 8.1.4 Add New Task

This date and time picker helps the user to navigate through months, dates and years easily. Our application gets the date and time using a reference to these when the fragments ends it session it returns the date and time respectively.

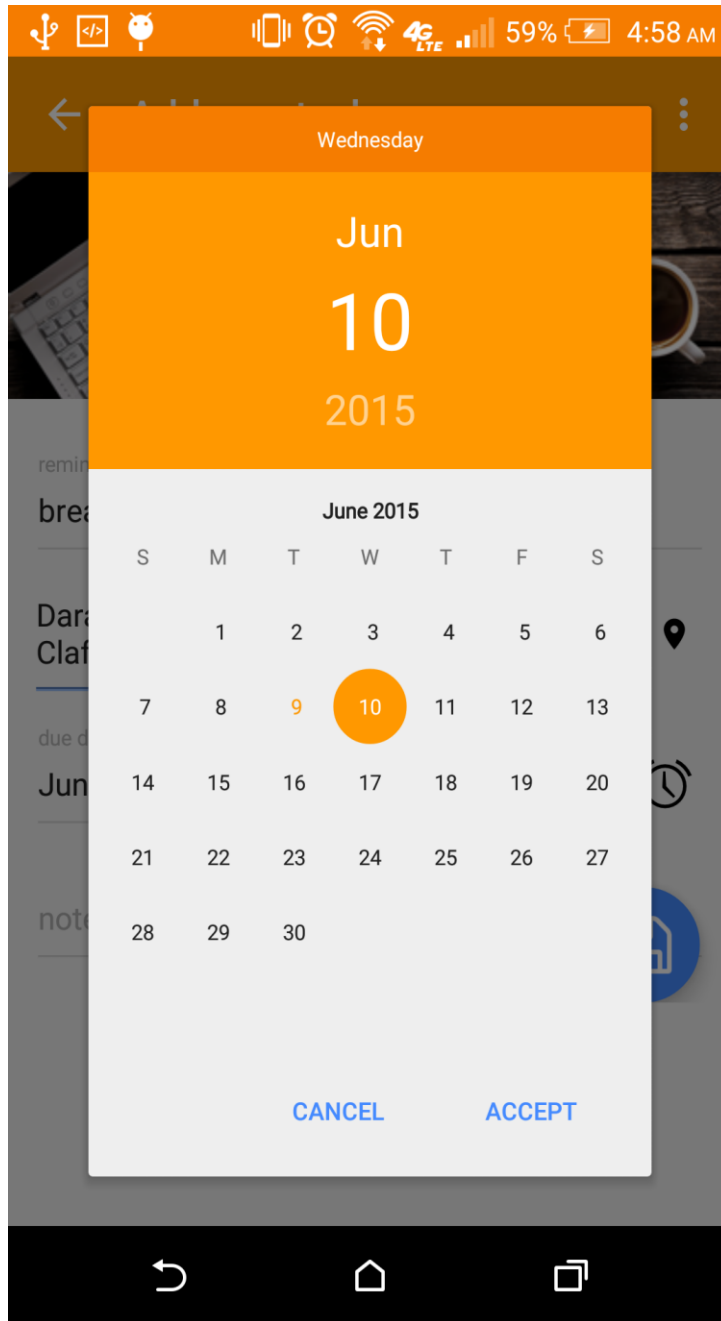


Figure 8.1.5 Add New Task- Add date

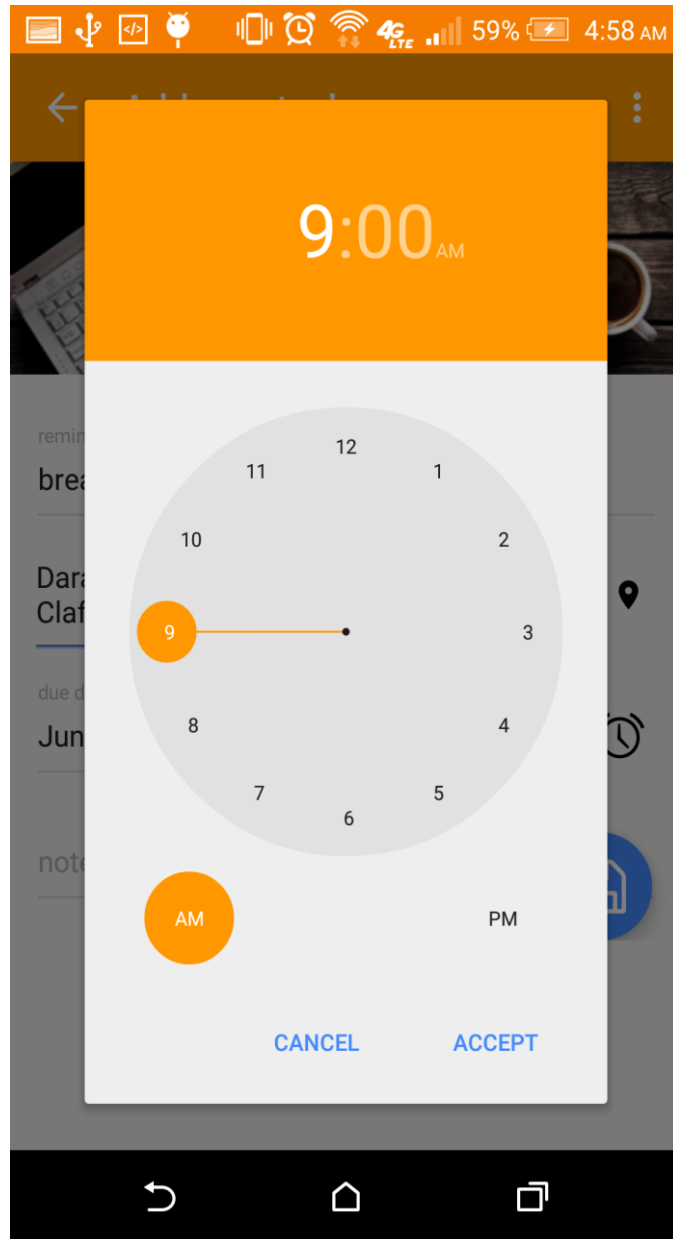


Figure 8.1.6 Add New Task- Add time

8.1.5 Adding Locations

While entering the location, it also suggests the addresses accordingly while entering the city or state by the user. This eliminates the need of typing of the whole address.

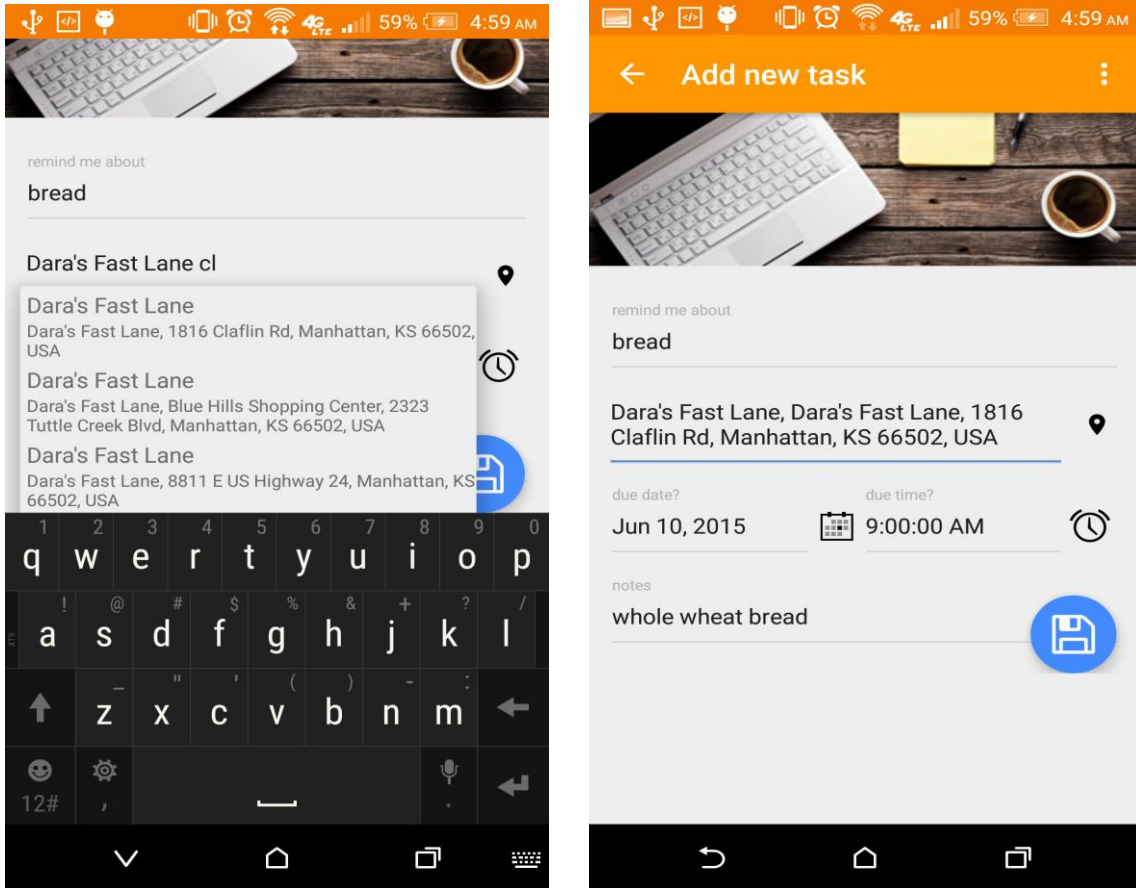


Figure 8.1.7 Adding Locations

8.1.6 Search Task

When you click on the user option tools, the search option helps the user to search any task from the task list stored.

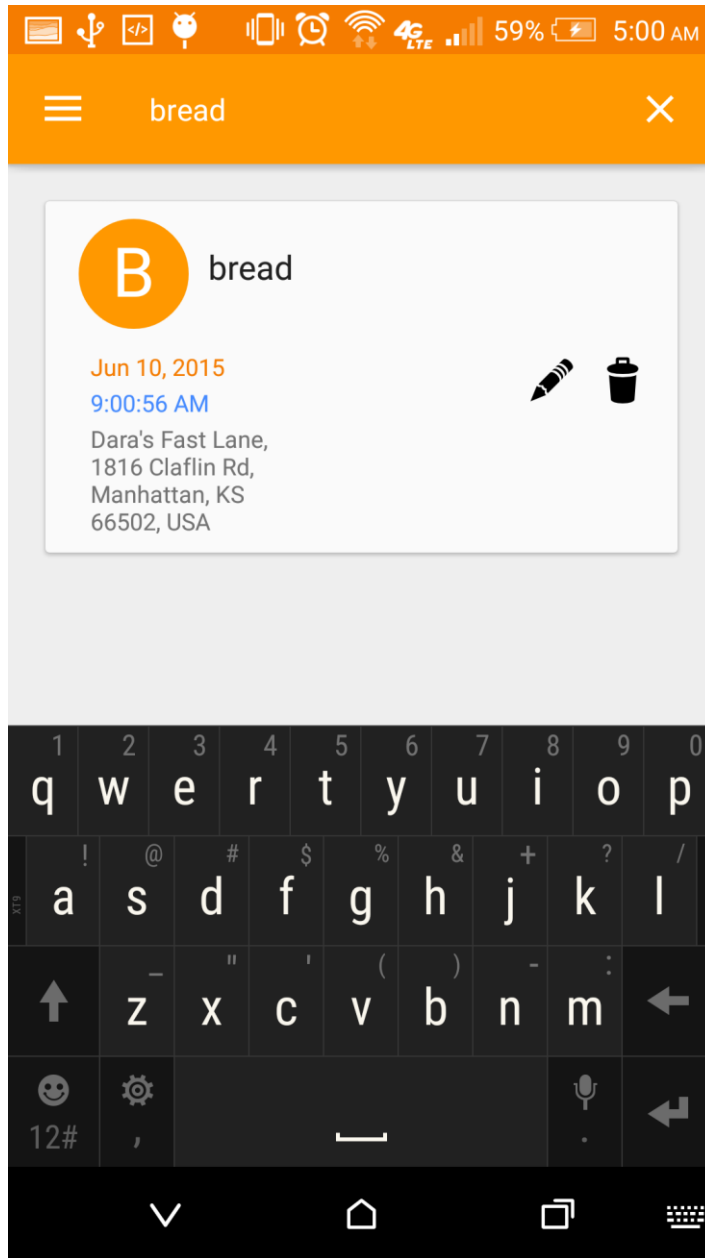


Figure 8.1.8 Search Task

8.1.7 Sorting Tasks

The user will be able to sort out the tasks by the following three ways:

- Distance
- Title
- Oldest First

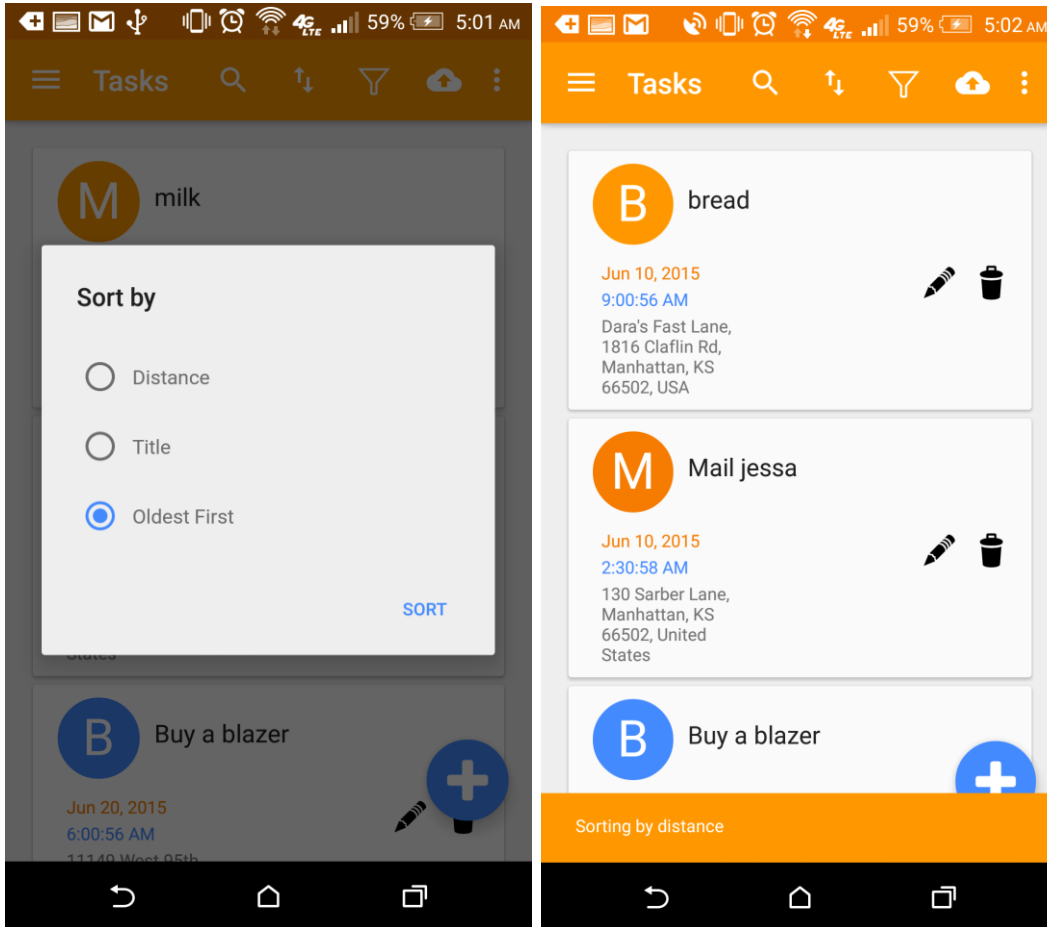


Figure 8.1.9 Sorting Options

8.1.8 Filtering Tasks

When there are lot of tasks in one's list, it would be easy to find few tasks by filtering them be following to options:

- Tasks with Due date
- Location enabled Tasks

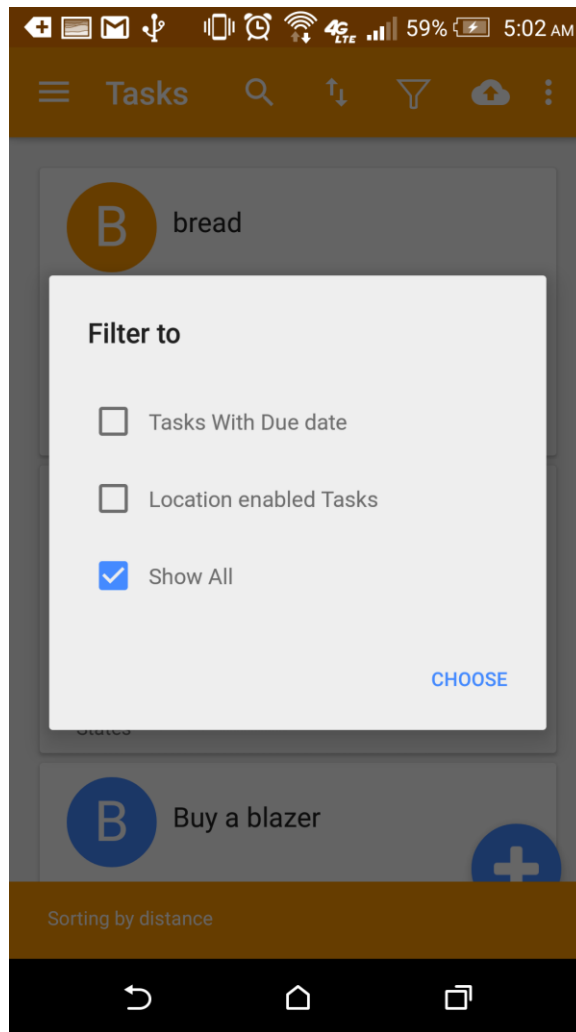


Figure 8.1.10 Filtering Options

8.1.9 Dropbox Sync

This option of uploading all the tasks unto Dropbox helps in backing up all the data on to the cloud.

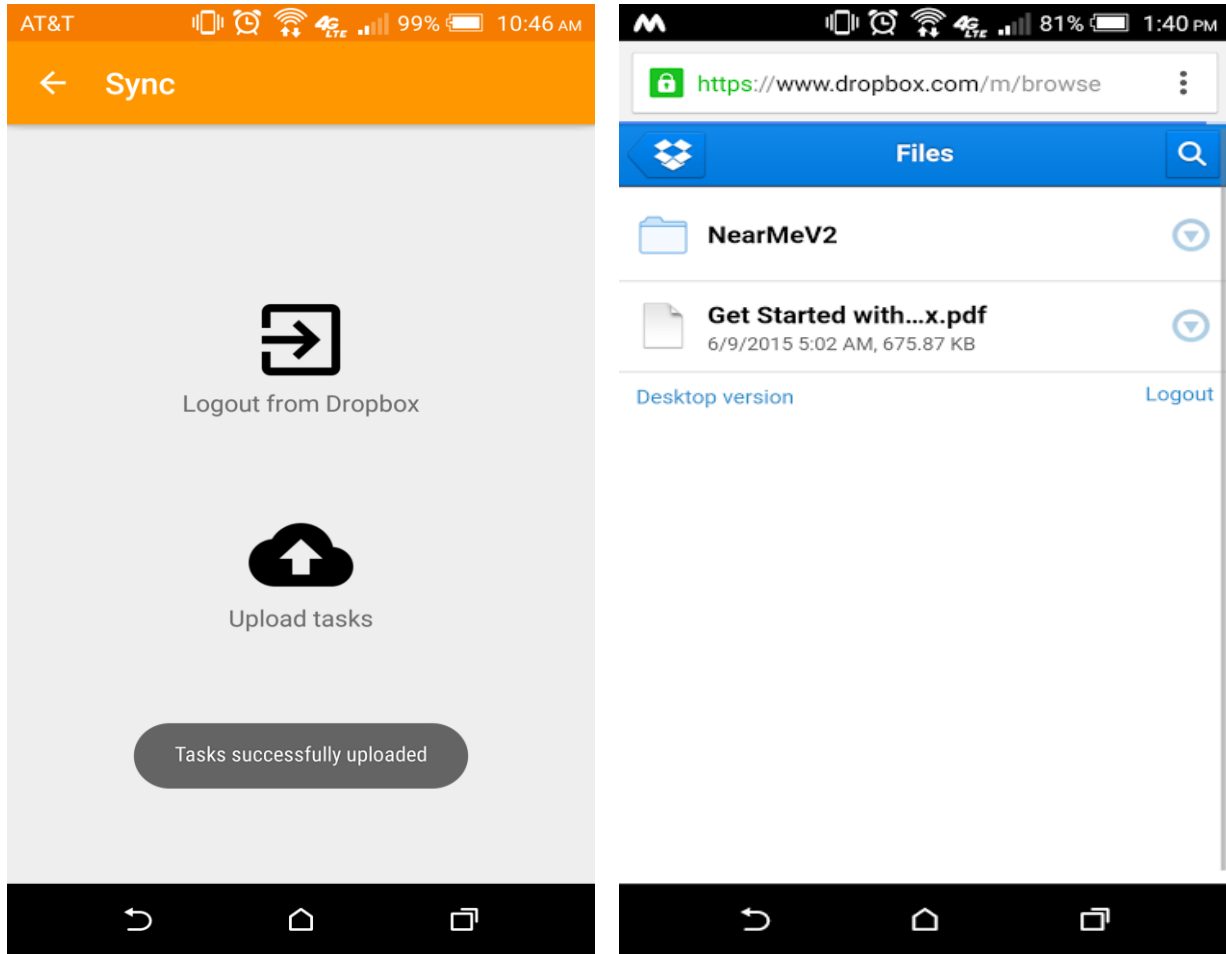


Figure 8.1.11 Dropbox sync

8.1.10 Edit tasks

All the tasks in the application can be edited with the task name, location, due date and time and notes respectively.

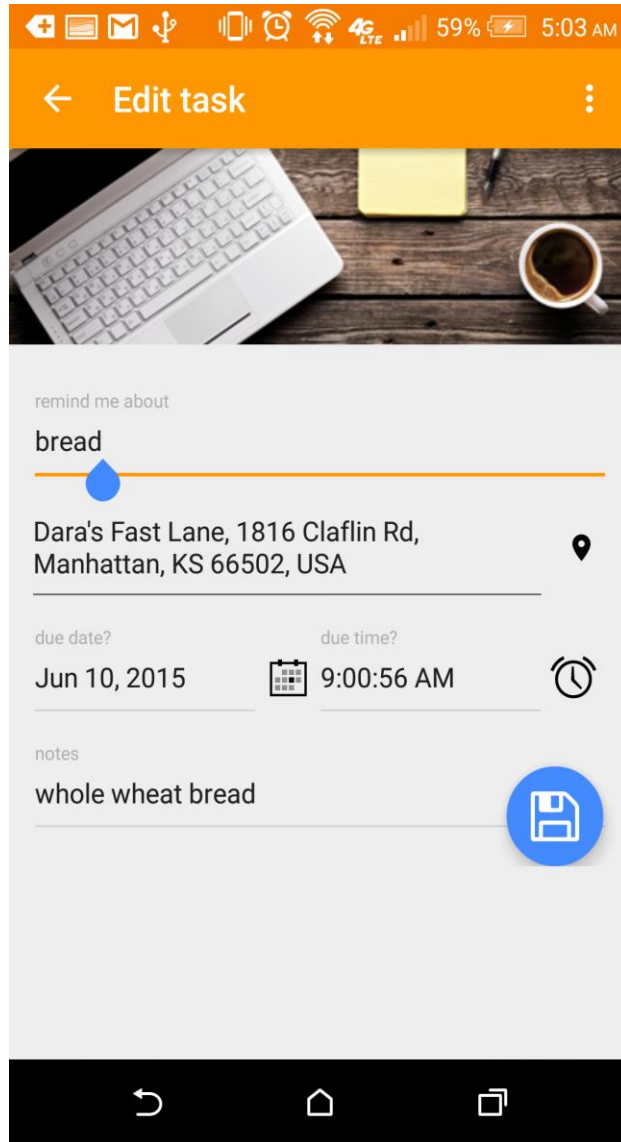


Figure 8.1.12 Edit tasks

8.1.11 Deleting Tasks

When user is able to view all the tasks on the screen, there is an option for the user to delete the task if completed or is not necessary anymore.

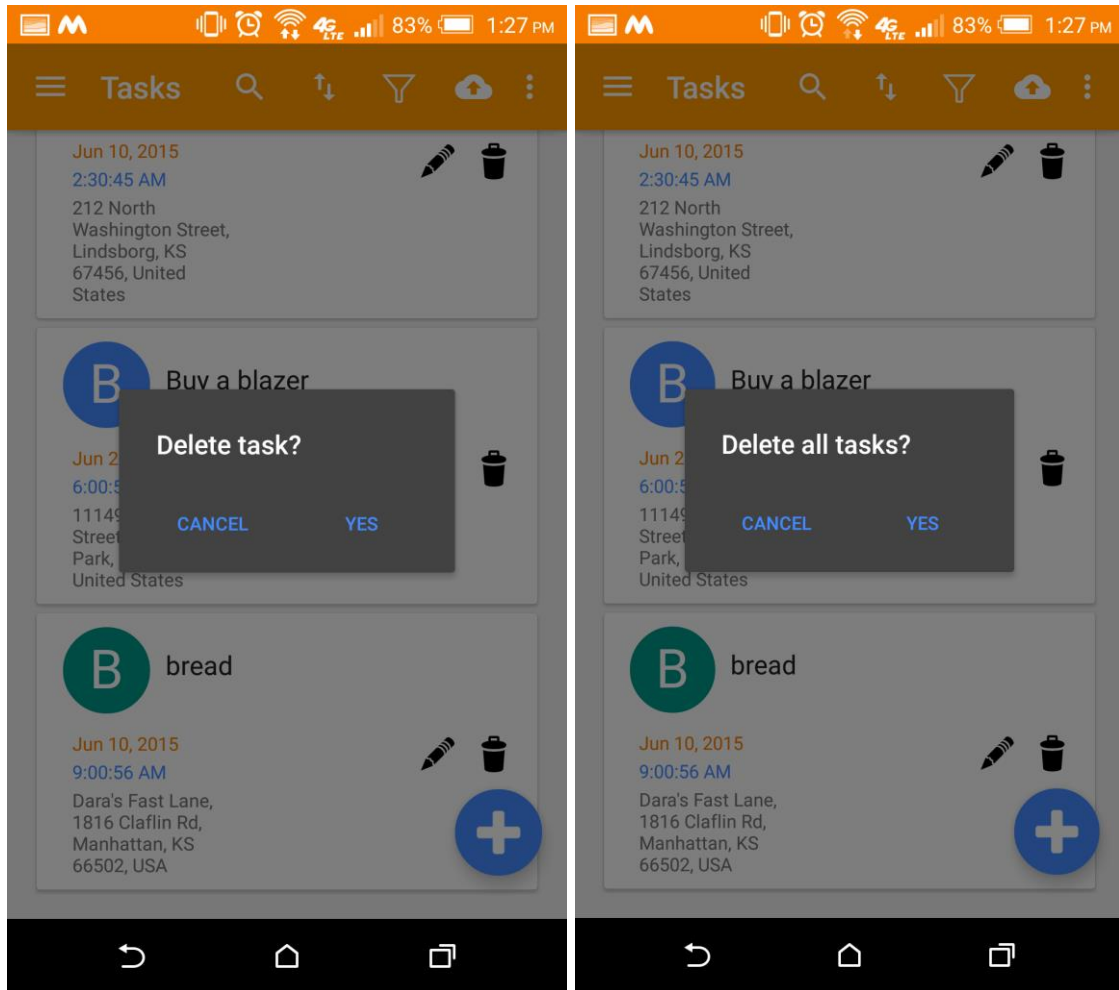


Figure 8.1.13 Delete Tasks

8.1.12 Notifications and Alarms

When the user is near the location associated with the task or when the due date of the task is approaching, the notification and alarms trigger to alert the user.

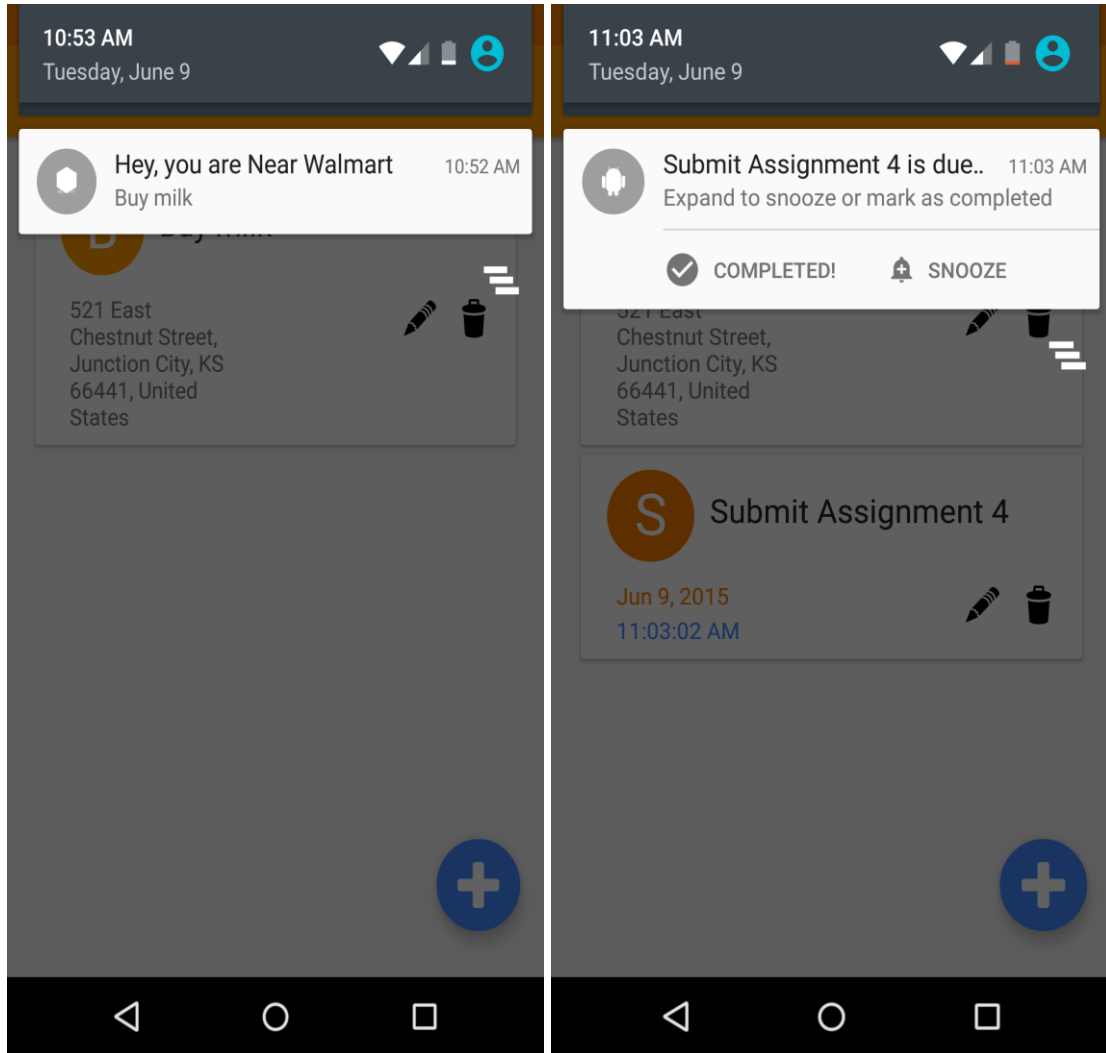


Figure 8.1.14 Notifications and Alarms

Chapter 9 – Google Places Android API V2

I used google places API to get the location services in Near Me Android application. I have downloaded and imported the google play services to android gradle file. Also generated the SHA – 1 fingerprint using the java key tool. Below is a screen shot that shows the process of creating a new Android Key under API access. Also on google API key console, we turn on the google places Android API.

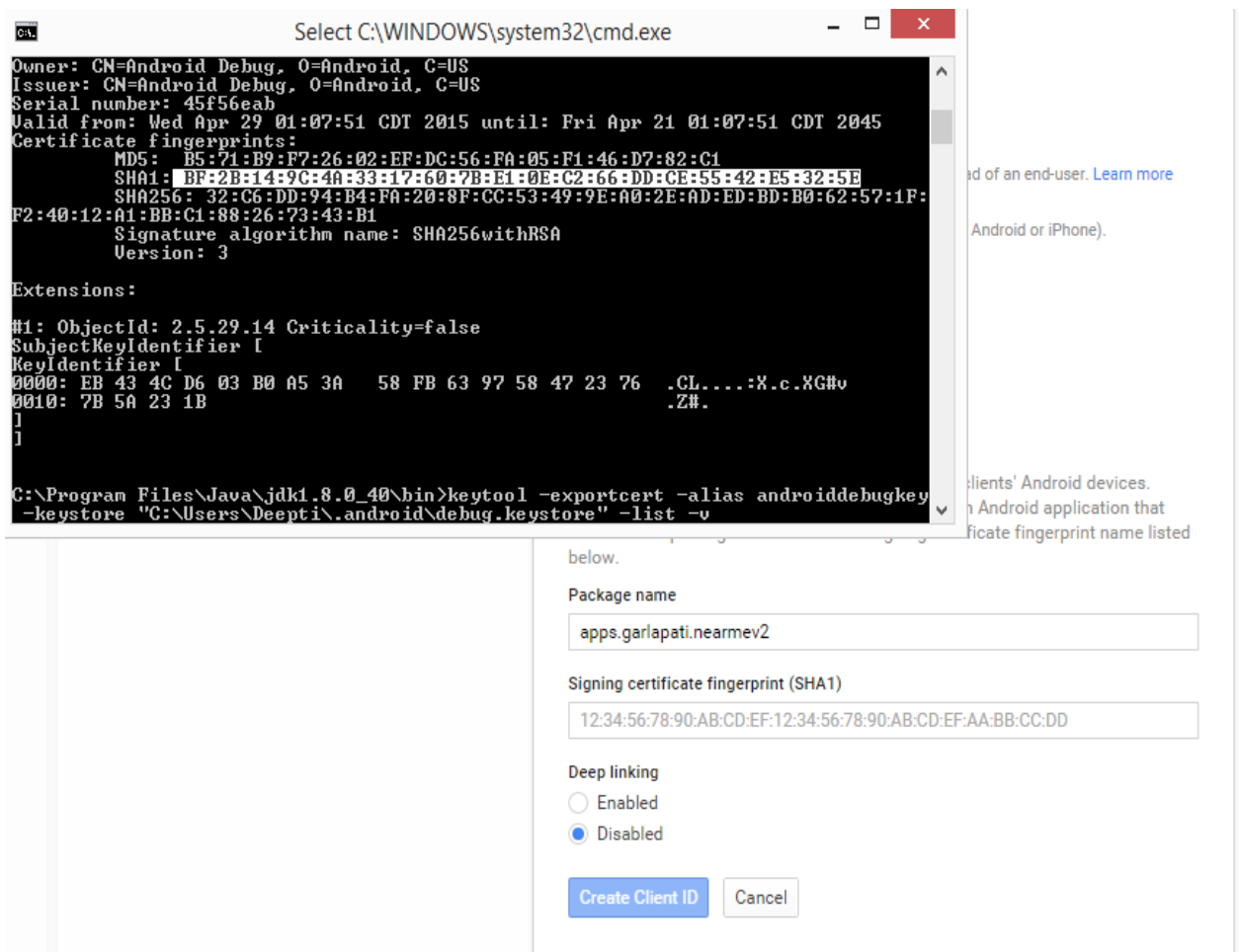


Figure 9.1 Google Places API

Chapter 10 – Testing

10.1 Debugging

Android has inbuilt debugging tool known as Dalvik Debug Monitor Server (DDMS) which stores the error information in the Logcat. It is an internal logging system which aggregates logs across all applications on the phone. For this project I have logged all the user interactions and also fatal when the app crashes. Logcat groups logs based on application name, process id, thread id which are helpful in filtering the logs related to the Near Me application. This helped me to visualize how the different parts of the application interacted with the system internals.

10.2 Unit Testing

Unit testing helps in testing and verifying specific code independently. Each part of the application components are tested to check its functionality in this phase. .

Below are the unit test cases that were run on HTC one M8 for its correctness.

Sr. no	Test Case	Expected Result	Result
1.	On load of application screen	Displays the splash screen and then tabs to Navigate like tasks, near me and stats. Also displays the tasks added before.	Pass
2	On click of Near me	Displays tasks according to the nearby locations associated	Pass
3	On click of Sort	Displays the ways of sorting and sorts accordingly	Pass
4	On click of create new task	Displays a screen to add task name, due	Pass

		date and time, location.	
5	On click of due date	Displays a calendar with date dialog.	Pass
6	On click of due time	Displays a time dialog.	Pass
7	On click of location	Displays the location suggestions	Pass
8	On click of filtering	Displays options to filter tasks and filters accordingly	Pass
9	On click of delete tasks	A dialog box pops up asking to delete the tasks or no.	Pass
10	On click of sync tasks	It displays option whether to sync with drop box	Pass
11	On click of view task	Displays information of the task and gives us an option to delete the task or close	Pass
12	On click of edit task	Displays a screen to edit the task information.	Pass

Table 10.2 Unit Test Cases

10.3 Integration testing

Integration tests for the application are done through Junit test framework.

- The app is tested with various parts of android instrumentation eg. getActivity(), getContext(), Activity switching etc.
- Input fields in the Create new task were tested by programmatically sending dummy text and triggering a save.
- Location notifications are simulated by sending fake GPS co-ordinates to the system and then checking if the notification is displayed or not.

- Dropbox connectivity is tested by uploading a fake 0 byte file and deleting subsequently.

Multiple tasks were created and deleted to see how the app performed with many inserts and deletes.

10.4 Compatibility Testing

This type of testing is a non-functional testing that is done by testing the developed application on the various available devices for compatibility. Near Me android application is developed and is supported by wide range of android devices along with varied screen sizes. This Android application was tested on Android devices like Nexus 4, Moto G, Samsung S4, HTC one M8 and also on Samsung tab 10.1 .The application works perfectly fine with these devices along with bar code scans and attaching images without any glitches.

10.5 Usability testing

This type of testing is a technique that should be used for user –centered interaction design for evaluating an android application on various users for testing purpose. It is a form of a direct input from users of how it works in real time. This application was given to my friends to try and test on, while taking their valuable inputs I could complete it in a more productive way of implementing bar code scanning of the products and also the selling option of the product. Also while taking images, storing the most recent image and it appearing on the main screen is one such change I have made after taking the users inputs.

10.6 Performance testing

10.6.1 Navigation of Screens and its performance.

The screen navigation is performed and performance of the android application is tested using Traceview tool for each screens. This tool helps us to know efficient the application is performing. Below is the response times for each screen navigation.

Sr. no	Type of Screen	Response Time (ms)
1	Splash screen	0.10
2	Tasks view	0.26
3	Near me screen	0.56
4	Add Tasks	0.24
5	View Task	0.22
6	Location suggestion	0.47
7	Clear Tasks	0.58
8	Options Screen	0.15
9	Edit Task	0.20

Table 10.3 Response time for screens

10.6.2 Analysis.

From the above table, we have the performance testing of all the screen navigations and we could say that with these response times, the performance of the application is good.

Chapter 11 - What I have learned in developing this Application

Developing this application has helped me in learning new technologies and has helped me in learning how can it be done by android SDK, how SQLite comes into use for the data storage and implementing it by following SDLC (Software development life cycle).

While implementing the android application, I have had hands on Android studio and learned to develop Android application in it. The new features in the market of how the android applications make it possible to do things in a simpler way has encouraged me to develop this application to track all the To-Do things and also which is a location aware.

I have used Google places API to deal with location suggestions in the application and this experience helped me to even learn Facebook integration where the user needs to sign in to the Android application via Facebook

In order to have the backup data of all the tasks stored by the user, I have provided a Dropbox sync option. This is how the user can upload all the tasks to Dropbox and access whenever necessary.

All these options in Near Me application has helped me in being updated on how they can be integrated in an android application.

Chapter 12 – Conclusion

The present world has evolved so much and everyone is being very much updated with the latest technologies, Android is one among the favorites for many users and usage of Android phone and tablets has become very common. Hence creating applications on Android that help the users to access, track everything on mobile is very helpful. Based on the requirements of the users, many applications can be developed as it's an open source and there are a lot of resources available in the internet to learn about Android and develop the related Android applications.

The main motto of this report is to develop a Near Me Application that helps a user to create To-Do lists and save the locations associated with the To-Do items. These lists can also be synced through Dropbox and Google Drive. In this way, we can have the lists uploaded in cloud with easy accessibility.

The Near me android application was developed initially from the start acquiring all the requirements and following the software development life cycle process. The development was done smoothly meeting all the requirements and runs successfully on android smart phones. This android project helped me learn the current version of android sdk and google maps.

Chapter 13 - Future Work

Future work for this Android application can be done by enhancing and implementing the following features.

- By making this app available for iOS devices as well.
- By implementing an online application that allows the user to login and access the to-do tasks from desktops.
- By adding an image options while creating a new task.
- Also, to provide a share to-do tasks options with other people.

Chapter 14 - References

1. *Android System Architecture* [online] [Cited : January 28 , 2015]
<http://www.tutorials4android.com/android-architecture>
2. *About Architecture layers* [online] [Cited : January 28 , 2015]
http://www.tutorialspoint.com/android/android_architecture.htm
3. *Activity Lifecycle* [online] [Cited : January 30 , 2015]
<http://developer.android.com/guide/components/activities.html>
4. *Android Basics* [online] [Cited : February 5 , 2015]
<https://developer.android.com/training/index.html>
5. *All about intents* [online] [Cited : April 2 ,2015]
<http://developer.android.com/guide/components/intents-filters.html>
6. *Unified Modeling language* [online] [Cited : April 23 ,2015]
http://en.wikipedia.org/wiki/Unified_Modeling_Language
7. *Class Diagrams* [online] [Cited : April 23 ,2015]
http://en.wikipedia.org/wiki/Class_diagram
8. *Custom date picker* [online] [Cited : April 28 ,2015]
<http://developer.android.com/guide/topics/ui/controls/pickers.html>
9. *Location –Aware* [online] [Cited : May 12 ,2015]
<https://developer.android.com/training/location/index.html>
10. *Creating and adding Geofence* [online] [Cited : May 14 ,2015]
<https://developer.android.com/training/location/geofencing.html>

APPENDIX

The complete Android Manifest code of the Near Me Android Application is below:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="apps.garlapati.nearmev2" >

    <permission
        android:name="apps.garlapati.nearmev2.permissions.MAPS_RECEIVE"
        android:protectionLevel="signature" />
    <permission
        android:name="apps.garlapati.nearmev2.permission.C2D_MESSAGE"
        android:protectionLevel="signature" />

    <uses-permission android:name="apps.garlapati.nearmev2.permission.C2D_MESSAGE" />
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-
permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-
permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" /
>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-
permission android:name="com.garlapati.nearme.app.permissions.MAPS_RECEIVE" />
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />

    <uses-feature
        android:glEsVersion="0x00020000"
        android:required="true" />

    <meta-data
        android:name="com.google.android.gms.version"
        android:value="@integer/google_play_services_version" />
    <meta-data
        android:name="com.google.android.maps.v2.API_KEY"
        android:value="AIzaSyCdR-dwU084LAW8IyYdzN3SFvnizdFk1qk" />

    <application
        android:name="com.activeandroid.app.Application"
        android:allowBackup="true"
        android:icon="@drawable/nearme_logo_with_text"
```

```

android:label="@string/app_name"
android:theme="@style/MaterialDrawerTheme.Light.DarkToolbar" >
<activity
  android:name=".Activities.MainActivity"
  android:label="@string/title_activity_main" />
<activity
  android:name=".Activities.SplashActivity"
  android:label="@string/AppName"
  android:noHistory="true" >
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
<activity
  android:name=".Activities.NewTaskActivity"
  android:label="@string/title_activity_new_task"
  android:parentActivityName=".Activities.MainActivity" >
  <meta-data
    android:name="android.support.PARENT_ACTIVITY"
    android:value="apps.garlapati.nearmev2.Activities.MainActivity" />
</activity>
<activity
  android:name=".Activities.EditTaskActivity"
  android:label="@string/title_activity_edit_task"
  android:parentActivityName=".Activities.MainActivity"
  android:windowSoftInputMode="adjustResize" >
  <meta-data
    android:name="android.support.PARENT_ACTIVITY"
    android:value="apps.garlapati.nearmev2.Activities.MainActivity" />
</activity>
<activity
  android:name=".Activities.DetailActivity"
  android:label="@string/title_activity_detail" />

<service
  android:name="com.esri.android.geotrigger.GeotriggerService"
  android:exported="false" />

<receiver
  android:name="com.esri.android.geotrigger.MessageReceiver"
  android:permission="com.google.android.c2dm.permission.SEND" >
  <intent-filter>
    <action android:name="com.google.android.c2dm.intent.RECEIVE" />
    <action android:name="com.google.android.c2dm.intent.REGISTRATION" />

    <category android:name="apps.garlapati.nearmev2" />
  </intent-filter>
  <intent-filter>
    <action android:name="android.intent.action.PACKAGE_REPLACED" />

```

```

        <data
            android:path="apps.garlapati.nearmev2"
            android:scheme="package" />
    </intent-filter>
</receiver>

<!-- database stuff -->
<meta-data
    android:name="AA_DB_NAME"
    android:value="nearmev2.db" />
<meta-data
    android:name="AA_DB_VERSION"
    android:value="9" />
<meta-data
    android:name="com.facebook.sdk.ApplicationId"
    android:value="@string/facebook_app_id" />

<activity
    android:name=".Activities.AlarmActivity"
    android:label="@string/title_activity_alarm" />
<activity
    android:name=".Activities.SignInActivity"
    android:label="@string/title_activity_sign_in" />

<!-- dropbox settings -->
<activity
    android:name="com.dropbox.client2.android.AuthActivity"
    android:configChanges="orientation|keyboard"
    android:launchMode="singleTask" >
    <intent-filter>
        <data android:scheme="db-yx97tx779ptv47m" />

        <action android:name="android.intent.action.VIEW" />

        <category android:name="android.intent.category.BROWSABLE" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
<activity
    android:name=".Activities.SyncActivity"
    android:label="@string/title_activity_sync"
    android:parentActivityName=".Activities.MainActivity" >
</activity>

<receiver
    android:name=".Receivers.BootCompletedBroadcastReceiver"
    android:enabled="true"
    android:exported="true" >
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />

```

```

        <action android:name="apps.garlapati.nearmev2.action.BOOT_COMPLETED" />
    </intent-filter>
</receiver>

<service
    android:name=".Services.NearMeService"
    android:enabled="true"
    android:exported="true" >
    <intent-filter>
        <action android:name="apps.garlapati.nearmev2.Services.RECREATE_ALARMS" />
        <action android:name="apps.garlapati.nearmev2.Services.START_GEO_TRACKING_
SERVICE" />
    </intent-filter>
</service>

<receiver
    android:name=".Receivers.OnAlarmBroadcastReceiver"
    android:enabled="true"
    android:exported="true" >
</receiver>
</application>

</manifest>

```