TOGO CABS - AN ANDROID PHONE CAB RESERVATION APPLICATION

by

NIRUPAMA MRINALINI MEESALA

B.E., Muffakham Jah College of Engineering and Technology, Hyderabad, India, 2008

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

2012

Approved by:

Major Professor
Dr. Daniel Andresen

Abstract

Now a days, people are more inclined towards owning a smart phone. In such a scenario, mobile application development is one of most sought after platforms. Android is one of the largest platforms that run in most smart phones from manufacturers like Samsung, HTC etc.

ToGo Cabs is an Android phone cab reservation application which targets the residents of the state of Kansas. While some cab services boast about their cab being just a call away, some other taxi services boast about the punctuality of their service. Unfortunately, at the end of the day, the passengers are just tired of waiting at different locations for the cab that they have just reserved to pick them up and take them home. What we need is a reliable mobile application which reserves a cab for us from a specific place at a certain time and which tells its users the status of the cab in order to keep them from waiting for long hours. ToGo Cabs serves just that purpose.

ToGo Cabs allows the users to get a cab from any location in the state of Kansas, even if they seem to be lost. One does not have to spend hours on phone with the cab services to tell them where exactly you are located currently. The Global Positioning System takes care of the current location for the users. This application shows the route to the user, which the cab would take to reach to the destination. Once a user reserves a cab, he is acknowledged with a confirmation number which he can further use to check the cab status. The application provides the user the facility to e-mail or text his confirmation number. The application can also set a reminder notification just 15 minutes before the cab pick-up. The user can also check all the trips that he has made so far, from the application.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I take this opportunity to express my sincere gratitude and thanks to **Dr. Daniel Andresen**, my Major Professor for his constant guidance and support through his suggestions and ideas throughout my project. You played a significant role in the successful completion of my project. Dr. Andresen, it has been an honor to work with you.

I would like to express my appreciation to my committee members, **Dr. Mitchell Neilsen** and **Dr. Torben Amtoft** for accepting to serve on my committee. It was a privilege to be able to take courses under you, which provided me with more practical guidance and experience.

I am grateful to **Dr. Gurdip Singh**, the Head of Computing and Information Sciences Department who have guided and offered me his valuable support throughout my Masters. The CIS administrative and technical staff have been very supportive of me and helped me in the successful completion of my graduate study.

The most special thanks go to my best friends. Ranjit, you gave me your unconditional support and care all through my Masters and have been an excellent critic to my project. Surya, thank you for being there for me.

Dedication

To God and my parents. Without you, my life would fall apart.

God, you have given me strength to overcome every obstacle in life. Thank you.

Dad and Mom, you are my role models. Without you, I would not be able to make it.

Chapter 1 - Introduction

ToGo Cabs is an Android Phone Cab Reservation application. The main objective of the project is to provide an easy to use and handy mobile application to the android users which enables them to reserve a cab from one location to any location in the state of Kansas. The application generated Confirmation Number makes the application more reliable. The users can keep track of their reservations, the amount they have spent for each ride and the cab's current status. Additionally, the user can also see where he/she is currently located, see a route of the cab from the source to the destination point, set reminders for a future pickup, send confirmation details to his email id and send an SMS to his mobile phone.

The rest of the paper discusses the motivation behind the project in Chapter 2; the requirement analysis is highlighted in Chapter 3, followed by the system architecture and design in Chapter 4, Android Framework in Chapter 5, then the implementation in Chapter 6, Testing and Logging in Chapter 7, followed by conclusion and future work of the project.

Chapter 2 - Motivation

In the Manhattan Town Center, few months ago, my ride had a flat tire and I was stuck with no other option but a cab to get back home. I called Taxi-4-Less, the only popular cab service in Manhattan, Kansas only to be on the phone call for about 10 minutes. I had trouble telling him where I was exactly located for pick up. Then, I had to keep calling the cab service, from time to time as they were already running late by 30 minutes. The arrogance of the cab driver then added to my frustration. Manhattan has no proper cab service that can serve the customers in an organized fashion. Passengers have no guarantee whether the cab would arrive on time or if it would arrive at all. After reviewing a couple of websites, the idea of an Android Mobile Cab Reservation Application dawned on me. ToGo Cabs is that application which provides the users with an easy to use Cab Reservation interface. This application gives the user the facility to track the current status of the cab, thereby avoiding the frustration and long waits for the cabs.

Chapter 3 - Requirement Analysis

3.1 Requirements Gathering

A Cab Service application required quite a bit of research before coming up with the design. Cab service, in general, requires the availability of cabs in order to serve users' requests. This project was programmed, keeping in mind, the Graphical User Interface at the user end. After reviewing many cab reservation applications available for Android, a list of positives and negatives of the existing applications was made. Most of the applications targeted to serve a particular country or particular state, or in some cases a particular city. The city of Manhattan, Kansas does not have any application that meets the residents' requirements. Having started with the application targeting the Manhattan audience, the area of focus was extended to the major cities in the state of Kansas.

3.1.1 Related Work

To jot down a specified set of accurate requirements, I reviewed some mobile applications present in Android and iOS. Some apps such as CabGrab and CabSense start with showing your current location which wastes certain amount of useful time. The user might not want a cab from his current location. An app like myTaxi does not mention the Drop off address. The passenger is in a dilemma if the cab would take him/her to the requested place once he is picked up. However, the app does provide a lot of options like saving his preferences and card details. 13Cabs is a good application which provides many facilities like wheel chair/ scooter booking, finding points of interest and is not restricted to just cabs. However, the app does not support horizontal orientation. TaxiMagic app lists the cab services along with their phone numbers around your place and the best way to reserve one is to call them, which I wanted to avoid. On the funny side, Taxi Hold'em is one application which makes a whistling sound when launched. Apparently, the app is to hail a cab with a whistle on a busy road. Taxi Mojo is again an app that asks the user to call the cab service to get a cab.

Having a clear idea of what should not be done that frustrates the users and what should be done that makes it easier for a common man to access the app, the next big thing was to decide a platform to work on. Android comes in many platform versions. A survey through the current scenario shows that about 62% of the Android devices use version 2.1 or later. Hence, this project was developed on Android 2.3.3 Gingerbread platform.

3.2 Requirements Specification

3.2.1 Software Requirements

For the development of this project, the following software requirements have been considered.

Development end:

Operating System : Windows 7

Language : Android SDK, Java

Database : SQLite

Tools : Eclipse Helios IDE

Technologies : Java, SQLite, Android, XML, Google Maps API

Debugger : Android DDMS (Dalvik Debug Monitor Service), Android mobile device

Application end:

Framework : Android SDK Version 2.3.3

Network : Mobile network and Internet (cellular or Wi-Fi)

3.2.2 Hardware Requirements

For the development of this project, the following hardware requirements have been considered.

Development end:

Processor : Pentium IV or higher

RAM : 256 MB

Space on disk : 250MB or higher

Application end:

Device : Android phone with version 2.3 or higher

Space to execute : 3 MB

3.3 Feasibility Study

3.3.1 Economic Feasibility

The application is economically feasible as it only requires an android device with Android SDK 2.3 or higher. The application can be downloaded free. However, users should be able to connect to the internet either through cellular or Wi-Fi and should able to receive messages. This would be the only cost incurred on the project.

3.3.2 Technical Feasibility

To develop this application, an internet connection and a database server is required. The application was deployed and tested on Samsung Galaxy S2, thereby making it technically feasible.

3.3.3 Behavioral Feasibility

This application requires no user guidance or manual as it is easy to use and understand and also has a user friendly interface. The application works in accordance with the design.

Chapter 4 - System Architecture and Design

4.1 System Architecture

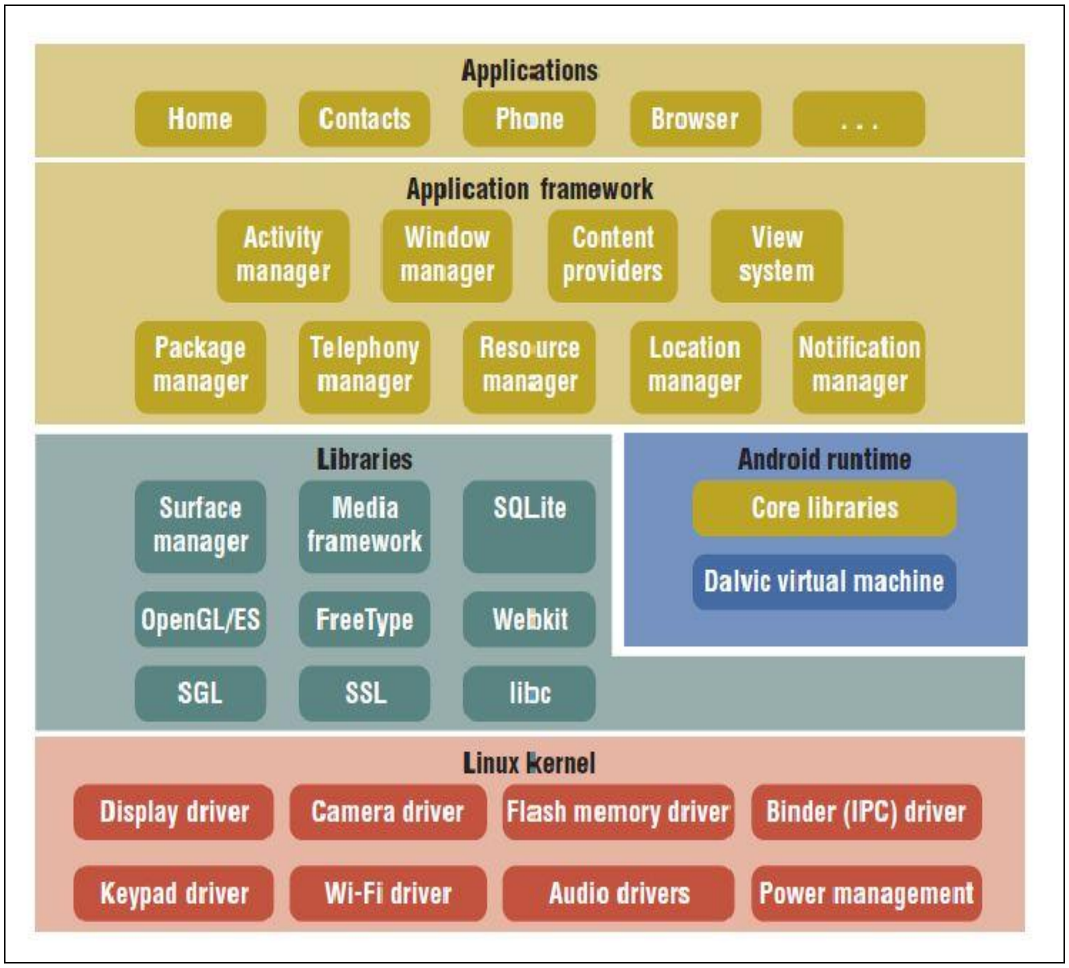


Figure 4.1 Android Architecture

(Word Press, 2011)

The above diagram shows the major components of the Android operating system. In the Android architecture above, it can be observed that the software stack contains the java applications above the Linux kernel. The platform adopts a replace and reuse methodology, which allows the user customizability. Android is a software stack that includes an operating system, middleware and a set of key applications.

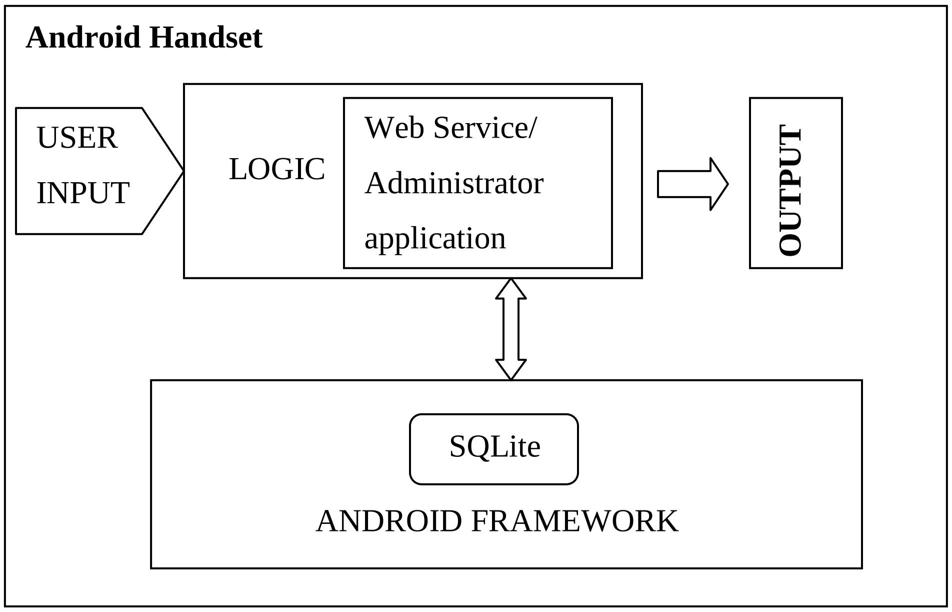


Figure 4.2 System Architecture

The System Architecture can be seen in the above diagram. The application runs on the Android platform present in the Android framework of the Android device. The application interacts via the touch input of the user. When the user touches the screen, the respective action takes place depending on where the user touches and the user is given the next screen to be acted upon or provides any other information to the user. The application interacts with the web service which is installed in the mobile device currently as an administrator application. Depending upon the user input, the availability of the cabs in checked from the administrator application. Once a cab is available, a Confirmation Transaction is generated which is stored in the central database. The generated confirmation transactions are also stored in the local device. The user can see only the transactions which he has generated from that device.

SQLite Database is used to store the transactions. Content Providers are used to store the main central database. Content providers allow the data to be shared among other applications. The local data is stored using the Data Access Objects of the SQLite.

4.2 System Design

After the requirements are gathered, the design of the system is created using UML, Unified Modeling Language. The UML is used to visualize and document the artifacts of the system under development. The two major diagrams of the UML are Use Case diagram and Class diagram. In these diagrams, the major entities of the system are identified and the relation between these entities is highlighted. Each of these diagrams is discussed in detail in the following paragraphs.

4.2.1 Use Case Diagram

A Use Case diagram defines the interactions between the user and the system in order to achieve the desired functionality of the system. This diagram identifies the sequence of actions the user performs.

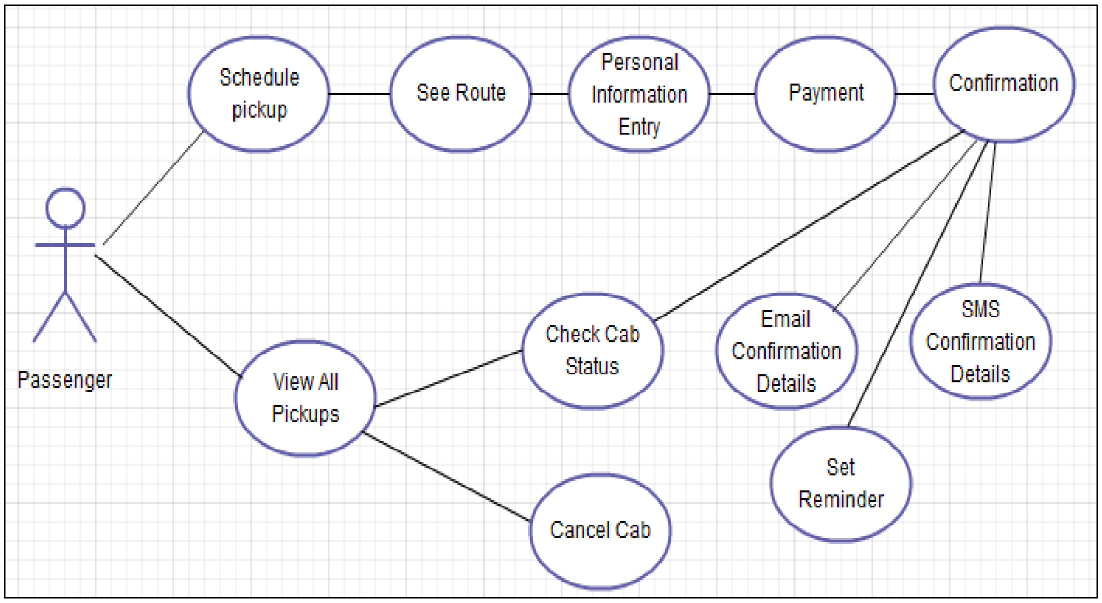


Figure 4.3 Use Case Diagram

The Passenger can either schedule a pickup or view all the pickups history that he has made through the application. After the pickup details like time, pickup place, date, and drop off place are entered, the user is directed to the next page which shows the route the cab takes. Then, the user is asked for his personal information like name, email id, phone number from where the information is redirected to the Payment module. In this page, the user either pays through the application or can select the option of paying the fare to the cab driver. If a cab is available, the user is given a confirmation number through which he can either check cab status or set reminder or send confirmation details to this phone or email. The user can also view his pickup history from the home page, from where he can either check cab status or cancel a cab.

4.2.2 Class Diagram

The Class Diagram describes the structure of the system by showing its classes, their attributes and methods of each class involved in the application. Again, Unified Modeling Language is used to represent the class diagram. The following Class diagram shows the list of classes and their interactions in the package com.ToGoCabs.

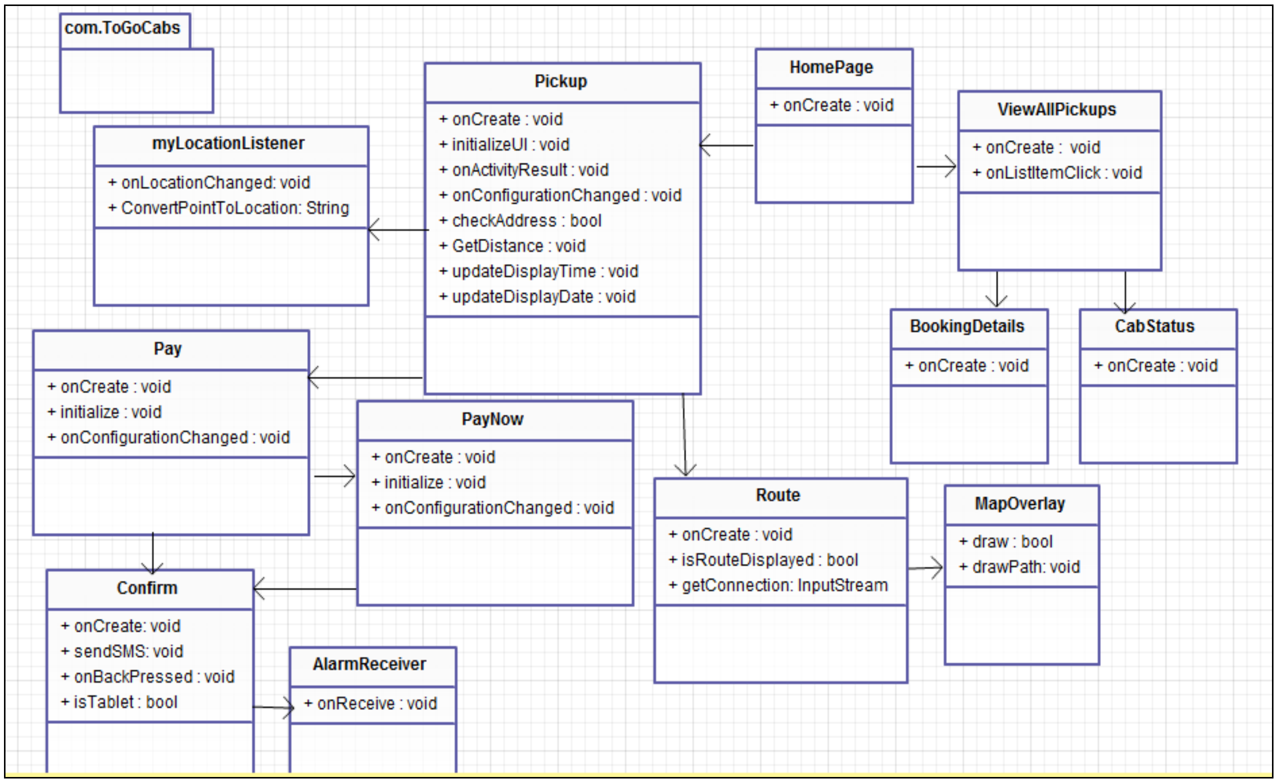


Figure 4.4 Class Diagram (Part I)

The following class diagram shows the classes in the package com.ToGoCabs.dao.

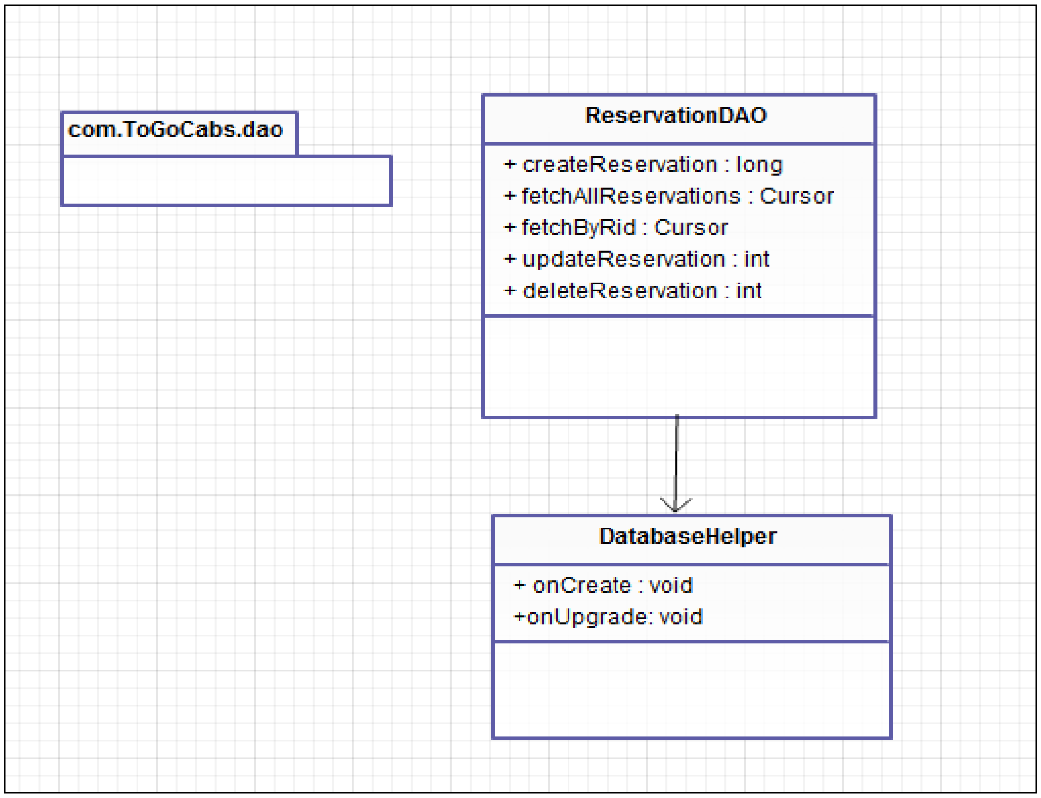

com.ToGoCabs.dao
ReservationDAO
+ createReservation : long
+ fetchAllReservations : Cursor
+ fetchByRid : Cursor
+ updateReservation : int
+ deleteReservation : int
DatabaseHelper
+ onCreate : void
+onUpgrade: void

Figure 4.5 Class Diagram (part II)

The application consists of two packages. The first package com.ToGoCabs consists of 11 classes which contain the logic of the cab reservation along with the GUI of the application. The second package, com.ToGoCabs.dao consists of the access to the database using SQLite. A brief description of the major classes is given in the following paragraphs.

HomePage Activity class is the main activity of the application. It launches the home page of the application from where the user can either schedule a new pickup or view all the pickups that he has made through the application. Lines of Code in this class are 98.

Pickup Activity class takes the pickup date, time, pickup place and drop off place of the user. The user can either manually enter the location from where he wants to be picked up or he can directly access the current location in this screen. The user is also given the facility of choosing GPS or internet to get his current location. The user is also warned that GPS gives a more accurate location when compared to internet. Lines of Code in this class are 810.

Pay Activity class displays the estimated time, distance and the fare for the ride. This screen also shows the route taken by the cab in order to remove the user of all doubts. This could help the user shows the places via which he could travel through the ride. This also makes the reservation more reliable. The user is then asked for his personal details like name, email id, phone number, if at all there would be children travelling, or any further notes to the driver. From this screen, the user can choose between paying through card now and paying to the driver. Lines of Code are 388.

Route Activity class shows the route taken by the cab. It is shortest time and shortest distance based. This activity uses Google Maps Directions API and Overlay classes to draw the route from the pickup place to the dropoff place.

PayNow Activity class is called if the user selects the option to pay for the ride through the application, either through is credit or debit card. In such a case, the user is shown a screen where he can enter his details like card number, expiry date, month, name on card , card type. Lines of Code are 271.

Confirm Activity class shows the confirmation number generated through the application to the user. This screen tells the user that he has had himself reserved a cab from his pickup place. From this screen, the user can email the confirmation and booking details to his emails id or send himself an SMS. Further, the can set a notification reminder to remind him of his cab at least 15 minutes before its pickup time. The user can also check the cab status from this screen. Lines of Code are 225.

CabStatus Activity class shows the current status of the cab to the user. The customer is notified if the cab has been reserved for the customer, if it has been dispatched from the taxi station, if the user has been picked up or if the cab is available for a pickup. Lines of Code are 65.

ViewAllPickups Activity class shows the list of cab reservations that the customer has done through the application in the device. It is local to the device. Lines of Code are 101.

BookingDetails Activity class is called when the user selects an item from the list of all pickups. In this screen, the user is shown his ride details like pickup time, place, date, dropoff place, cab fare, his confirmation number and assigned cab number. The user can also cancel a cab from this screen. Lines of Code are 167.

RoadProvider class uses KML to get the Google maps route from the pickup place to the drop off place. The output KML is parsed to draw the route. Lines of Code are 205.

ReservationDAO class uses the Database Helper class to create a local database of the transactions. It also fetches the history, fetched the details by each confirmation id. It uses the Data Access Object (DAO) of the SQLite database. Lines of Code are 108.

In this project, the administrator application or the main database has also been installed on the device to demonstrate the functionality of the application. In order to make the administrator app and the ToGo Cabs application talk to each other, the central database which holds the details of the cab is stored using SQLite Content Providers.

Chapter 5 - Android Framework Components

An Android application is a package which consists of loosely coupled components which are connected to each other at runtime. The components are the basic building components of any Android application. All these components work together in collaboration by responding to events or Intents or Android system.

5.1 AndroidManifest.xml file

Every Android application has a "manifest" file which contains a list of all activities, intents and permissions which the app includes.

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.ToGoCabs" android:versionCode="1" android:versionName="1.0" >

<uses-sdk android:minSdkVersion="10" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>

<application android:icon="@drawable/cablogo"
android:label="@string/app_name"  >

<uses-library android:name="com.google.android.maps" />
      <activity   android:name=".HomePageActivity"
      android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".PickUpActivity"
            android:windowSoftInputMode="stateHidden"
             android:configChanges="orientation|keyboardHidden">
            <intent-filter>
            <action android:name="android.intent.action.PICKUP" />
            <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".PayActivity"
            android:windowSoftInputMode="stateHidden"
            android:configChanges="orientation|keyboardHidden">
            <intent-filter>
            <action android:name="android.intent.action.PAY" />
            <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".RouteActivity"
            android:windowSoftInputMode="stateHidden">
            <intent-filter>
            <action android:name="android.intent.action.ROUTE" />
            <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".PayNowActivity"
            android:windowSoftInputMode="stateHidden"
            android:configChanges="orientation|keyboardHidden">
            <intent-filter>
            <action android:name="android.intent.action.PAYNOW" />
            <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
         <activity
            android:name=".ConfirmActivity"
            android:windowSoftInputMode="stateHidden">
            <intent-filter>
            <action android:name="android.intent.action.CONFIRM" />
            <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".DenyActivity"
            android:windowSoftInputMode="stateHidden">
            <intent-filter>
            <action android:name="android.intent.action.DENY" />
            <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".NotifyActivity"
            android:windowSoftInputMode="stateHidden">
            <intent-filter>
                <action android:name="android.intent.action.NOTIFICATION" />
            <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
            <activity android:name=".CabStatusActivity"
            android:windowSoftInputMode="stateHidden">
            <intent-filter>
            <action android:name="android.intent.action.STATUS" />
            <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".ViewAllPickupsActivity"
            android:windowSoftInputMode="stateHidden">
            <intent-filter>
            <action android:name="android.intent.action.ALLPICKUPS" />
            <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".BookingDetailsActivity"
            android:windowSoftInputMode="stateHidden">
            <intent-filter>
            <action android:name="android.intent.action.SELECTION" />
            <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
            <receiver android:name=".AlarmReceiver"></receiver>
            <receiver android:name=".ConnectionChangeReceiver"
             android:label="NetworkConnection">
            <intent-filter>
            <action android:name="android.net.conn.CONNECTIVITY_CHANGE"/>
            </intent-filter>
            </receiver>
    </application> </manifest>
```

The AndroidManifest.xml file is the starting point of any application. It mentions the permissions that are installed as part of the application. ToGo Cabs application uses internet, sending and receiving SMS, accessing locations as its permissions. As Google maps are used to show the route map, the application also uses the Google maps library, which is also mentioned in the manifest file. It also includes the broadcast receiver which notifies whenever the connection to the internet is lost in the application.

5.2 Intents

An Intent is an abstract description of action that is to be performed. The function *startActivity()* is used to start an intent. Its most significant is in the launching of different activities, where it plays the role of a connector between different activities. The most important components among the components of an application - activities, broadcast receivers and services are activated through intents. Intents can be divided into two groups. Explicit intents are those intents which are called when the component's designate name is known. On the other hand, implicit intents do not target a name.

The two primary attributes of an intent are action and data. While the action is specified by ACTION_CALL, ACTION_MAIN and other such actions, data is the information that is used when an intent is activated.

5.3 Activities

An Activity is basically a single screen that the user views with a user interface. An application can consist of multiple activities. In ToGo Cabs application, scheduling a pickup is an activity, viewing all pickups is another activity, seeing the route map of the ride is an activity, entering payment information is an activity and many more. Whenever the user touches an appropriate button on the screen, the android system, depending upon the program either launches a new activity or performs another action. When a new activity starts, the old activity is pushed onto the back stack. The old activity is paused and new activity is shown to the user. If the user pressed the back button, the current activity is finished and the old activity which is stored in the stack is called and resumes it activity. However, each activity is monitored by several functions like onPause(), onResume(), onStart() and many more which are a part of the Activity lifecycle.

The list of activities in this application is - HomePage Activity, PickUp Activity, Pay Activity, PayNow Activity, Route Activity, Confirm Activity, Deny Activity, Notify Activity, CabStatus Activity and BookingDetails Activity.

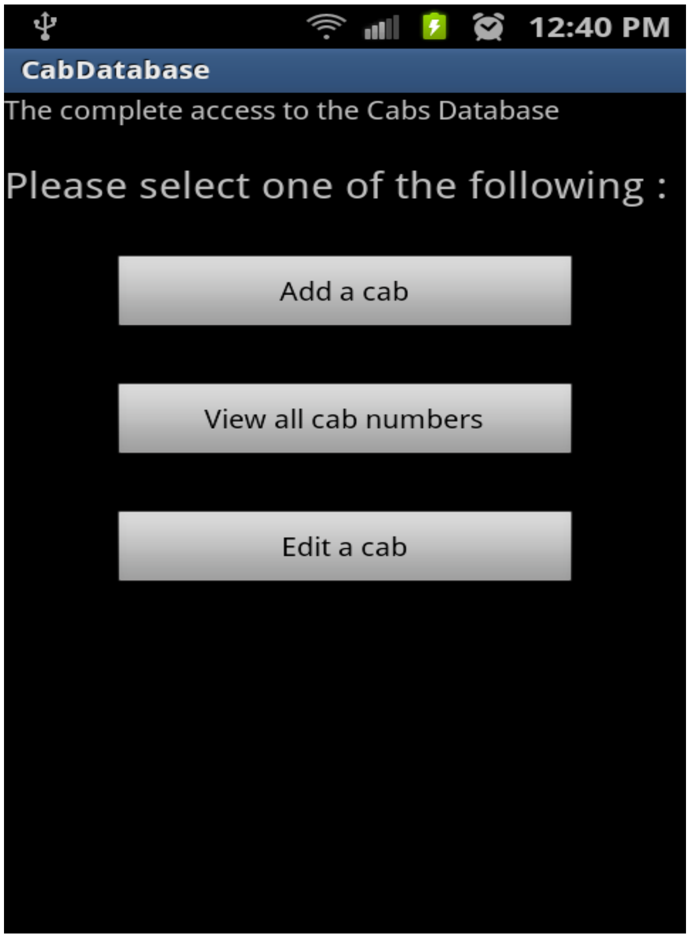
5.4 SQLite Database

SQLite is an open source database which is embedded into Android. SQLite supports standard relational database operations. The major advantage of SQLite is that it requires approximately only 250 Kbytes of memory at runtime. SQLite is available on every Android running device and does not require any additional setup. Data can be accessed, updated, inserted into the tables with simple queries. The ToGo Cabs application uses the table *Reservations* in the database *BookingsDB* through Data Access Object (DAO). The transactions made by a user are stored in the local database called BookingsDB on the device. The Reservations table is accessed via the primary key which is the Confirmation Number.

5.5 Content Providers

Content Providers are the application components that are used to store and retrieve data and make it accessible to the other applications. Some sample content providers are the Contacts Content provider, media list, etc. Content providers are the only way to share data across multiple applications. Each content provider manages data in simple table on a database and has a public URI that uniquely identifies its dataset.

The central database which stores the cabs those are available in the cab station. To make matters simple, the functionality of ToGo Cabs is demonstrated by installing the administrator app in the same device as of the application. In order for the user application to use the central database, an administrator app by name *CabDatabase* is created using Content Provider which is used in the ToGo Cabs application.



The complete access to the Cabs Database

Please select one of the following :

Figure 5.1 Administrator application

Chapter 6 - Implementation

The main objective of the ToGo Cabs application is to provide an easy to use interface for scheduling a pickup by the user and to view all the pickups that the user made using the application. Additionally, the other *main features* of the app include:

- Obtaining the current location
- Showing the estimated duration, distance of the ride
- Showing the route map
- Send details to Email
- Send details to phone via SMS
- Check cab status
- Set a reminder notification
- Canceling a cab

The Android application is developed using the Eclipse Helios IDE. The ADT plug-in for eclipse is the most important tool for this application. Android SDK Version 2.3.3 has been installed for this application.

The business logic is written in Java. The user interface is developed using XML. The user interface is made as simple as possible with no technical details. The user requires no manual or guidance to access the app. Every screen or layout is based upon a common and consistent theme with uniformly sized letters and buttons. The following sections discuss the implementation and user interface each screen in more detail.

This application contains 5437 lines of code including the Java and the xml files. The split up of the lines of code is given as follows.

Language/ Module	**Lines of Code (LOC)**
Java	2794
XML (Portrait and Landscape) + Manifest	1259 * 2 + 125 = 2643
Total	**5437 LOC**

Table 6.1 Lines of Code

6.1 Graphical User Interface

The Graphical User Interface has been kept clean, neat and easy to understand and navigate. The user does not require any memorization of what he done and what he has to do next. The button texts are easy to read and show the purpose of the button. This application has been targeted for the Android mobile phone users and not tablets. The Android tablet does not support sending or receiving SMS because of the lack of cellular network provider. On practical observation, people on the move do not generally carry a device as big as a tablet. Hence, this application is limited for the mobile phone users.

6.1.1 Home Page

The Home Page is the starting screen of the application. It contains a page which mentions the name of the application. The user can navigate to the 2 other screens from this screen. The Pick Me Up button takes the user to schedule a new pick up. The View All Pickups button takes the user to a screen where the list of his transactions is displayed.

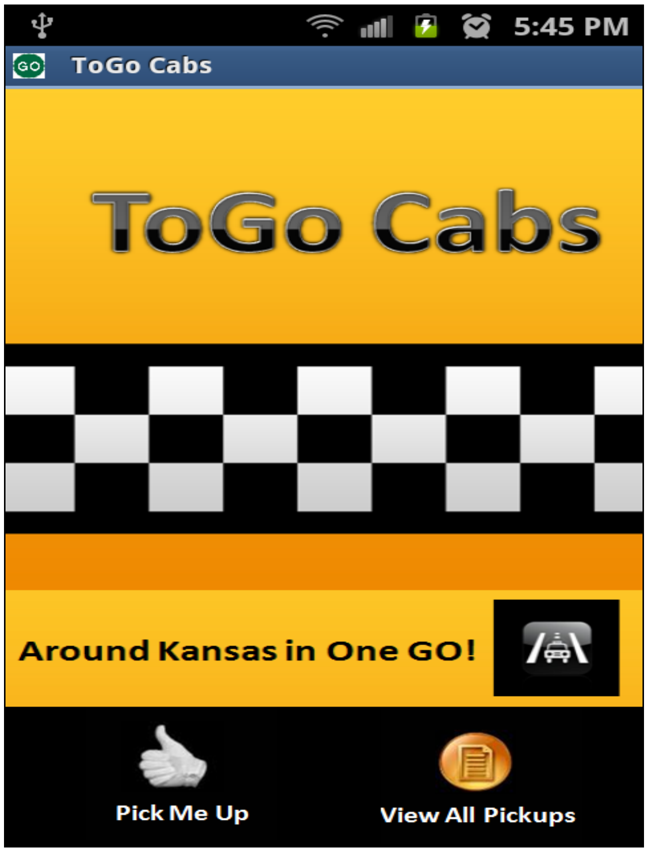
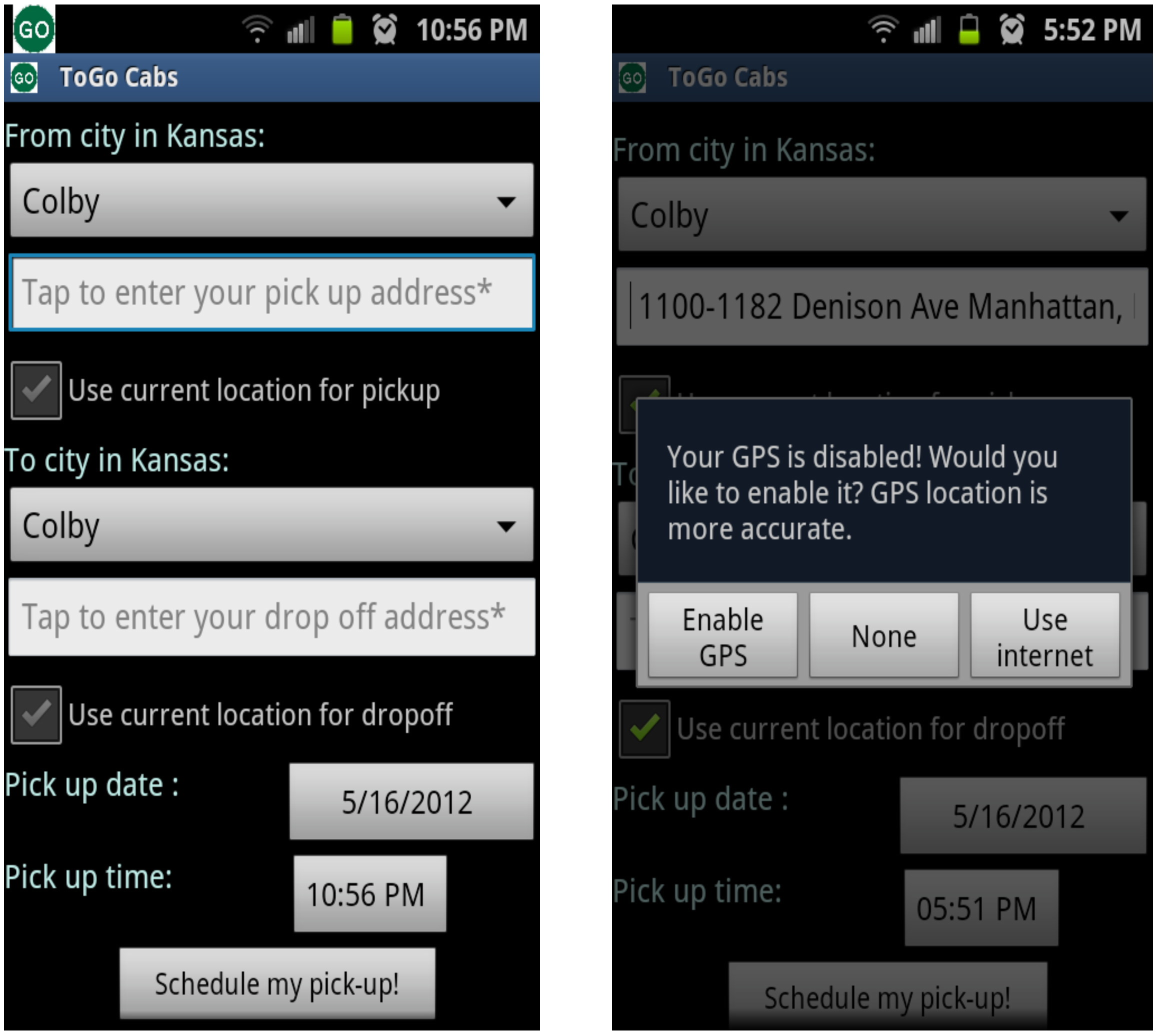


Figure 6.1 Home Page Screen

6.1.2 Pick Up

When the user touches the Pick Me Up button on the Home page, he is redirected to the Pick Up screen. In this screen, the user is asked to enter his pick up location, his drop off location, pick up time and date. Basic validation rules are applied to make sure the user enters all the mandatory fields. The user can either manually enter the place he wants to be picked up from or he can check the *Current Location* box to retrieve his current location. The same applies to the drop off location too. The cities are restricted to the major cities of the state of Kansas.


10:56 PM

ToGo Cabs

From city in Kansas: Colby

Tap to enter your pick up address*

Use current location for pickup

To city in Kansas: Colby

Tap to enter your drop off address*

Use current location for dropoff

Pick up date : 5/16/2012

Pick up time: 10:56 PM

Schedule my pick-up!

5:52 PM

ToGo Cabs

From city in Kansas: Colby

1100-1182 Denison Ave Manhattan,

Your GPS is disabled! Would you like to enable it? GPS location is more accurate.

Enable GPS | None | Use internet

Use current location for dropoff

Pick up date : 5/16/2012

Pick up time: 05:51 PM

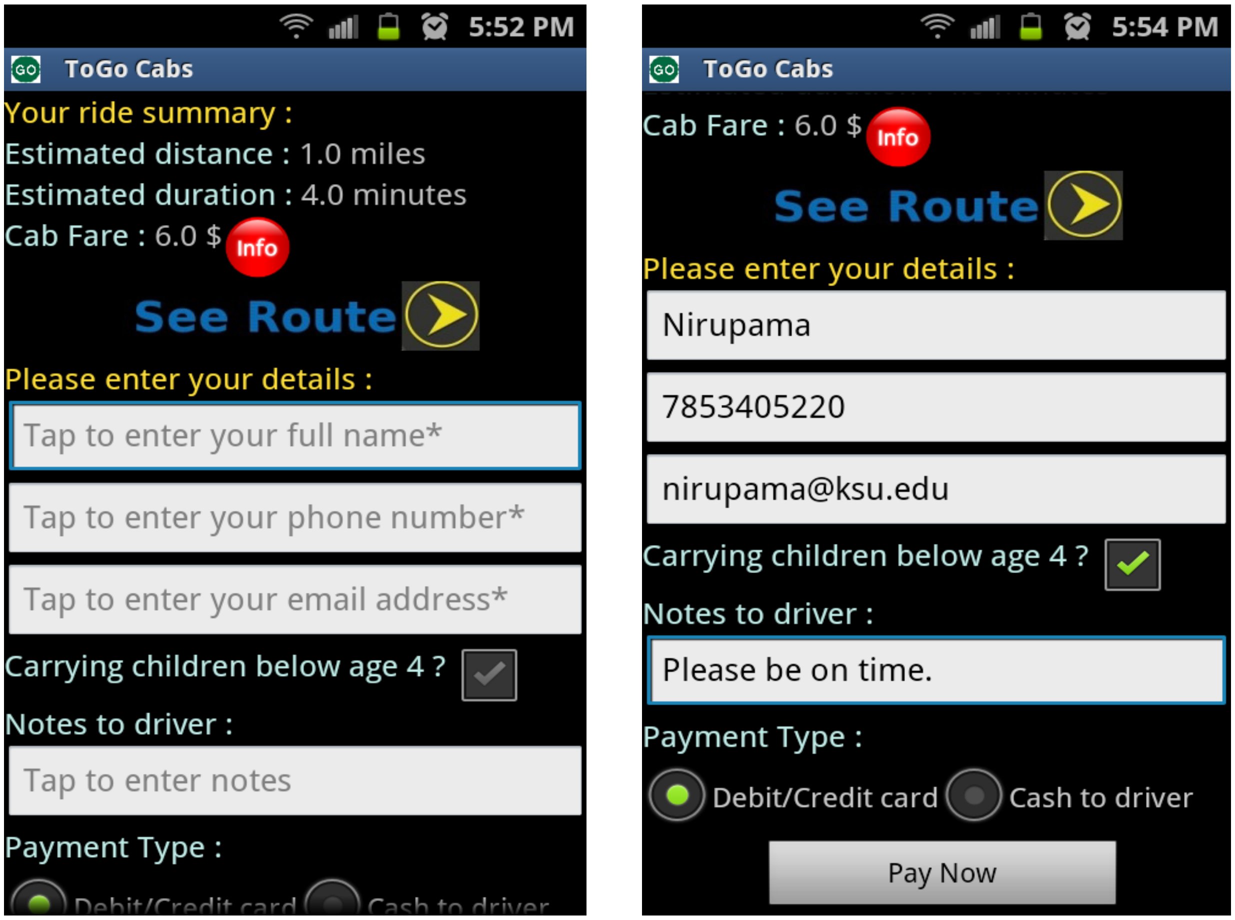
Schedule my pick-up!

Figure 6.2 Pick Up Screen

The current location can be got either by using Global Positioning System (GPS) or through internet. If the device has the GPS already on, the current location is obtained without prompting for any further action from the user. However, if the GPS of the device is not on, the user is prompted to either enable the GPS or use internet to access the current. Location Manager and Location Listener are used to access the current location of the user. Once, all the details are entered by the user, the user is navigated to the next screen.

6.1.3 Route and Personal Details

When the user selects the "Schedule my pick up" button in the Pick Up screen, he is directed to the Route and Personal Details screen. In this screen, the user is shown the estimated time and distance of the ride, along with the cab fare. This screen has a button "See Route" which shows the user the route, the cab takes to the destination from the pick-up place. The screen also takes in the user's personal information like person name, user email address, user phone number, whether the user is bringing children under the age of 4, any additional notes to the driver and a payment option.



5:52 PM
ToGo Cabs
Your ride summary :
Estimated distance : 1.0 miles
Estimated duration : 4.0 minutes
Cab Fare : 6.0 $ Info
See Route
Please enter your details :
Tap to enter your full name*
Tap to enter your phone number*
Tap to enter your email address*
Carrying children below age 4 ?
Notes to driver :
Tap to enter notes
Payment Type :
Debit/Credit card Cash to driver

5:54 PM
ToGo Cabs
Cab Fare : 6.0 $ Info
See Route
Please enter your details :
Nirupama
7853405220
nirupama@ksu.edu
Carrying children below age 4 ? ✓
Notes to driver :
Please be on time.
Payment Type :
[✓ Debit/Credit card] Cash to driver
Pay Now

Figure 6.3 Personal Details Screen

On touching the *See Route* button, the user is redirected to the route map of the ride. The Pick Up point is shown with a blue marker and the destination point is shown with a checked flag symbol. On tapping any of the markers, the user is shown the physical address of the point. The route is drawn using Google directions API where the output KML file is parsed to draw a path using Overlay class.

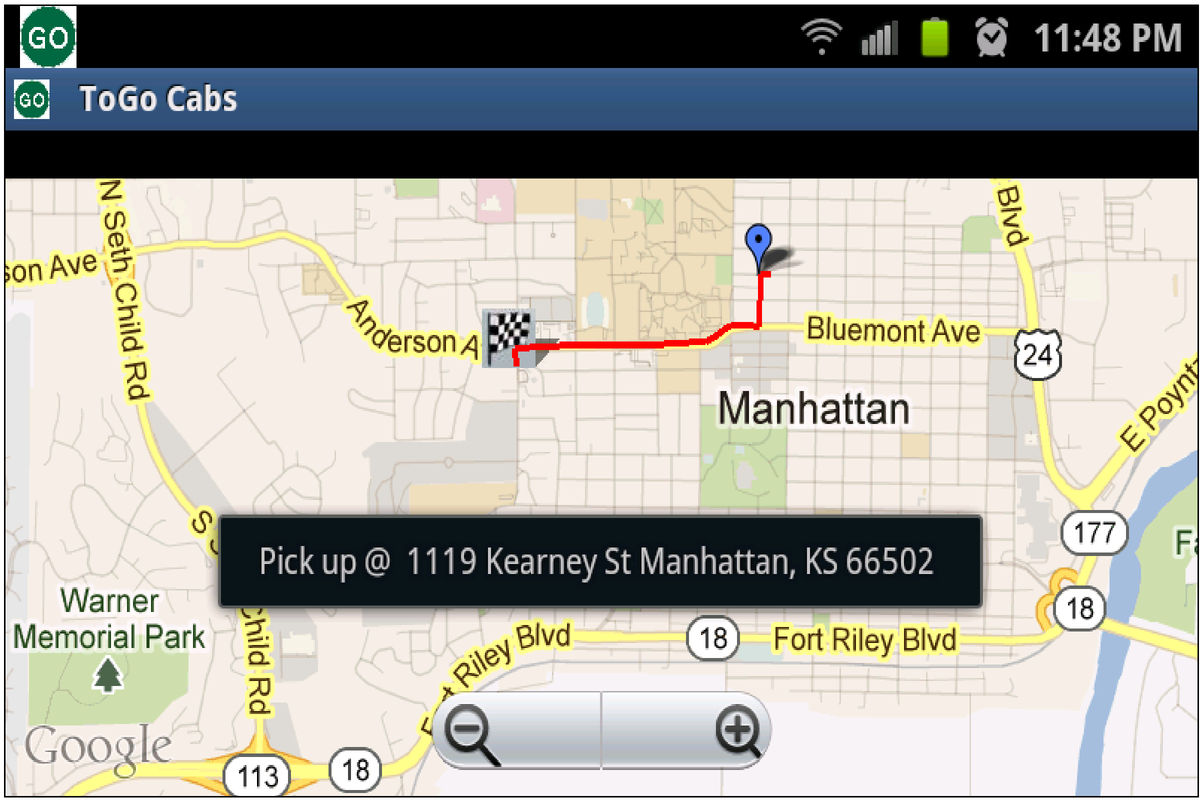
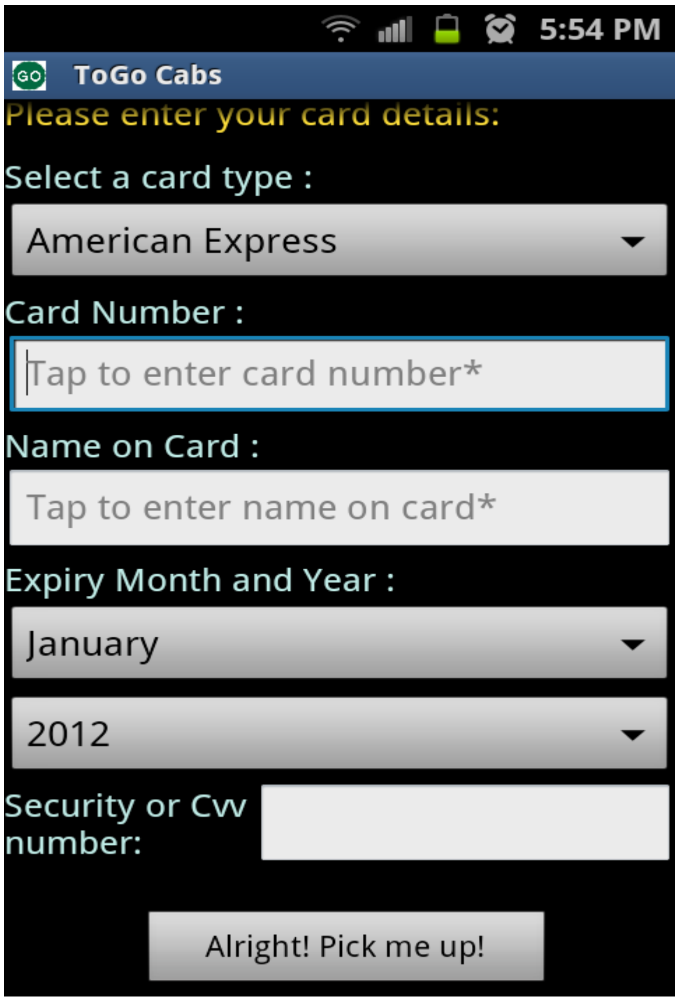


Figure 6.4 Route Map Screen

6.1.4 Payment Screen

In the Personal Details screen, the user is provided with two payment options to choose from. He can either pay to the driver or pay through the application via credit or debit card. If the user selects the *Pay To Driver* option, the user is directed to the confirmation page, which is discussed in the following section. However, if the user touches the *Pay Now* option, the user is directed to the Payment screen. In this screen, the user is asked to enter his card details. The basic mandatory validations and the credit card validations are done in this screen.


5:54 PM

ToGo Cabs

Please enter your card details:

Select a card type : American Express

Card Number : Tap to enter card number*

Name on Card : Tap to enter name on card*

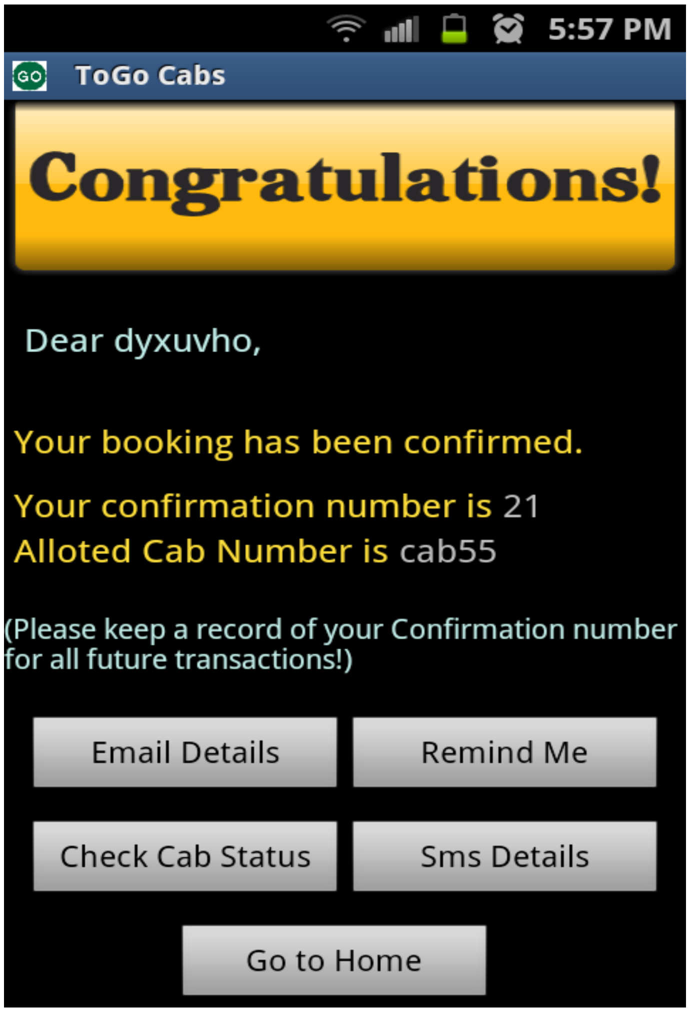
Expiry Month and Year : January 2012

Security or Cvv number:

Figure 6.5 Payment Screen

6.1.5 Confirm Screen

After the payment screen, the central administrator app is contacted to see if they are cabs available at that time for a pick up. If at least one cab is available, then the reservation is confirmed and a Confirmation number is generated and displayed to the user in the Confirm Screen. The user is also displayed the assigned cab number. In addition, the user can SMS the details to his phone, email the details to his email id, check cab status. Further, the user can set a notification to remind him 15 minutes before the pickup time. The user should be careful not to close the application or switch the phone off in order for the notification to work correctly.



5:57 PM

ToGo Cabs

Congratulations!

Dear dyxuvho,

Your booking has been confirmed.

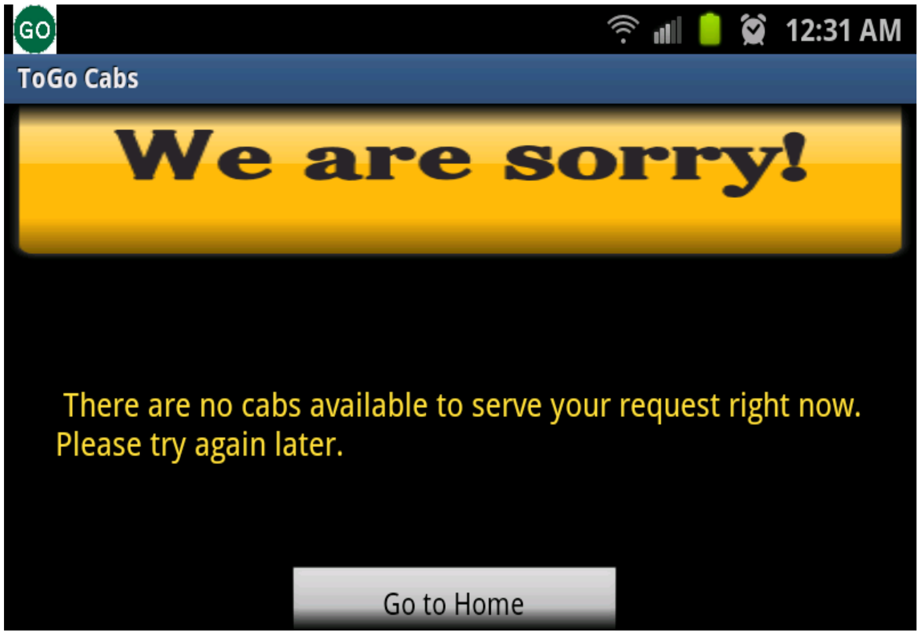
Your confirmation number is 21
Alloted Cab Number is cab55

(Please keep a record of your Confirmation number for all future transactions!)

Figure 6.6 Confirm Screen

6.1.6 Deny Screen

If the cabs are not available for the customer at the desired pick up time, the user is displayed a screen asking him to try again later. The user can then go to the home screen from this screen by clicking on the *Go To Home* button.



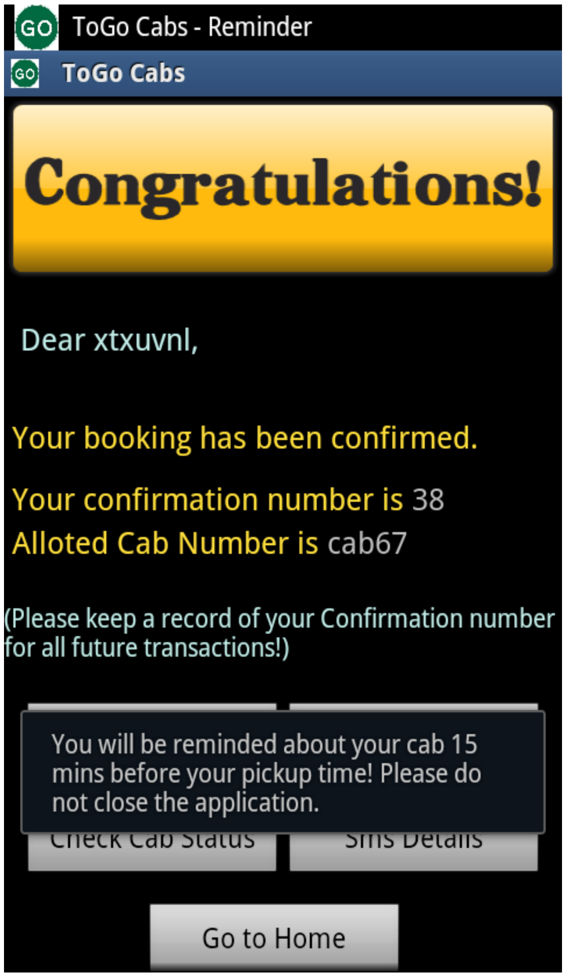
There are no cabs available to serve your request right now. Please try again later.

Figure 6.7 Deny Screen

6.1.7 Notify Screen

The user can set the reminder to remind him at least 15 minutes before his pickup time. Whenever the notification is set, the user is displayed a message, asking him to let the application remain open and not to switch off the mobile(which is a limitation of the application), in order for the notification to be activated. When the *Remind Me* button is touched or tapped in the Confirm screen, an alert message is displayed to customer about the reminder that has been set.

When the time is 15 minutes before to the pick-up time, the user is alerted about his next pick up which is 15 minutes away. Additionally, he is alerted with a notification sound which keeps buzzing until the user attends to it.



ToGo Cabs - Reminder

ToGo Cabs

Congratulations!

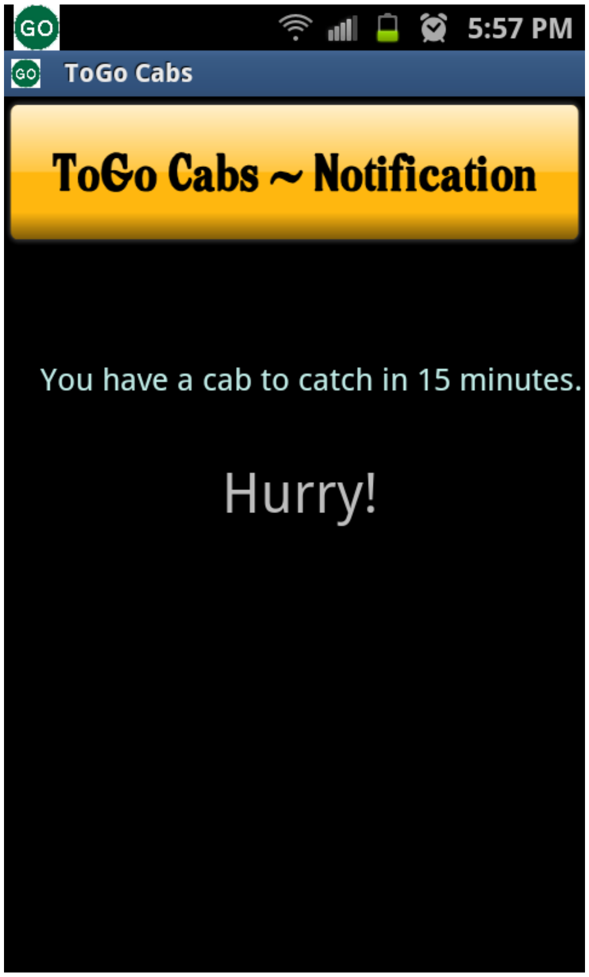
Dear xtxuvnl,

Your booking has been confirmed.

Your confirmation number is 38
Alloted Cab Number is cab67

(Please keep a record of your Confirmation number for all future transactions!)

You will be reminded about your cab 15 mins before your pickup time! Please do not close the application.



ToGo Cabs ~ Notification

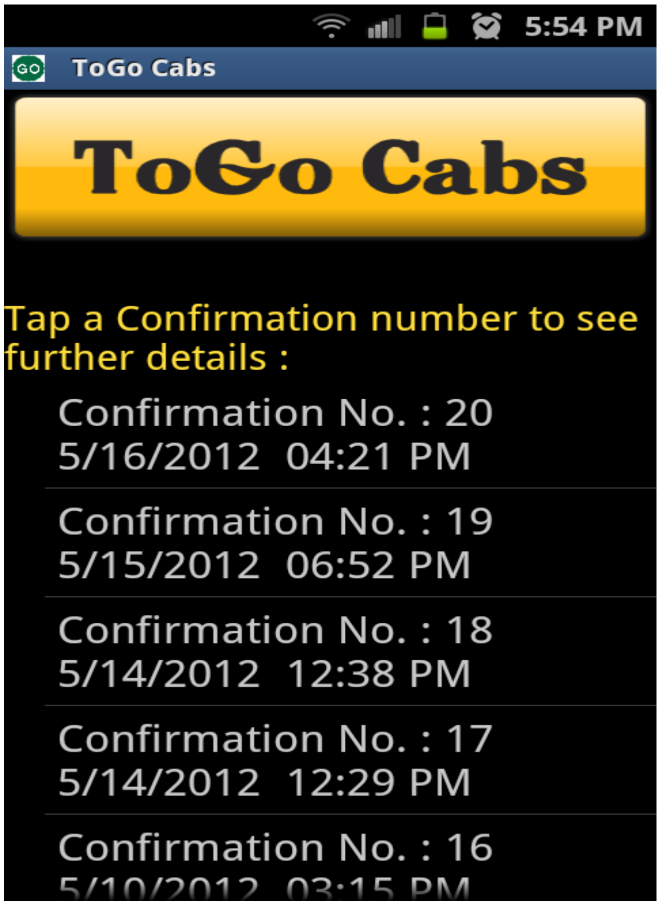
You have a cab to catch in 15 minutes.

Hurry!

Figure 6.8 Notify Screen

6.1.8 All Pick Up History Screen

The Home Page screen is the source to navigation to scheduling a pick up or view all pickups made by the user from the application. The screen "View All Pickups" displays the list of confirmed reservations made by the user. The list is ordered by the descending order of Confirmation Id numbers. Each reservation is shown by its Confirmation number, pickup date and pick up time. Each item in the list is selectable.


5:54 PM

ToGo Cabs

ToGo Cabs

Tap a Confirmation number to see further details :

Confirmation No. : 20
5/16/2012 04:21 PM

Confirmation No. : 19
5/15/2012 06:52 PM

Confirmation No. : 18
5/14/2012 12:38 PM

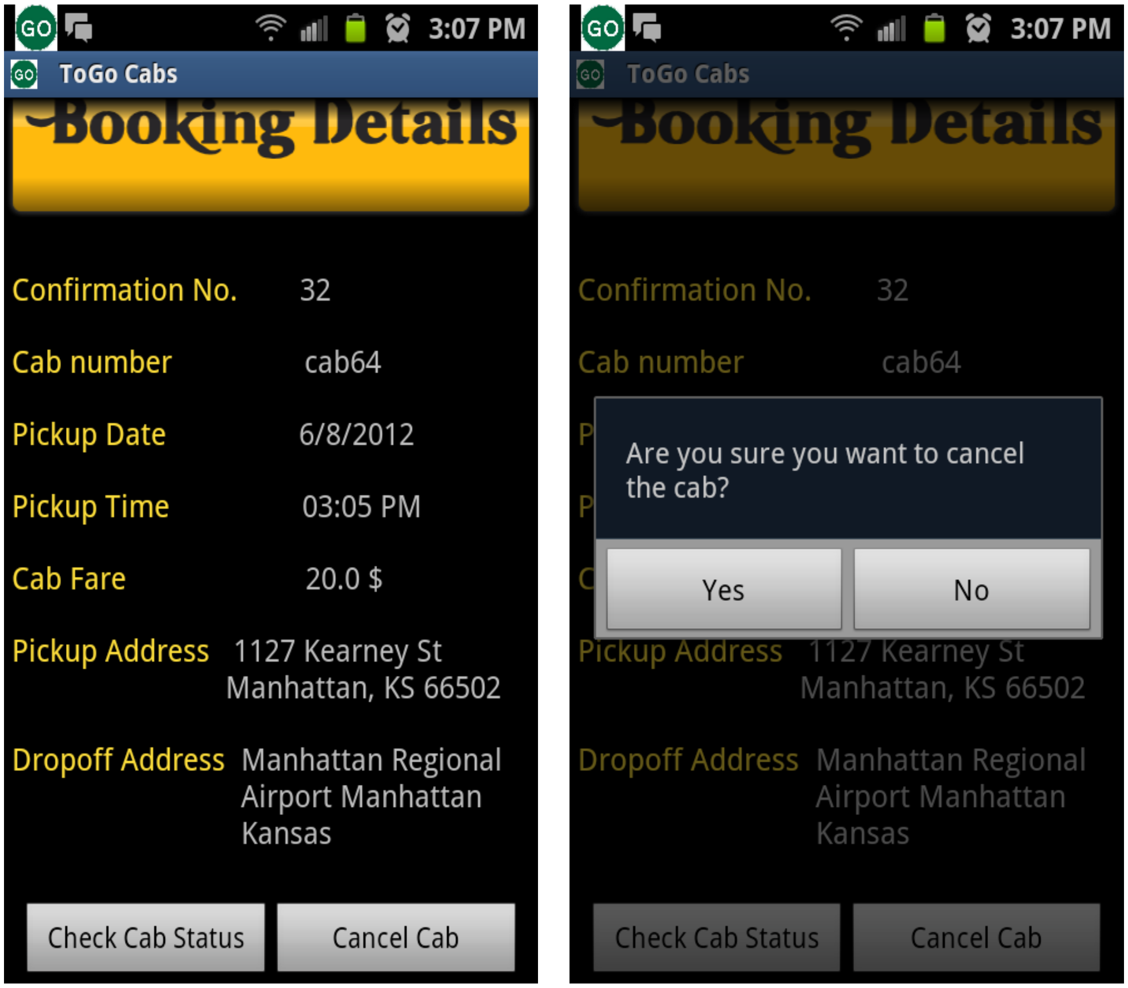
Confirmation No. : 17
5/14/2012 12:29 PM

Confirmation No. : 16
5/10/2012 03:15 PM

Figure 6.9 View all pick-ups Screen

6.1.9 Booking Details Screen

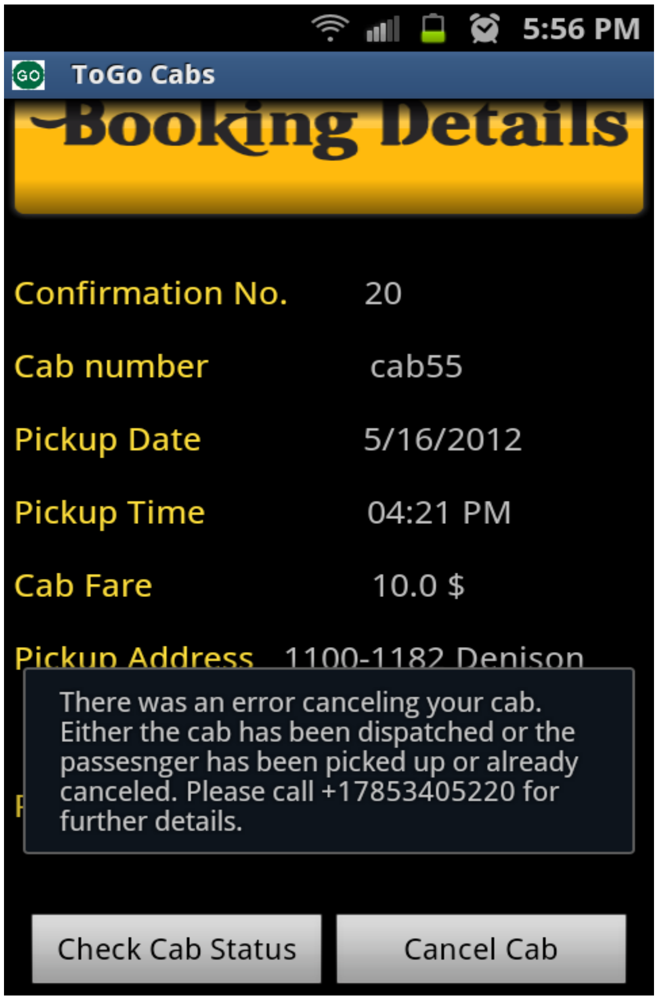
From the Pickups history screen, the user can select any reservation from the displayed list. On selecting any reservation, the user is displayed all the details of his reservation like pick up date, time, pick up place, drop off place, cab fare, Confirmation number and the assigned cab number. The screen also shows two buttons, Check Cab Status and Cancel a Cab. The user can cancel a cab if he changes his mind on his reservation. When the user touches the *Cancel Cab* button, he is displayed an alert message for confirmation. If the cab has already been dispatched, the user is not allowed any cancelation and is displayed a message suggesting him to call the taxi service for further inquiry. However, the cancelation is allowed if the cab is in the state of assignment only and not dispatched.



Confirmation No.	32
Cab number	cab64
Pickup Date	6/8/2012
Pickup Time	03:05 PM
Cab Fare	20.0 $
Pickup Address	1127 Kearney St Manhattan, KS 66502
Dropoff Address	Manhattan Regional Airport Manhattan Kansas

Are you sure you want to cancel the cab?

Figure 6.10 Booking Details Screen


5:56 PM

ToGo Cabs

Booking Details

Confirmation No. 20

Cab number cab55

Pickup Date 5/16/2012

Pickup Time 04:21 PM

Cab Fare 10.0 $

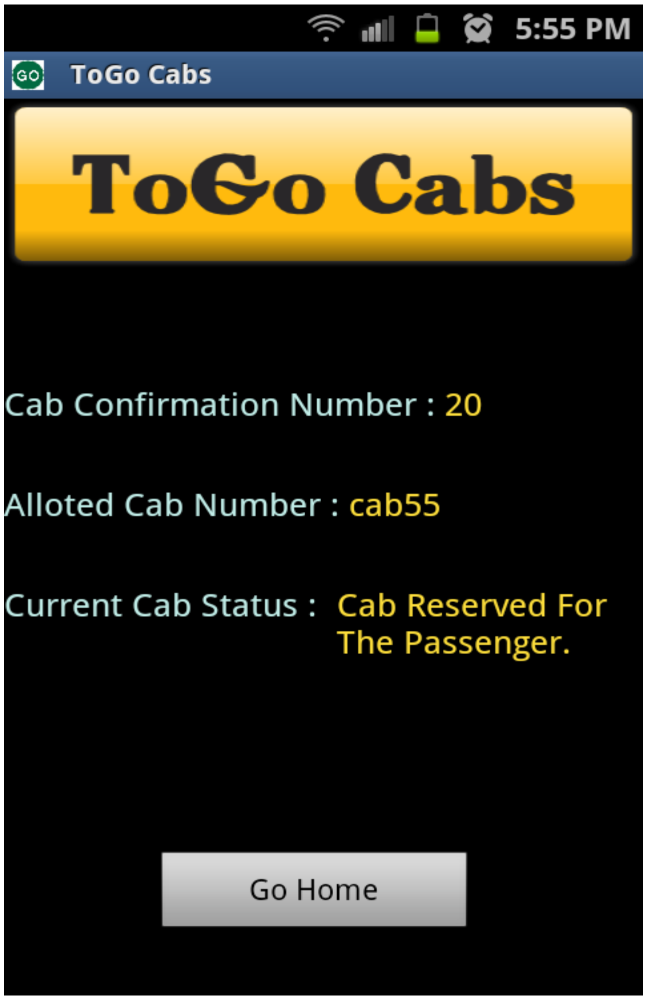
Pickup Address 1100-1182 Denison

There was an error canceling your cab. Either the cab has been dispatched or the passesnger has been picked up or already canceled. Please call +17853405220 for further details.

Figure 6.11 Cancelation Error Screen

6.1.10 Cab Status Screen

The user can check the status of the cab that he has reserved in the Cab status screen. The cab status screen can be accessed either from the Confirmation Screen or by selecting a reservation from the history screen. On selecting an item, the user is shown the current status of the cab which is either the Cab is available for reservation, dispatched to the customer, user has been pickup or assigned to the customer. The status of the cab allows the user to know the status of the cab facilitating the user to avoid long hours of waiting and frustration.



5:55 PM

ToGo Cabs

ToGo Cabs

Cab Confirmation Number : 20

Alloted Cab Number : cab55

Current Cab Status : Cab Reserved For The Passenger.

Figure 6.12 Cab Status Screen

Chapter 7 - Testing and Debugging

7.1 Logging and Debugging

Every Android application runs in its own process. Android comes with a default debugging and logging tools called Dalvik Debug Monitor Server (DDMS). This server provides screen capturing of the device, logcat, location spoofing, etc. DDMS is integrated into Eclipse and works with both the Emulator and the Android device. Android Debug Bridge (adb) is tool that helps in the communication with the android device or the emulator via USB cable. Additionally, logcat dumps a series of system messages such as stack traces, print statements and other error messages which allow us to debug the program more easily.

7.2 Performance

In a particular survey that I conducted around in Manhattan, by talked to some people, it was observed that people requiring cab services are generally a set of people who are in a hurry to get to their destination. In such a time, people do not wish to waste a lot of time waiting for the loading of the application screen. They want to book a cab faster and in a smart way. ToGo Cabs has been designed to improve the efficiency of the application. Care has been taken to minimize the waiting time while loading the application or while performing an activity after clicking or touching a specific button.

To improve the performance of the application, the application does not access the user's current location unlike other apps until and unless user wishes to. Secondly, the current location accessing is not available on the Home Page of the application. This saves time by not accessing the current location whenever the user comes to the home page. The current location can be found out either by using the GPS or through internet. The user is provided an option to choose from either of these to access his current location. Using internet makes the current location access much faster when compared to using GPS to gauge the current location. However, it must be noted the location obtained via GPS is more accurate when compared to the location obtained via internet. By providing an option to choose between GPS and internet, the application again saves a lot of time for the user.

The application has been designed in such a way that the user enters only the information needed to reserve the cab and nothing extra. Unlike many other apps which take in a lot of information and then say that they could not serve the request, this app takes the information primarily needed for the cab first, like pick up time, location, date and drop off location, gives the fare, distance and time to the user. If the user decides to proceed further and he is okay with the price, the estimated time and the route taken by the cab, then he can enter his personal information and proceed to pay. This way user is removed the question of doubt. To add, the route and time displayed to the user is the shortest distance and shortest time based on Google maps.

To test the performance on the application, the time to make the transition from one activity to another was checked manually. The time between the click of a button and the time to display the activity screen to the user was taken into account to find the time taken by the application to load the screen. The following table displays the average time taken to load the screens to the user.

Screen	Time to load in Seconds
Pick Up Details Screen	0.024
Personal Details Screen	0.818
Route Map Screen	0.466
Payment Details Screen	0.072
No available cabs Screen	0.187
Confirmation Screen	0.250
All Pickups History Screen	0.012
Booking Details Screen	0.056
Cab Status Screen	0.054

Table 7.1 Loading Times of Screens

7.3 Unit Testing

In Unit testing, each module of the application is tested separately and individually without the interference of the other modules. Unit testing is done manually to find bugs and to test the functionality of the application. The manual unit testing was done using the Samsung Galaxy S2 Phone. The testing was done manually with the help of the test cases below.

S.No	Screen	Test Case	Expected Result	Result
1	Home Page	Check the buttons *Pick Me Up* and *View All Pickups*	Each of these buttons open up the Pick Up screen and screen with a list of confirmation numbers respectively	Pass
2	Pick Up	Check the box for current location for pick up and drop off locations	The current location is displayed in the boxes either using GPS or internet depending upon user's choice	Pass

3	Pick Up	Enter the address manually in pick up and drop off locations and touch the *Schedule* button	No error on the locations is found	Pass
4	Pick Up	Enter an invalid address or address outside Kansas	Error displayed saying the location is not found	Pass
5	Pick Up	Mandatory fields validations	Error is displayed to the user asking him to enter the mandatory fields	Pass
6	Personal Details	Tap the *See Route* button	The route map from the source to the destination is displayed	Pass
7	Personal Details	Mandatory fields validations	Error is displayed to the user asking him to enter the mandatory fields	Pass
8	Personal Details	Phone number, emails id format validations	Error is displayed if the phone number or email address format entered is invalid	Pass
9	Payment Details	Mandatory fields validations	Error is displayed to the user asking him to enter the mandatory fields	Pass
10	Payment Details	Credit Card Number and CVV number format validations	Error is displayed if the Credit card number or CVV format entered is invalid	Pass
11	Confirm Screen	The confirmation number, assigned cab numbers are displayed to the user.	The confirmation details are displayed to the user	Pass
12	Confirm Screen	User clicks "Remind Me", "Send Email" and "Send	On clicking *Remind Me* button, a notification is set	Pass

		"SMS".	and message about the set notification is displayed to the user. On touching *Send Email* button, the confirmation details are emailed to the user. On touching the *Send SMS* button, the details are sent to the user's phone number	
13	View All Pick Ups Screen	The screen shows a list of confirmation numbers	Each confirmation number should be followed by a pickup date and pick up time	Pass
14	Booking Details Screen	The Booking Details screen should show the confirmation number, assigned cab number, pick up date, time, pickup place, drop off place and cab fare.	All the reservation details are shown to the user	Pass
15	Cab Status Screen	The screen should display the Confirmation number, assigned cab number and the current status of the cab	The cab status is displayed to the user	Pass
16	Cab Status Screen	Touch the "Cancel Cab" button	If the cab is assigned to the user, it is canceled and a success message is displayed to the user. If the cab is dispatched or	Pass

			already canceled, an error is displayed to the user	
17	Notify Screen	The screen displays a message about his upcoming pick up	The notification alerts the user with an upcoming pick up	Pass

Table 7.2 Unit Testing Table

7.4 Integration Testing

After testing of the individual modules, all the modules are made to communicate in collaboration and tested if they work successfully on integration or not. This testing is important to make sure the navigation among the modules works according to the expected behavior. The integration testing was done with the help of the following test cases.

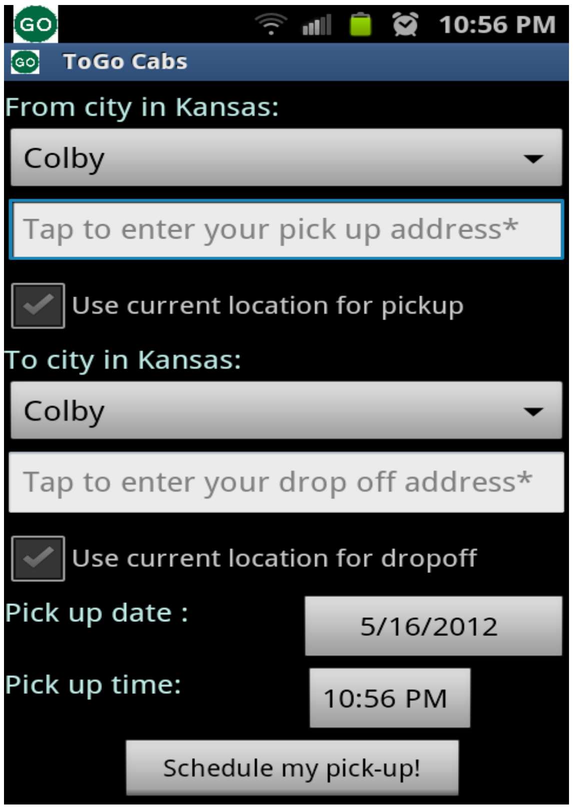
S.No	Test Case	Expected Result	Result
1	The "Schedule my pick up" button action in the Pick Up screen	This action should result in the Personal Details section being displayed with the details about the cab fare, distance, time and the map with pick up and drop off places given in the pickup section.	Pass
2	Pick up Screen - Select Debit/Credit Card	The button changes to "Pay Now", on clicking which the user is taken to the Payment Details screen	Pass
3	Pick up Screen - Select Pay to Driver	The button changes to "Alright Pick me Up" and takes you to either the Confirmation screen or the Cabs unavailable screen	Pass
4	Confirm Screen - Touch the Check Cab Status button	The cab status screen is displayed to the user along with his confirmation number	Pass

		and assigned cab number	
5	All Pick Ups History Screen - Select any reservation	The user is displayed the entire confirmation information on selecting a reservation.	Pass
6	Booking Details Screen - Touch the Check Cab Status button	The cab status screen is displayed to the user along with his confirmation number and assigned cab number	Pass

Table 7.3 Integration Testing Table

7.5 Compatibility Testing

The device has been tested for different screen sizes as part of compatibility testing. This application has been basically designed to support the Android mobile phones. That is because, firstly, the Android Emulator does not work with Wi-Fi and does not provide SMS support. Secondly, this app has been designed for people on the go. Assuming that people carry mobile phones during travelling or shopping, this app would definitely serve their purpose. The application has also been tested for the orientation changes. Whenever the orientation of the device changes, the android's framework reloads the entire activity thereby taking a lot of time and the data entered during the previous orientation is lost in the current orientation. In order to avoid that, the page is not reloaded but instead, displays the previously saved state with the new layout.


10:56 PM

ToGo Cabs

From city in Kansas:

Colby

Tap to enter your pick up address*

Use current location for pickup

To city in Kansas:

Colby

Tap to enter your drop off address*

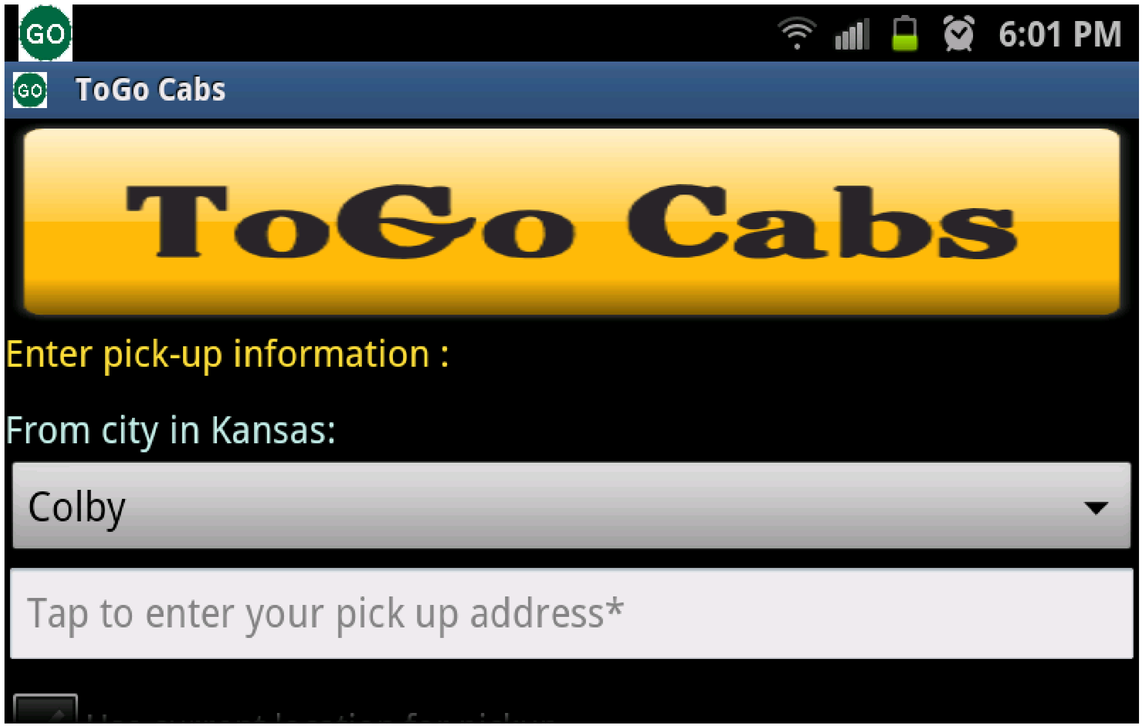
Use current location for dropoff

Pick up date : 5/16/2012

Pick up time: 10:56 PM

Schedule my pick-up!

Figure 7.1 ToGo Cabs in Portrait Mode


6:01 PM

ToGo Cabs

ToGo Cabs

Enter pick-up information :

From city in Kansas:

Colby

Tap to enter your pick up address*

Figure 7.2 ToGo Cabs in Landscape Mode

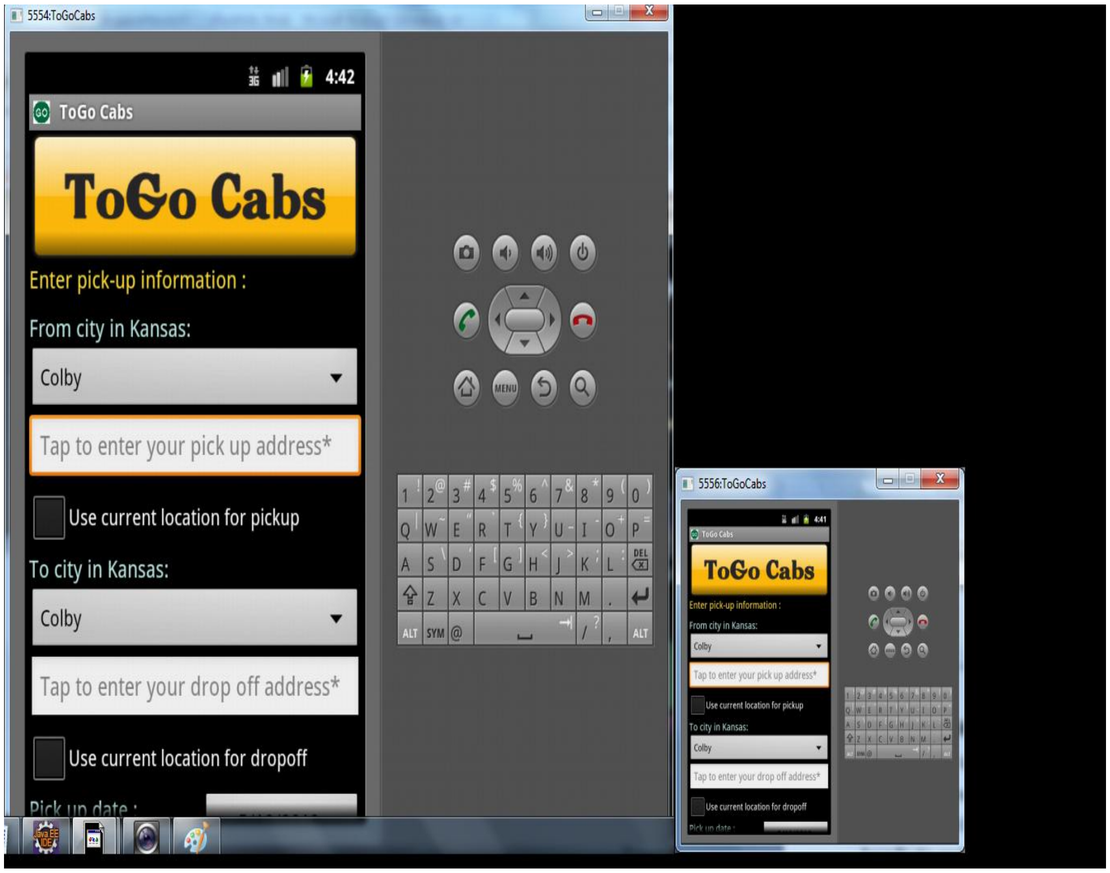

5554:ToGoCabs
4:42
ToGo Cabs
ToGo Cabs
Enter pick-up information :
From city in Kansas:
Colby
Tap to enter your pick up address*
Use current location for pickup
To city in Kansas:
Colby
Tap to enter your drop off address*
Use current location for dropoff
Pick up date :
5556:ToGoCabs
4:41

Figure 7.3 Different Screen Sizes

Chapter 8 - Conclusion

Android now provides a neck to neck competition with iOS. Having been a user of both the technologies, both seem to enhancing and adding in new features to make their products more stable in the market. This project development has put the curiosity of mobile development to rest. My level of study has been increased. I have developed a ToGo Cabs online website as part of my Implementation project (CIS 690) on Visual Studio 2010 and C#. Working with Android to develop an application to suffice the same requirements is a whole new experience.

I have tested this application on the mobiles of my friends and this applications works in the expected successful pattern.

Chapter 9 - Future Work

The current application has been built bearing in mind the current demands and requirements of a decent mobile application that would get a cab to the required location with a minimal waiting time and minimal frustration. This application meets those requirements. As part of future enhancements, this application can be extended in an extensive basis. This application has a large scope of enhancement.

Firstly, the user could be provided t save his locations by adding them with a unique name and whenever he wants a cab from that location, he could just use that name instead of typing the entire address. The same could be done with the credit or debit card details.

Secondly, the user can be provided with an option of viewing the total expenditure he made on the cabs in a week, month or a year. This is just an option of showing his total investment on cab services.

Thirdly, the application can be extended to support the Android tablets.

Fourthly, online cab tracking system can be implemented in which the user could actually where the cab currently is and can see the cab as it appears closer to the destination.

Fifthly, the application does not support a login feature currently because assuming that people are in a hurry when they want a cab to take them somewhere, it would be annoying to login every time to reserve a cab. This can be done if the credit or debit card saving is provided to ensure more security to the application user.

Chapter 10 - References

[1] Google Maps in Android

http://developer.android.com/resources/tutorials/views/hello-mapview.html

[2] Android Coding

http://stackoverflow.com/

[3] Android Development Tutorial

http://www.vogella.com/articles/Android/article.html

[4] Google Directions API

https://developers.google.com/maps/documentation/directions/

[5] Android Developer Guide

http://developer.android.com/index.html

[6] SMS Messaging in Android

http://mobiforge.com/

[7] Android SQLite and Content Providers

http://www.vogella.com/articles/AndroidSQLite/article.html