TRIP TRACKER APPLICATION ON ANDROID

A Thesis

Presented to the

Faculty of

San Diego State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

by

Siddartha Sreenivasa Reddy

Summer 2011

SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Siddartha Sreenivasa Reddy:

Trip Tracker Application on Android

Joseph Lewis

Joseph Lewis, Chair
Department of Computer Science

Kris Stewart

Kris Stewart
Department of Computer Science

[illegible]

Caroline A. Macera
Graduate School of Public Health

5/10/2011

Approval Date

DEDICATION

This work is dedicated to my Parents.

ABSTRACT OF THE THESIS

Trip Tracker Application on Android
by
Siddartha Sreenivasa Reddy
Master of Science in Computer Science
San Diego State University, 2011

Travel has always been a man's best pass time, a method to rejuvenate from the daily stress, a break from the monotonous life and to experience the thrill of adventure. Until the last decade, camera was a traveler's best friend but little did we know things are going to change a lot better. In today's world, life is always on the move. With the advancement of technology, smart phones today have immense capabilities to provide rich user experience with interactive facilities.

Trip Tracker is an Android based application for travelers to obtain the geo-location and tag it with multimedia features. This application allows users to create, store and view their trips, trip related information and all the memories that bring with it. Trip Tracker combines places visited, notes taken and the images captured, and display all this information on a map at the exact location where it all took place.

This application is developed to provide the users a rich user experience by having all the information in one place, easy-to-access and interactive. With the help of Google Maps, each trip can be drawn out on the map with all the locations visited and the route taken. The user will also be able to view the description, the location address and the image captured if any.

Trip Tracker, developed in Android, provides extensive flexibility, supports many features and can be among the best travel friendly app.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

1.	GPS	Global Positioning System
2.	NLP	Network Location Provider
3.	LBS	Location-based services
4.	GIS	Geographical Information System
5.	OHA	Open Handset Alliance
6.	VM	Virtual Machine
7.	SDK	Software Development Kit
8.	GNU	GNU’s Not Unix!
9.	API	Application Programming Interface
10.	USB	Universal Serial Bus
11.	URL	Uniform Resource Locator
12.	UI	User Interface
13.	API	Application Programming Interface
14.	GUI	Graphical User Interface
15.	SD	Secure Digital
16.	URI	Uniform Resource Identifier
17.	JPEG	Joint Photographic Experts Group

ACKNOWLEDGEMENTS

This research project would not have been possible without the support of many people. I wish to express my gratitude to my adviser, Prof Dr. Joseph Lewis who offered me invaluable assistance, guidance and support. My deepest gratitude to Prof Dr. Kris Stewart and Prof Dr. Caroline A. Macera without whose support and assistance this study would not have been successful.

Thanks to my parents for their moral support. Special thanks to my friend Neha Karnam for providing invaluable assistance and advice.

CHAPTER 1

INTRODUCTION

With changing times, the mobile technology has changed a lot and in the last few years we have seen the arrival of various new kinds of gadgets in the form of smartphone, camera-phone, Android and tablet phones. In fact, the handset industry has turned from simple budget handsets to ultra-modern high end mobile phones. Today’s device is almost everything - it is fashionable, innovative, appealing, high-performing, durable, stylish and multi-tasking. Latest gadgets can be used for various purposes like browsing mobile, internet, playing games, emailing, blogging, messaging and accessing all popular social networking sites like YouTube, Google search, Gmail and more.

Along with this, there has been a booming market for the multimedia mobile phones. Modern gadgets are coming with built-in cameras, which enable users to capture photos, print them and finally upload them on to Facebook, YouTube and other social networking sites. These devices pack music players, FM radio and come pre-loaded with amazing games. Thus, they are both entertainment as well as talking devices.

With the rise of mobile phone applications, so-called apps, people today are more looking for information on the go. This is one area of mobile phone technology enhancement that allows developers and programmers to offer users just what they seek under their preferred area of interest. Google’s Android is one of the latest and unique innovations, which instantly has taken over the mobile market. It is an open source mobile platform which allows developers from around the world to develop applications for Android supported mobile devices. Android supports to develop a location-aware application utilizing Global Positioning System (GPS) and Android’s Network Location Provider to acquire the user location. Although GPS is most accurate, it only works outdoors; it quickly consumes battery power, and doesn't return the location as quickly as users want. Android's Network Location Provider (NLP) determines user location using cell tower and Wi-Fi signals, providing location information in a way that works indoors and outdoors, responds faster, and uses less battery power. To obtain the user location in the application, both GPS and the Network Location Provider can be used or just one.

Ten billion apps have been downloaded in the past three years. There are 17,000 location-based travel apps on the market, and 160 million app-compatible devices are owned worldwide – iPhones, Androids, BlackBerrys and tablet devices such as the iPad and Motorola Xoom [1]. There are apps that can make our holidays a little easier, a bit more fun and more memorable. They let you do anything you can do online or with a guidebook, but more quickly and easily and while you're on the move – with maps and GPS to tell you where you are and capture wonderful memories.

This research is based on development of a user-friendly Android-based application called Trip Tracker. This is one such social travel mapping application designed to organize and store memories of the travels and adventures with a lot more information about them. Trip Tracker combines photos and notes of the travelers, and display all this information on a map at the exact location where it all took place.

This research is oriented in creating a travel logger integrated with GPS to track places to store the route along with capturing images using the build-in cameras in the mobile devices. This application is developed with intensive research on Location-based services and Map Overlays to provide the users to create a personalized travel journal and tag photos and other information to the places visited.

CHAPTER 2

LITERATURE

2.1 Mobile Platform

The mobile phone is one of the quickest to be adopted technologies in human history. As smart phones drop in price, we will see a rapid shifting how mobile phones are perceived: from simple communication devices to general purpose mobile computers. The Apple iPhone and Google Android have already begun to popularize this paradigm shift. Soon, even low end phones will be deployed with fast processors, long battery life, and rich sensing capabilities (such as GPS, accelerometer, infrared light, etc.). And, unlike their desktop counterparts, mobile phones have the unique property of always being on hand, a near-constant companion of their users. In this way, the mobile phone will become a sort of digital extension of the user, sensing context as the user moves about the world. However, tools and techniques that have long been refined for creating successful desktop computing environments do not translate well to the mobile environment. The focus of our proposal is on developing the next generation of mobile computing applications, which will incorporate a near-constant internet connection, novel interactions (e.g., multi-touch), sensors, and machine learning (e.g., activity inference) to provide rich, interactive experiences [2].

Global Positioning System, popular and commonly known as GPS, is a kind of space-based navigation technology providing us with the exact location and time details no matter where we are. It’s not affected by bad weather or any other obstruction in the navigation process.

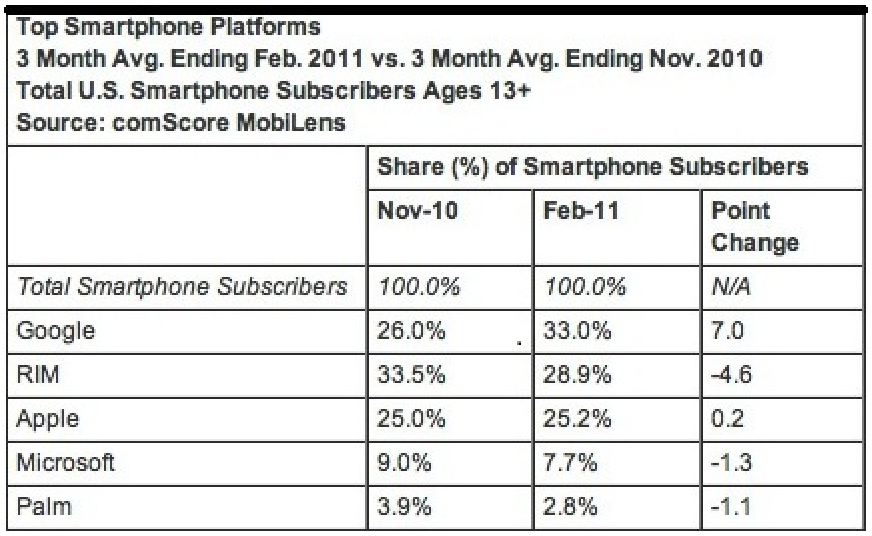
The technology was brought into the practice at the first time back in 1973 by U.S. Defense Department. From the testing experience, the technology showed it’s much superior and advanced, and it has superseded all the earlier versions of navigation systems. With the rapid development of navigation technology, GPS is widely used in many fields such as army, banking and surveillance, science, technology, commerce, geology, telecommunication, and etc. [3].

In theory, the GPS technology works in this model. A GPS Data Logger that tracks the exact location of a vehicle, or a device, and then the tracking information would be acquired through the connection of the GPS data Logger with a computer via local or global Internet [3].

Thanks to the development of GPS technology, the most advanced navigation technology-real time GPS service becomes true. It’s been fast and wildly adopted in many fields and is becoming to play a crucial role in our modern life, especially in security and safety field [3].

2.2 RESEARCH AND STUDY

Over the past year, Google’s Android and Apple’s iOS have been strong competitors in the Mobile app market. According to comScore, iOS mobile devices captured 25% of the market in February 2011. That's up only slightly from November 2010, despite the introduction of the iPhone on Verizon's network. On the other hand, iOS' biggest competitor (in the eyes of many), Google's Android, has grown 7% since November 2010, and now commands 33% of the smart phone subscriber market in the United States [4]. Figure 2.1 shows the total U.S. smartphone subscribers [5].



Top Smartphone Platforms
3 Month Avg. Ending Feb. 2011 vs. 3 Month Avg. Ending Nov. 2010
Total U.S. Smartphone Subscribers Ages 13+
Source: comScore MobiLens

	Share (%) of Smartphone Subscribers		
	Nov-10	Feb-11	Point Change
Total Smartphone Subscribers	*100.0%*	*100.0%*	*N/A*
Google	26.0%	33.0%	7.0
RIM	33.5%	28.9%	-4.6
Apple	25.0%	25.2%	0.2
Microsoft	9.0%	7.7%	-1.3
Palm	3.9%	2.8%	-1.1

Figure 2.1. Top smartphone platforms. Source: comScore, Inc. comScore Reports February 2011 U.S. Mobile Subscriber Market Share, 2011. http://www.comscore.com/Press_Events/Press_Releases/2011/4/comScore_Reports_February_2011_U.S._Mobile_Subscriber_Market_Share, accessed Apr. 2011.

By providing an open development platform, Android offers developers the ability to build extremely rich and innovative applications. Developers are free to take advantage of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, and much, much more. Figure 2.2 iOS versus Android [6].

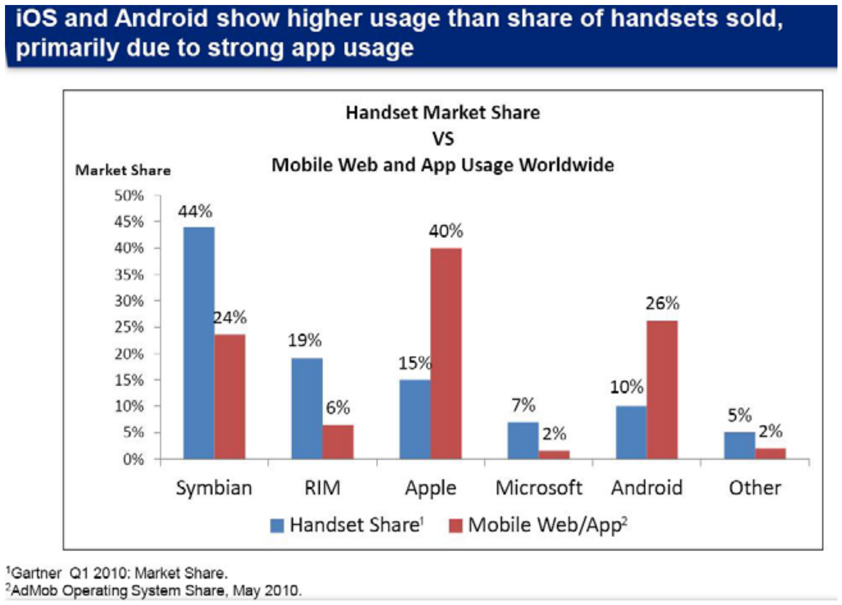


Figure 2.2. IOS versus Android. Source: AdMob Google Inc. AdMob Mobile Metrics, 2010. http://metrics.admob.com/#handset share or mobile app, accessed Aug. 2010.

Location-based services (LBS) [7] are one of the most popular services based on a different navigation technologies provided by the mobile communication network. Location-based applications gained a lot of momentum in 2010, nearly tripling the number of users from 2009 (from 12.3 million in 2009 to 33.2 million in 2010). This growth is expected to continue with the prediction that mobile location-based ad spending will increase from $42.8 million in 2010 to $1.8 billion by 2015 [8].

LBS receives the location coordinates from the ground based mobile station and sends it to the mobile phone user and the communication center that can be used for various location based services. So in nutshell, LBS is a wireless service that determines the location information of the information device (mobile) user by using Geographical Information System (GIS) and other satellite navigation platform.

With the proliferation of mobile email and mobile browsing, the growth of multimedia messaging, mobile music, social networking, and even location-based services, mobile is indeed the next wave for travel-related services. Apps for mobile-enabled services are increasingly rooted in every day's life and travel habits.

2.3 Challenges in Mobile Application Development

Mobile phones are not just "phones" anymore with only "voice and SMS" functionalities. The continuing spread of mobile technology will have a dramatic impact on the lives of individuals and institutions. Convergence of internet and telecommunication technologies is increasing rapidly. However, in the progress of mobile application development, due to complex structure of mobile ecosystem, there is a fragmentation in terms of different mobile "Operating Systems", "Screen Resolutions", "Device Models and Capabilities, " and "User Experience". Fragmentation is the word that defines the biggest barrier. Fragmentation increases the cost and the time to develop mobile applications. Figure 2.3 shows the fragmentation/cost of application development [9].

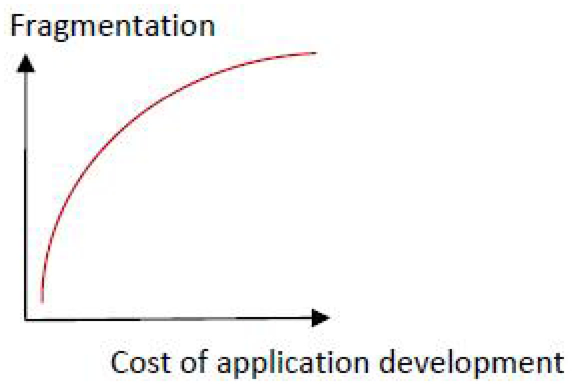


Figure 2.3. Fragmentation/cost of application development. Adapted from source. Source: Israfil Coskun and Osman Celik. Challenges of Mobile Application Development, 2010. http://developer.smartface.biz/challenges-of-mobile-application-developement-n74.html, accessed Dec. 2010.

Different device models support different functionalities, such as touch screen, gravity censors, camera flash, etc. Hardware performances also vary between devices. In addition to that, some applications need to support external device functionalities [9].

Screen resolution is crucially important in application development. Ongoing trend is to have bigger screen resolutions for expanded multimedia support, data presentation, and browsing. However, device manufacturers tend to support multi-range of resolutions to address the needs of different user segments [9].

User experience, which is simply the way the users behave while using the keyboard, screen, entry-exit functions, number of clicks, etc., is different across device models. Usage scenarios and actions are different between device models and these differences should be realized and taken into consideration in order to deploy successful mobile applications [9].

CHAPTER 3

DESIGN SPECIFICATION

3.1 Overview

Trip Tracker is an application built using Android framework for mobile platform. The application is used to create and maintain trip journals with all the places visited with a short description and/or an image captured.

3.2 Android Platform

Android is an open source operating system for mobile devices. Android was initially developed by Android Inc., and sold to Google in 2005 [10]. On November 2007, the Open Handset Alliance (OHA) was announced amongst a consortium of several top companies [10]. The goal was to develop an open mobile platform every developer to contribute towards improving the performance and features of the product.

Android is built on top of Linux kernel and GNU software. Software stack of the Android runs Java applications using Java core libraries. Each instance of Java application runs on its own Virtual Machine (VM) called Dalvik [11].

Well as for Dalvik VM, it has its own Java Byte code and is designed to be to be optimal on memory and processor usage. The VM executes Dalvik Executables with (.dex) extension. Among the various tools built in the Android, ‘dx’ tool is used to generate the executable that converts the Java classes into .dex format [11].

Android relies on the Linux kernel to perform system level functions such as memory management and threading & even the more dependent on it for hardware interactions and power management.

Developers can build applications using the software development kit (SDK) developed by Google. It consists of Application Programming Interface (API) used to develop robust Java applications [10]. These API’s facilitate to access the contents on the phone such as contacts and calendar information and also integrate them with external web-service in order to provide online services.

3.3 COMPONENTS OVERVIEW

Trip Tracker is an Android application to create a mini travel journal. This app allows you to add extra data to your pictures, to associate them with the places where they were taken, the possibility to add GPS coordinates to photos, and organize them.

Figure 3.1 shows the flow of the Activity to create a new trip.

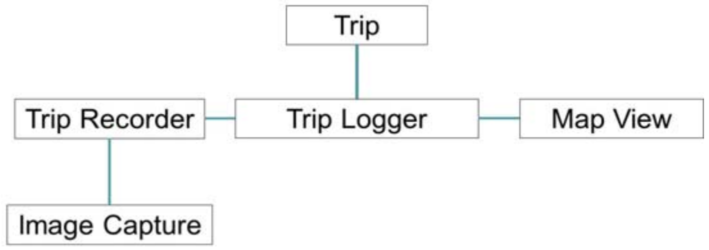


Figure 3.1. Application's overview design.

To create a new trip, the user selects "New Trip", this loads to a TabActivity displaying three tabs: Trip recorder, Trip logger and Map view. Each tab leads to a different activity.

Trip Recorder triggers the GPS activity and captures the current location. This also provides the user to specify the location name, capture a picture and write a short description of that place.

Trip Logger shows the list of the places visited with their respective addresses, also displaying the short description. If a particular location is selected, it loads into the editing activity enabling the user to modify any of the parameters.

Map View displays the location points on a regular map for the user to easily locate them with route overlays. Once clicked on the location point, it displays the details of the location like the location address, picture captured and the description written by the traveller.

The detailed process will be explained in the later chapters.

3.4 HIGH-LEVEL COMPONENTS

High-level components are the very essential in any project to get a clear understanding of the purpose and the various high-level modules involved. The design of high-level components form the backbone of the project upon which the detail components stem out. This section illustrates the high level components of this application explaining the design and the importance of each component.

3.4.1 Overview

An Activity is a single focused task that the user can do at any given point in time. All the activities interact with the user. The activities are responsible for creating the User Interface with setContentView(View) method. They play a vital role in application's overall lifecycle. Each activity is managed in an activity stack. When a new activity is created, it is placed on top of the current activity and becomes the running activity. The previous activities are resumed after the current activity finishes its execution. Figure 3.2 shows the application's high level overview design.

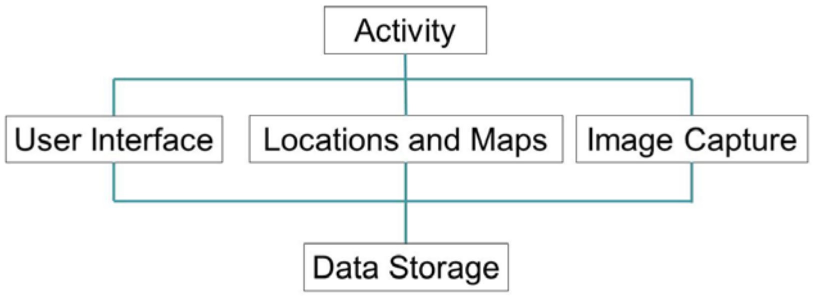


Figure 3.2. Application's high-level overview design.

Intents are used to send asynchronous messages between the activities in the application. Objects of type "android.content.Intent" communicate between the activities to send/receive data using the method getIntent() method. All the activities and intents are specified in the AndroidManifest.xml file [12]. This is a must-needed file for every application and is present in the root directory. This file presents essential information of the application to the Android system, the information which the system must have before it runs the application code. It describes the components of the application, the classes that implement the components and other related information. This lets the Android system know what the components are and under what circumstances they can be launched.

3.4.2 User Interface

The User Interface is build using Views and ViewGroup objects. Android UI is responsible for the Layout manager and the widget organizer. View objects are the basic units of user interface expression on the Android platform. The View class serves as the base for subclasses called "widgets, " which offer fully implemented User Interface (UI) objects, like text fields and buttons. The ViewGroup class serves as the base for subclasses called "layouts, " which offer different kinds of layout architecture, like linear, tabular and relative. A View object is a data structure whose properties store the layout parameters and content for a specific rectangular area of the screen. A View object handles its own measurement, layout, drawing, focus change, scrolling, and key/gesture interactions for the rectangular area of the screen in which it resides. As an object in the user interface, a View is also a point of interaction for the user and the receiver of the interaction events.

The Android UI toolkit offers several layout managers that are easy to use to implement an user interface. This application employs Linear Layout, Frame Layout and Relative Layout. Linear Layout wraps the TextView, EditText, ListView and Buttons; MapView exploits Frame Layout and Relative Layout is used in Trip Logger Activity for displaying the various locations in the trip.

The Android UI framework is organized around a Model-View-Controller pattern. It provides structure and tools for building a Controller that handles user input (like key presses and screen taps) and a View that renders graphical information to the screen.

The Model is the soul of the application, what it actually does. The View is the application’s feedback to the user. The graphical portion of the Android UI framework's View is implemented as a tree of subclasses of the View class. Graphically, each of these objects represents a rectangular area on the screen that is completely within the rectangular area represented by its parent in the tree. The Controller is the portion of an application that responds to external actions: a keystroke, a screen tap, an incoming call, etc. It is implemented as an event queue. Each external action is represented as a unique event in the queue. The framework removes each event from the queue in order and dispatches it.
Figure 3.3 shows MVC concept model [13].

When an external action occurs (for example, when the user scrolls, drags, or presses a button; a call comes in; or an MP3 player arrives at the end of its playlist), the Android system enqueues an event representing the action on the event queue. Eventually the event is dequeued—first in, first out—and dispatched to an appropriate event handler. The handler

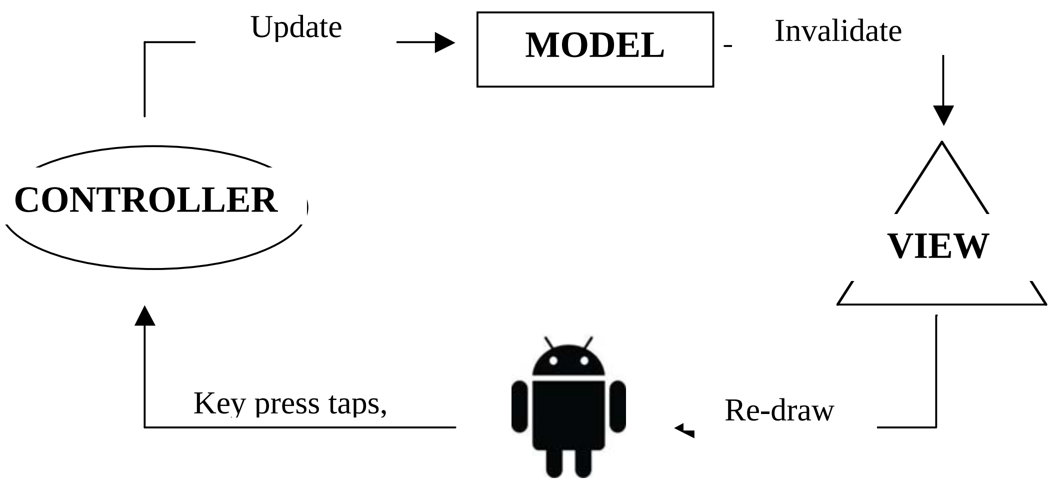


Figure 3.3. Model-view-controller concept. Source: Rick Rogers and John Lombardo. Android Application Development, 2009. http://androidapps.org.ua/i_sect13_d1e11121.html#Android GUI Architecture, accessed Aug. 2010.

responds to the event by notifying the Model that there has been a change in state. The Model takes the appropriate action.

3.4.3 Location and Maps

Android application can access the location services supported by the device through the classes in the android.location package and the Google Maps external library. The main component of the location framework is the LocationManager [14] system service, which provides APIs to determine location and bearing of the underlying device. Google provides a Maps external library that includes the com.google.android.maps package. The com.google.android.maps package used in this application offered built-in downloading, rendering, and caching of Maps tiles, as well as a variety of display options and controls.

The key class in the Maps package is com.google.android.maps.MapView, a subclass of ViewGroup. A MapView displays a map with data obtained from the Google Maps service. When the MapView has focus, it will capture key presses and touch gestures to pan and zoom the map automatically; including handling network requests for additional maps tiles. It also provides all of the UI elements necessary for users to control the map. A LocationListener [14] is the interface implemented to receive location updates. To use Google Maps in the application, a Maps API key had to be obtained to register with the service and Android system had to be notified that the application wishes to implement the add-on Google APIs which are external to the base APIs. This was done by using the uses-library element in the Android manifest file, informing Android that the application used classes from the com.google.android.maps package.

This application also uses Map Overlays to portray to the users the various locations visited and a short description of each. This is done by creating Map markers and lay-overs. This has been possible by using ItemizedOverlay class to manage all the individual items placed on the map.

MapActivity is the spacing activity defined to show Google Maps. Map View is the view that supports and displays the map. This is contained in the Map Activity.

Utilizing Location and Maps pose a huge design consideration for this application, since it uses high GPS activity and more power consumption.

3.4.4 Image Capture

Multimedia is another feature of this application. This integration provides a higher-end GUI application to the users. The mobile device camera is utilized to capture pictures and tag them to the location points. To access the device camera, permission had to be set in the Android Manifest file by including the <uses-permission> as CAMERA and <uses-features> manifest element to declare the camera features used by the application.

The Camera class is used to set image capture settings, start/stop preview and snap pictures. This class manages the actual camera hardware and is the client for the Camera service. Camera class is not thread safe and is meant for use from only one event. Thus this class's methods could not be called from multiple threads at once.

MediaStore [15] is the class which contains the meta data for all available media on both internal and external storage devices. ACTION_IMAGE_CAPTURE [15] is the standard intent which is used by the camera application to capture an image and return it. The EXTRA_OUTPUT [15] parameter is set to store the images captured in the gallery and name them with the place name previously named by the user. The Application stores the images in the SD card of the device and can also be viewed in the Image Gallery of the phone.

3.4.5 Data Storage

The data storage options which the Android supports: Shared Preferences, Internal Storage, External Storage, SQLite Databases and Network Connection. Trip tracker uses Internal storage, External Storage and SQLite databases for storing persistent application data.

By default, the application once installed is stored in the internal storage of the Android system. This is private to the application and other applications cannot access it. When the user uninstalls the application, these files get removed.

Since every Android-compatible device supports a shared “external storage” to save files, this application uses Secure Digital (SD) card to store the images captured. External storage are open to the public and can be read by all who can access it. It can be modified by the user when connected to a computer and the mode is set to USB Mass Storage which allows transferring files.

Android also provides full support to SQLite databases. All databases that are created in the application are accessible by name to any class in the application but none outside. This is implemented by creating a sub-class to SQLiteOpenHelper [16] and overriding the onCreate() method to execute the SQLite command to create the tables in the databases. The methods getWritableDatabase() [16] and getReadableDatabase() [16] are called for write to and read from the database which return SQLiteDatabase [16] object.

The Android SDK includes a sqlite3 database tool which is required to browse the table contents, run SQL commands and perform other SQL functions. Executing an SQL query returns a Cursor which stores the result set pointing to all the rows found by the query.

In this application, relational database is used to create two tables for storing trip names, trip details: location name, location description, image URI and geo points of the location.

CHAPTER 4

MODULES

4.1 OVERVIEW

The Android application, Trip Tracker is developed using many user-defined Activities which are based on many user-defined .java classes. Figure 4.1 illustrates the control flow among the classes and will be explained in detail further down.

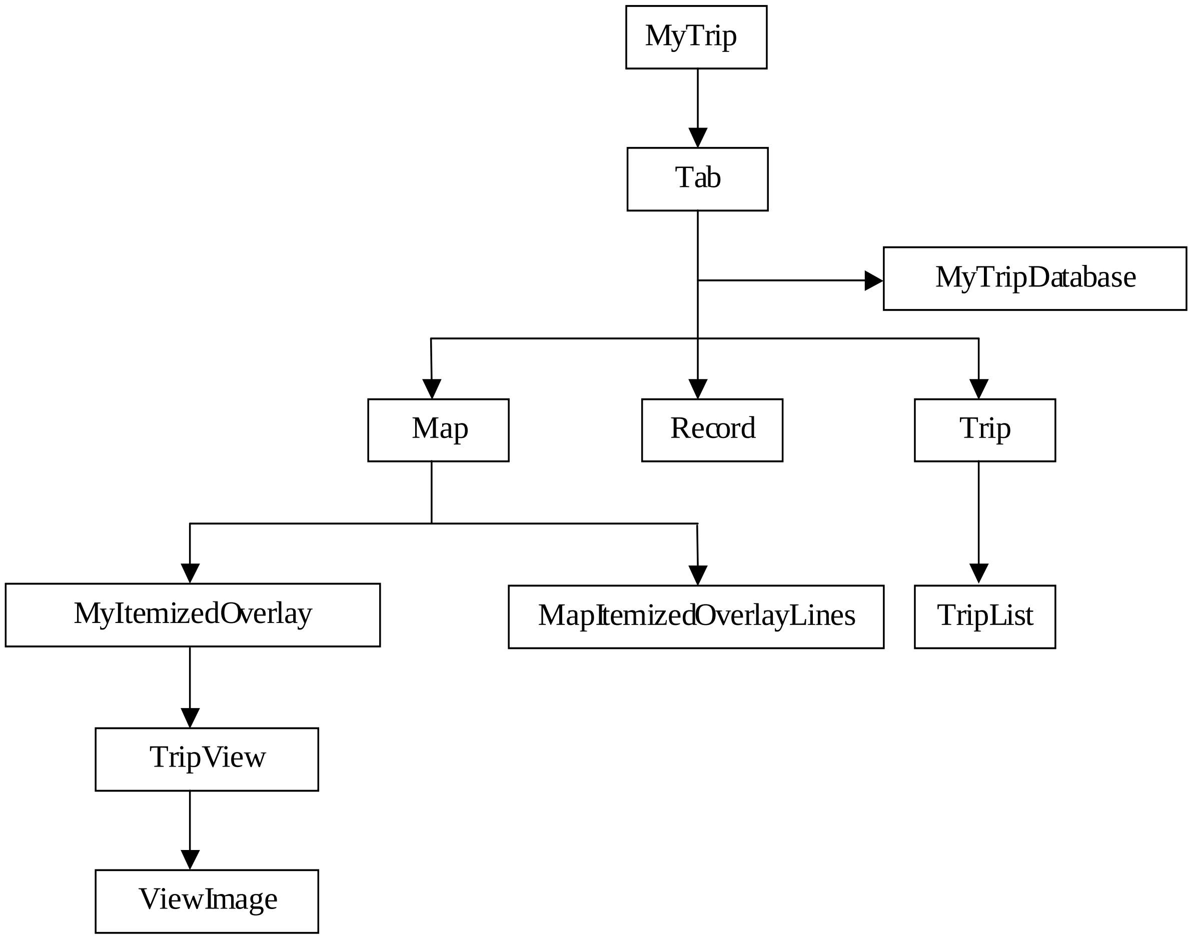


Figure 4.1. Displaying the workflow diagram with all the class files.

These class files form the backbone of the entire application. Each class file has a specific purpose in creating/calling the Activities of the application.

4.2 Implementation Details

There are seven Activities which are a part of the lifecycle of this application: MyTrip, Tab, Map, Trip, TripView, Record and ViewImage.

MyTrip is the initial Activity which gets loaded when the application starts. This is designed to display all the trips the user has been to, using a ListView Layout and also providing an option to create a new trip. Figure 4.2 shows the new trip creator view.

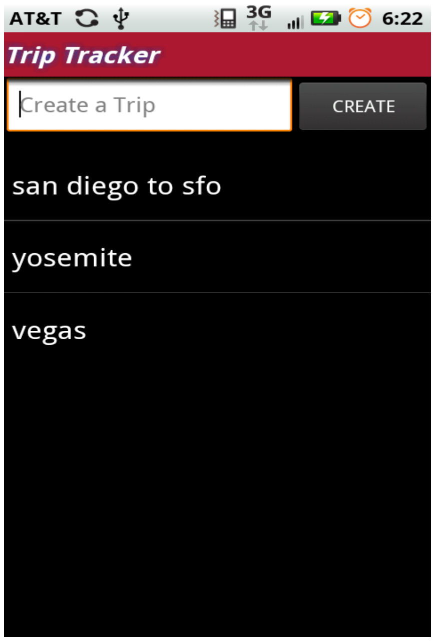


Figure 4.2. New trip creator.

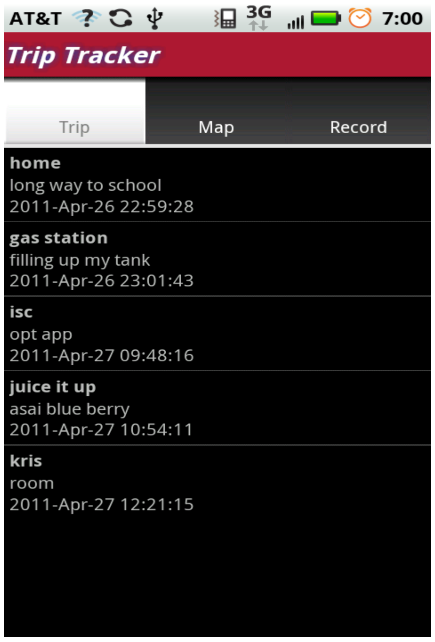
MyTrip implements two listeners:

1. setOnItemClickListener(): This listener is activated when the user clicks on any of the trip from the list and then calls the method onItemClick(). This method redirects to intent to call the Tab class showing the details of that trip.
2. setOnClickListener(): This listener is activated when the user clicks on the “New Trip”. It opens up an alert box to create a new trip. This also opens a new entry for the new trip in the database using the method addTripName().

There are also two default methods that are always implemented: getDataList() and showDataList(). getDataList() retrieves already existing trips and calls the showDataList() to display it during the launch of the Activity. getDataList() also manages a Cursor to handle the records of the stored database. If the user clicks on the New Trip button, the application calls the Tab Activity showing three tabs: Trip Logger, Map and Recorder. These three tabs define each trip.

4.2.1 Trip Logger

Trip Logger is the host tab and is displayed as the default tab which gets loaded once the New Trip button is pressed. This is used to list all the locations visited in the trip. Each location displays the location name, a brief description and date. This tab also offers the opportunity to edit the location parameters such as location name and location description, once clicked on the location listed in the Trip Logger. Figure 4.3 shows the trip logger view.



home
long way to school
2011-Apr-26 22:59:28

gas station
filling up my tank
2011-Apr-26 23:01:43

isc
opt app
2011-Apr-27 09:48:16

juice it up
asai blue berry
2011-Apr-27 10:54:11

kris
room
2011-Apr-27 12:21:15

Figure 4.3. Trip logger tab.

A unique identifier (ID) is assigned to each trip. When a new trip is selected, an ID assigned to the trip is passed on to the Tab Activity and this ID holds active throughout the active state of the trip. The methods getDataList() and showDataList() are used to retrieve the places and place info from the database storage and display it in the list view.

4.2.2 Map View

Map View is opened through the click of another Tab. Map View uses the location services supported by the device through the classes in android.location package [14]. The Maps package com.google.android.maps.MapView [14], a subclass of ViewGroup is used to display a map with data obtained from the Google Maps service. Figure 4.4 shows the map view.

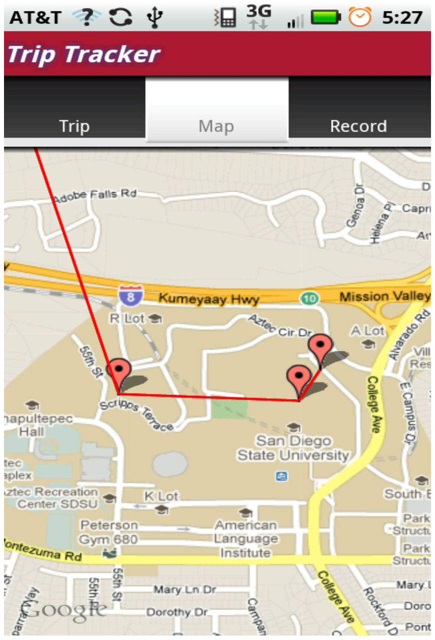


Figure 4.4. Map view tab.

To use the Google maps data, Maps service had to be registered and MD5 fingerprint of the certificate which is required to sign the application had to be provided to obtain a Maps API key. This key had to be inserted in the .xml file of the Map View for the value of android:apiKey to enable the Google maps.

To use the Maps external library, <uses-library> element in the Android Manifest file is defined to “com.google.android.maps”. This thus enables the build tools to link the application against the Maps external library. This is to also ensure that the Android system checks the required library is available to install the application on the device.

To use the Maps in the application, another step is required. MapActivity class had to be extended and a layout had to be created that included a MapView element. MapActivity is a special sub class of Activity which has important map capabilities.

Map View properties are set. To enable zoom controls in the map, setBuiltinZoomControls() is set to true. getZoom() and initial Zoom, max Zoom().

MapView class provides a wrapper around the Google Maps API to manipulate the Google Maps data through class methods.

To create map markers and lay-overs, ItemizedOverlay class had to be implemented. All the locations that need to be tagged are added to OverlayItem ArrayList. The populate() method is called each time a new OverlayItem is added to the list which reads each of the objects in the list and prepare them to be drawn.

To handle the touch events, onTap() method is called. This method obtains the latitude and longitude of the point. This method uses the member android.content.Context to create a new AlertDialog.Builder and uses the tapped OverlayItem's title and snippet for the dialog's title and message text. So this application opens an alert box for the corresponding location name showing brief description of the place: place name and place description. On clicking on the Alertbox to show more details, it leads to a different Activity called TripView displaying the complete information of the place: Image captured at the location, complete description, location address and the location name.

4.2.3 Recorder

Recorder is another important tab for the creating the trip. The trip starts from the recorder. Upon selecting the Recorder tab, it loads into a new Activity called the “Record.” Figure 4.5 shows the recorder view.

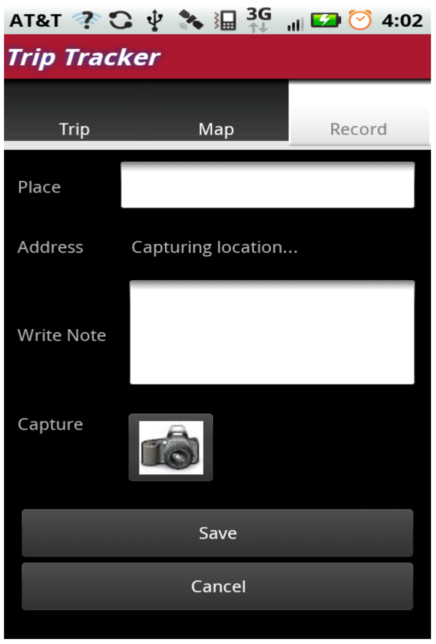


Figure 4.5. Recorder tab.

This activity concentrates on recording the location name, location details, photo and geopoints of the trip location.

To capture the geopoint of the current location, LocationListener [17] interface is implemented. This interface uses the onLocationChanged() [17] method, which gets the Latitude and Longitude of the current location.

Another method requestLocationUpdates() [17] registers the current activity to be notified periodically based upon the specific conditions which are specified as the four parameters of this method:

- minTime - minimum time interval for notifications i.e. the frequency of notification or new locations can be controlled thus allowing the LocationManager to potentially rest for minTime milliseconds between location updates to conserve power.
- minDistance - Location will only broadcast, if the device moves by minDistance.
- Criteria - contain parameters for location manager to choose provider and parameters to compute the location.
- Intent - intent to be sent for each location update.

After registering the current location, the latitude and longitude is used to obtain the location address using reverse geocoding [7] technique. This is possible on extending the Geocoder class and implementing the method getFromLocation() [17]. The latitude and longitude are passed as parameters to this method to obtain a partial address of the location.

The camera capture feature of this activity works by extending the MediaStore [15] class. This class implements intent called ACTION_IMAGE_CAPTURE [15] which is sent to the camera application to capture an image and return it. This intent opens up the camera preview. The preview has features to capture image and once the desired image is captured, it is stored in external storage space in the JPEG format. The URI of that image is then inserted into the database.

After obtaining the location name, location description, image and the geo points of the location, they are updated to the database as a new record.

CHAPTER 5

USER'S MANUAL

5.1 OVERVIEW

This chapter educates the users and developers of the application, by providing information on getting started and installation of the application on their mobile devices.

5.2 SETTING UP THE DEVELOPMENT ENVIRONMENT

A typical developer needs to set up an environment for android development. The installation steps given below assume that the developer is aware of setting up the development environment for the android development which is given in android developer site. This application can be developed on any platform which runs Java/android. The installation steps for the development:

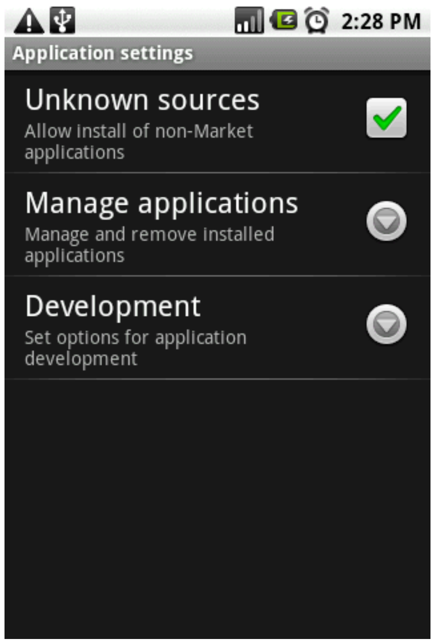
- Install Java JDK and JRE
- Install Eclipse IDE
- Install Android SDK
- Install ADT plug-in in Eclipse for Android Development

Once the setup of the machine is done, the next step is to download the source code and make changes on to it:

- Copy the zip files of the source code to your machine.
- Unzip the file to your workspace.
- Import the project into the Eclipse

5.3 INSTALLING APPLICATION ON DEVICE

The first step is to get the .apk files from the Unknown Source or the other way is to download from the Android Market. Installation from Android Market is direct and simple. If it has to be downloaded from other source (except Android Market) then the phone settings should be changed. Figure 5.1 shows the installation from unknown source.



2:28 PM

Application settings

Unknown sources ✓
Allow install of non-Market applications

Manage applications
Manage and remove installed applications

Development
Set options for application development

Figure 5.1. Installation from unknown source.

For the manual installation, the following instructions should be followed:

- Check if you are in HOME screen
- Click on Menu button
- Select Settings
- Click on applications
- Check the option for Unknown Sources

Now the application will be installed on to the device.

5.4 Using the Application on Device

A first time user needs to select the “New Trip” button to kick off the application and enter the first record in the database. Initially, the database in empty and shows no trips.

5.5 Creating a New Trip

Creating a New Trip can be done in two ways:

1. Tap on the Edit box provided next to the CREATE button. Enter the name of the trip and click on Create (see Figure 5.2).
2. Click on the CREATE button, an alert box pops up. Enter the Trip Name in the pop up edit box and save the name (see Figure 5.3).

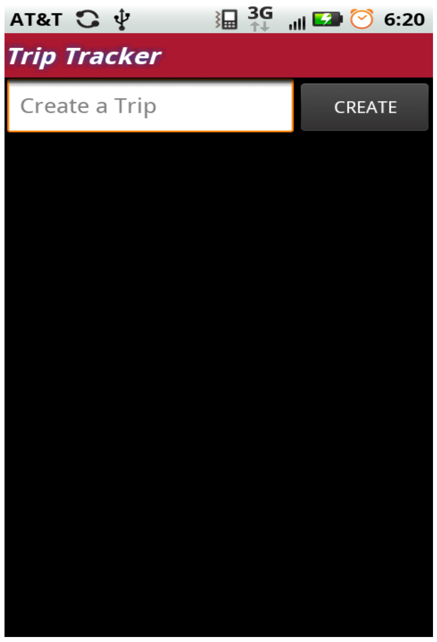


Figure 5.2. Step 1 to create a New Trip (a).

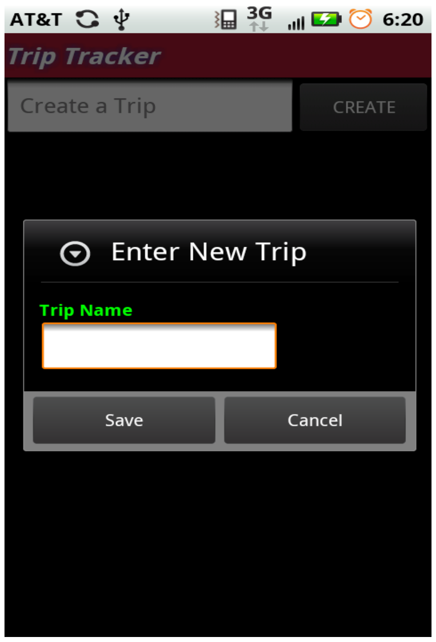
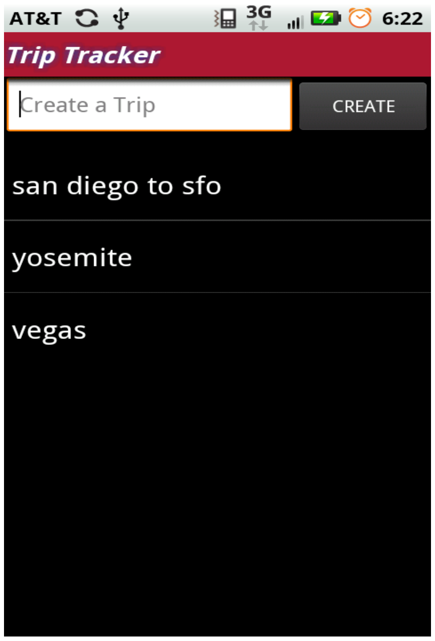


Figure 5.3. Step 1 to create a New Trip (b).

Once the Trip Name is saved, it can be seen in the list view (see Figure 5.4).. This list view displays all the trips you have created.

To add the contents to the Trip, click on the Trip Name. The new view with three tabs can be seen: the Trip, Map and Record. These tabs will represent the trip. To start recording the trip, click on the Record tab to start the GPS activity to capture your current location. Enter the location name, the address will be captured automatically as the GPS is ON and you can write a short note to describe the place. Click on the camera button to capture an image of the place and save it. The place details will be saved onto the database (see Figure 5.5). If you change your mind and need to move to a different location or you are not interested in the current location, you can simply press Cancel and the Record tab closes and you will be brought back to the Trip View.

Once the Save button is clicked, user will be taken to the Trip view. Here is the list of places that have been visited. Each record of the place is shown with the place name, brief description and the date and time of the place recorded. This is also placed in the order of the places visited to give you an idea of the route taken (see Figure 5.6).



san diego to sfo

yosemite

vegas

Figure 5.4. Step 2 on displaying the trips in list view.

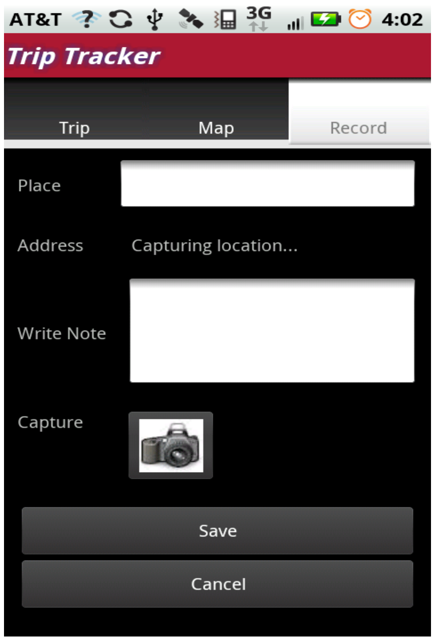
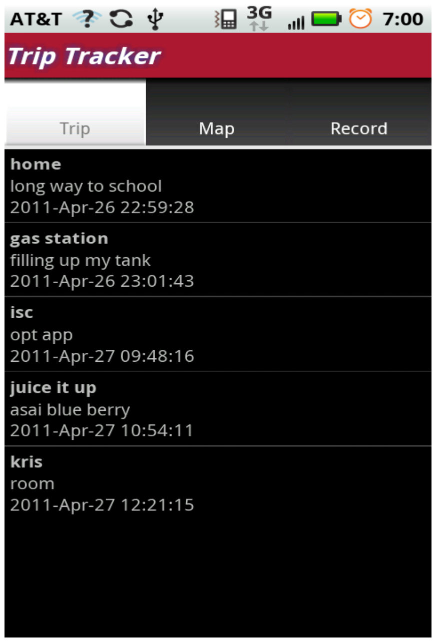


Figure 5.5. Step 3 on clicking the record tab.



home
long way to school
2011-Apr-26 22:59:28

gas station
filling up my tank
2011-Apr-26 23:01:43

isc
opt app
2011-Apr-27 09:48:16

juice it up
asai blue berry
2011-Apr-27 10:54:11

kris
room
2011-Apr-27 12:21:15

Figure 5.6. Step 4 the locations visited.

To make it more interactive and fun, select the Map tab, you can see all the places recorded on the live map (see Figure 5.7). The places are shown with the markers on them. The Markers are connected with lines as route. These lines are connected in the order of recording the places .

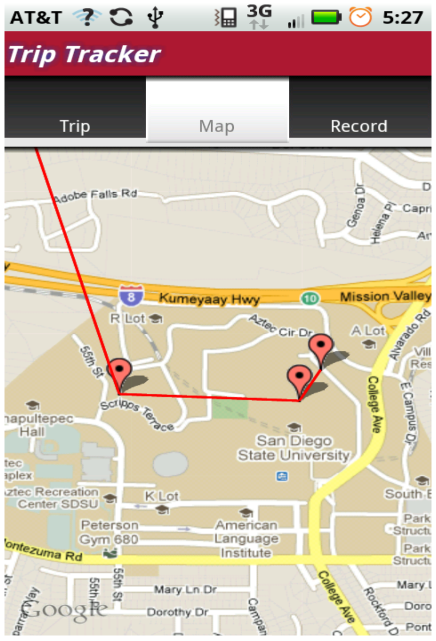
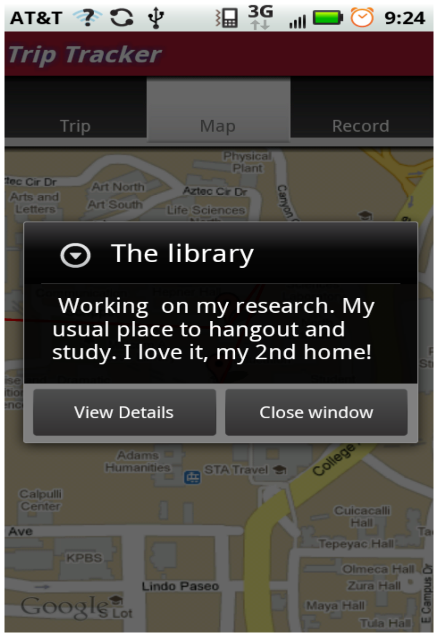


Figure 5.7. Step 5 on clicking the map tab.

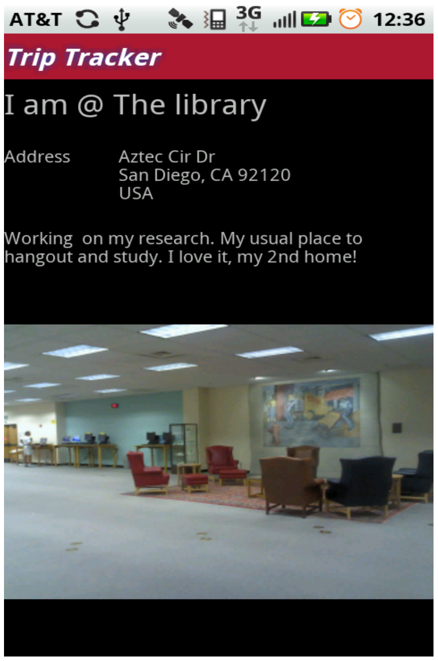
On tapping the markers, the place detail will be popped up with name of the place showing the brief description (see Figure 5.8). If you wish to see the complete details, click on the View Details button which brings you to another activity to display the name of the place, address of the place, full description and image corresponding to the place (see Figure 5.9). There is also a choice to edit the details of the location. To do so, go to the Trip tab, click on the place to be edited. This opens up the new editing view, where the place name and the description can be edited (see Figure 5.10).



The library

Working on my research. My usual place to hangout and study. I love it, my 2nd home!

Figure 5.8. Step 6 on clicking a marker.

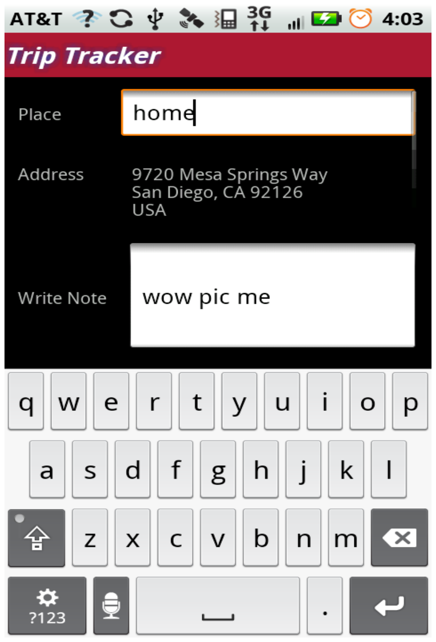


I am @ The library

Address Aztec Cir Dr
San Diego, CA 92120
USA

Working on my research. My usual place to hangout and study. I love it, my 2nd home!

Figure 5.9. Step 7 to view all the details.



Place: home

Address: 9720 Mesa Springs Way
San Diego, CA 92126
USA

Write Note: wow pic me

Figure 5.10. Step 8 to edit the details of the location.

CHAPTER 6

DISCUSSION

6.1 FUTURE ENHANCEMENTS

This thesis project is developed as a beginner application for trip logging. It can be further developed to provide a richer user experience. Trips saved can be shared to all friends through email. This would help to share the trips and trip details with the images to your friends.

The real time updates can be provided to your friends as and when a location is recorded in the trip. This would help to provide dynamic information as you travel to all your friends and family to keep you tracked.

Also, provision to upload more pictures can be provided to capture more memories and share them. There is also no option to delete or change the image captured before. This should also be implemented.

Multimedia features can be enhanced to upload a video. Since only images are supported now, video can add to a richer user experience.

Efforts to provide live navigation also can be included to re-visit the already listed trip.

6.2 CONCLUSION

The motivation of this thesis project is to build an Android based application for providing a simple trip journal for the users and to capture their memories of the trip. By using Android SDK and Eclipse IDE as the development environment the application is built keeping in mind about the design standards and maintainability of the code.

Trip Tracker makes use of GPS and multimedia features of the Android device to provide a rich user experience to the travelers enabling them to store all the locations visited along with the beautiful memories associated with them. This application is very simple to use and is helpful to record multiple trips displaying their routes and other details.

Trip Tracker has been tested on Motorola Bravo and Motorola Flip-Out having Android 2.2.

BIBLIOGRAPHY

[1] Francisca Kellett. The Best Travel Apps, 2011. http://www.telegraph.co.uk/travel/travelaccessories/8330846/The-best-travel-apps.html, accessed Mar. 2011.

[2] Google EMEA's AndroidEDU Programme. Sample Proposals, 2011. https://sites.google.com/site/androideduemea/sample-proposals#Context-Aware, accessed Jan. 2011.

[3] Lion. GPS Secure Your Life, 2011. http://www.myoversea.com/gps-secure-your-life, accessed Apr. 2011.

[4] Patricio Robles. Android versus iOS: The Cold Hard Truth, 2011. http://econsultancy.com/us/blog/7374-android-versus-ios-the-cold-hard-truth, accessed Apr. 2011.

[5] comScore, Inc. comScore Reports February 2011 U.S. Mobile Subscriber Market Share, 2011. http://www.comscore.com/Press_Events/Press_Releases/2011/4/comScore_Reports_February_2011_U.S._Mobile_Subscriber_Market_Share, accessed Apr. 2011.

[6] AdMob Google Inc. AdMob Mobile Metrics, 2010. http://metrics.admob.com/#handset share or mobile app, accessed Aug. 2010.

[7] Sandeep Kumar, Mohammed Abdul Qadeer, and Archana Gupta. IEEE location based services using Android. *Proceedings of the 2009 IEEE International Conference on Internet Multimedia Services Architecture and Applications (IMSAA)*, Bangalore, India, 2009.

[8] Carissa Richardson. The Rise of Location-Based Applications, 2011. http://blog.fahlgrenmortine.com/2011/02/the-rise-of-location-based-applications, accessed Apr. 2011.

[9] Israfil Coskun and Osman Celik. Challenges of Mobile Application Development, 2010. http://developer.smartface.biz/challenges-of-mobile-application-developement-n74.html, accessed Dec. 2010.

[10] Wikipedia.org. Android Operating System, 2011. http://en.wikipedia.org/wiki/Android_(operating_system), accessed Mar. 2010.

[11] Android Developers. What is Android, 2011. http://developer.android.com/guide/basics/what-is-android.html, accessed May 2010.

[12] Android Developers. The AndroidManifest.xml File, 2010. http://developer.android.com/guide/topics/manifest/manifest-intro.html, accessed Jun. 2010.

[13] Rick Rogers and John Lombardo. Android Application Development, 2009. http://androidapps.org.ua/i_sect13_d1e11121.html#Android GUI Architecture, accessed Aug. 2010.

[14] Android Developers. Android Location Package, 2010. http://developer.android.com/reference/android/location/package-summary.html, accessed Aug. 2010.

[15] Android Developers. MediaStore, 2010. http://developer.android.com/reference/android/provider/MediaStore.html, accessed Aug 12, 2010.

[16] Android Developers. SQLiteOpenHelper, 2010. http://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html, accessed Oct 24, 2010.

[17] Android Developers. LocationListener, 2010. http://developer.android.com/reference/android/location/LocationListener.html, accessed Aug 12, 2010.