Islamic University of Technology

"NearHand" An Android Based Application

By:

Md. Sazzad Hossain (094421)

Sanzidul Haque Seefat (094413)

Supervised by:

Mahmud Hasan

Assistant Professor

Department of Computer Science and Engineering

A project submitted in partial fulfillment of requirements for the degree of Bachelor of Science in Computer Science and Engineering

Academic Year: 2012-2013

Department of Computer Science and Engineering

Islamic University of Technology.

A Subsidiary Organ of the Organization of Islamic Cooperation

Dhaka, Bangladesh.

October 25, 2013

Declaration of Authorship

We, Md. Sazzad Hossain & Sanzidul Haque Seefat, declare that this project titled,"NearHand: An Android based Application" and the work presented in it are our own. We confirm that:

- This work was done wholly while in candidature for a Bachelor degree at this University.
- Where any part of this project has not been submitted previously for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.

Submitted By:

Md. Sazzad Hossain (094421)

Sanzidul Haque Seefat (094413)

NearHand: An Android Based application

Approved By:

Prof. Dr. M.A. Mottalib

Head of the Department,

Department of Computer Science and Engineering,

Islamic University of Technology

Mahmud Hasan

Project Supervisor,

Assistant Professor,

Department of Computer Science and Engineering,

Islamic University of Technology.

Islamic University of Technology

Abstract

Department of Computer Science and Engineering

Bachelor of Science in Computer Science & Engineering

NearHand: An Android based application

By

Md. Sazzad Hossain (094421)

Sanzidul Haque Seefat(094413)

Near Hand is an application that has been made especially for the recent day's leading Smartphone OS "Android'. It can be installed on any version of android (minimally it starts from gingerbread and ends up to Kitkat). Google Maps API V2 has been used in order to build up this application. The idea behind the development of such an application was to make it easier to the user the task of searching nearby jobs and house rent advertisements much and more smoothly than any application that is available at Google Play. We have served as our best in order to make the application much and more easy in case of operating, understanding etc. We hope that this application would be necessary enough in order to its related tasks and would help its user to make the task of finding nearby jobs and house rent advertisements.

Acknowledgements

Special thanks to:

Prof. Dr. M.A. Mottalib

Head of the Department

Department of Computer Science and Engineering

Special thanks to:

Mahmud Hasan

Project Supervisor

Assistant Professor

Department of Computer Science and Engineering

Contents

Chapter 1

Introduction

NearHand is an Android application that helps the users to find the nearby jobs and house rent advertisements according to their requirements. The main aim of the project is to provide the users an easy to use application that helps them to find these according to their tastes and convenience. The users can also give their opinions about the restaurants by giving a rating and writing a review. The New Google Maps API Version 2 makes the data representation smaller, so maps appear in the application faster, and use less bandwidth. . The application also makes it easy for the user to upload information regarding their necessary job and house rent advertisement finding directly from the device. The motivation behind the project is discussed in Chapter 2; the requirement analysis is done in Chapter 3 and then followed by the system architecture and design in Chapter 4, Android Framework in Chapter 5, implementation in Chapter 6, testing in Chapter 7, finally the conclusion and future work of the project.

Chapter 2

Motivation

The main motivation of this application is to learn the mobile application development. We were always curious to know how things work in mobile. This led us to develop the application. There are few such like applications in the Google Play that do the similar jobs. But most of these are complicated to use. So we have tried our best in order to develop an application with an simple interface that is easy to use. We have also integrated voice search option to make the task of finding nearby jobs and house rent advertisements much and more easier. That was the motivation behind our work throughout the project.

2.1: State of the art

As we have mentioned earlier that our main intention was to make such an application that would be helpful enough in order to find out nearby jobs and house rent advertisements easily and smoothly. In spite of having a lot of similar applications for Android platform, we think of our application as the simple but effective enough to bring out the desired result in its related fields.

2.2: Goal of the project

The main goal of our project was to learn the Android mobile development work. That's why we intended to make an application for Android. As we all know Android is becoming much and more popular in the recent day that's why we have chosen it to be our platform. Another goal can be described as to make a simple but powerful application.

Chapter 3

Requirement Analysis

The project involved analyzing the design of few applications so as to make the application more user friendly. To do so, it was really important to keep the navigations from one screen to the other well-ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the android version had to be chosen so that it is compatible with most of the Android devices. Hence Android 2.3.3 Gingerbread version was chosen. But I will run smoothly on both Jellybean and Kitkat.

3.1 Requirement Specification

3.1.1 Functional Requirements

- Graphical User interface which the user
- Provide accessibility to the application through Wi-Fi or cellular network.
- MySQL that stores the information to be displayed to the user.

3.1.2 Software Requirements

- Operating System: Windows 7
- Language: Android SDK, Java
- Database: MySQL
- Tools: Eclipse IDE, Android Plug-in for Eclipse
- Technologies used: Java, MySQL, PHP, XML.
- Debugger: Android Dalvik Debug Monitor service
- Server: XAMPP Server
- Android 4.0 or higher versions

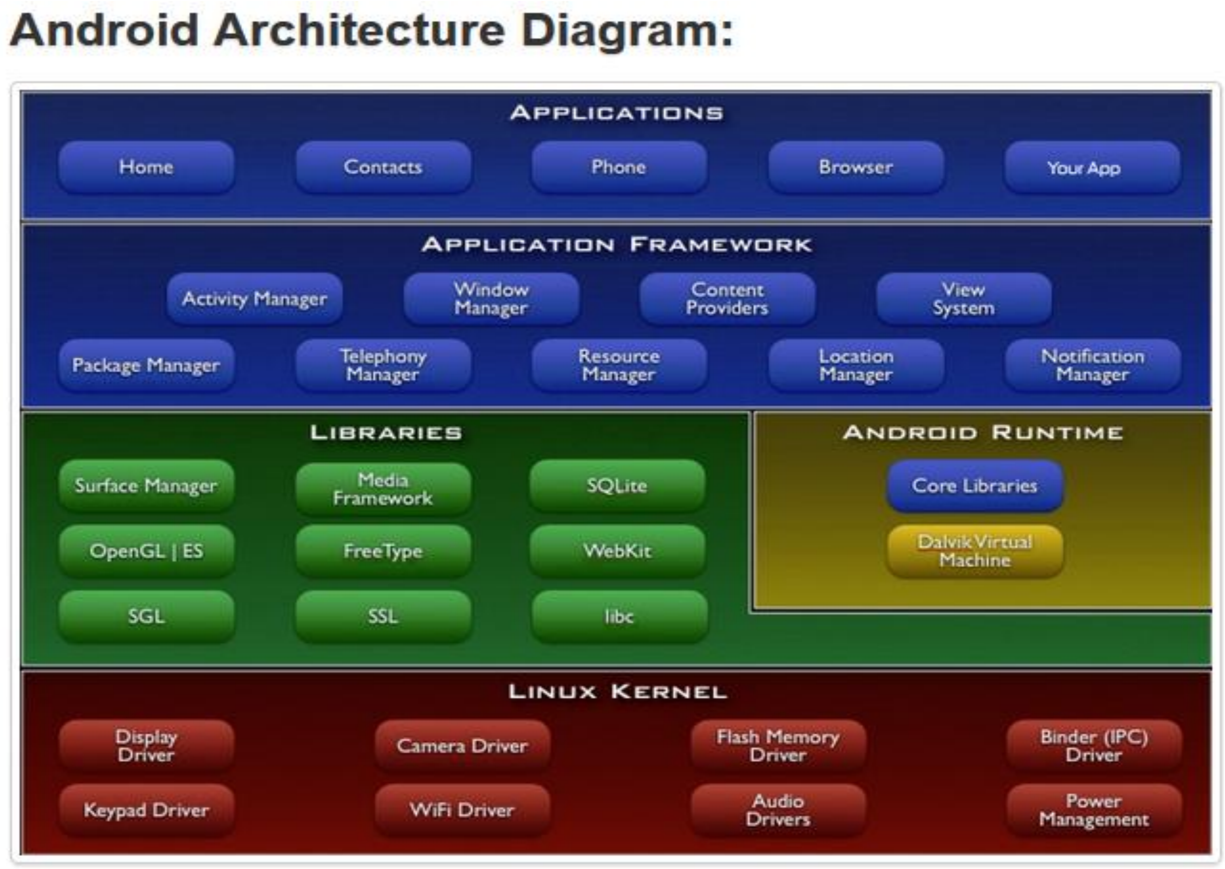
3.1.3 Hardware Requirements For developing the application:

- Processor: Intel Core 2 duo of higher
- RAM: 1 GB
- Space on disk: minimum 1.5GB for running the application:
- Device: Android version 2.3 and higher
- Minimum space to execute: 50.0MB

Chapter 4

Android

4.1 : Android Architecture:



The above figure shows the diagram of Android Architecture. The Android OS can be referred to as a software stack of different layers, where each layer is a group of several program components. Together it includes operating system, middleware and important applications. Each layer in the architecture provides different services to the layer just above it. We will examine the features of each layer in detail.

Linux Kernel

The basic layer is the Linux kernel. The whole Android OS is built on top of the Linux 2.6 Kernel with some further architectural changes made by Google. It is this Linux that interacts with the hardware and contains all the essential hardware drivers. Drivers are programs that control and communicate with the hardware. For example, consider the Bluetooth function. All devices have Bluetooth hardware in it. Therefore the kernel must include a Bluetooth driver to communicate with the Bluetooth hardware. The Linux kernel also acts as an abstraction layer between the hardware and other software layers. Android uses the Linux for all its core functionality such as Memory management, process management, networking, security settings etc. As the Android is built on a most popular and proven foundation, it made the porting of Android to variety of hardware, a relatively painless task.

Libraries

The next layer is the Android's native libraries. It is this layer that enables the device to handle different types of data. These libraries are written in c or c++ language and are specific for a particular hardware.

Some of the important native libraries include the following:

Surface Manager: It is used for compositing window manager with off-screen buffering. Off-screen buffering means you can't directly draw into the screen, but your drawings go to the off-screen buffer. There it is combined with other drawings and form the final screen the user will see. This off screen buffer is the reason behind the transparency of windows.

Media framework: Media framework provides different media codecs allowing the recording and playback of different media formats

SQLite: SQLite is the database engine used in android for data storage purposes

WebKit: It is the browser engine used to display HTML content

OpenGL: Used to render 2D or 3D graphics content to the screen

Android Runtime: Android Runtime consists of Dalvik Virtual machine and Core Java libraries.

Dalvik Virtual Machine: It is a type of JVM used in android devices to run apps and is optimized for low processing power and low memory environments. Unlike the JVM, the Dalvik Virtual Machine doesn't run .class files, instead it runs .dex files. .dex files are built from .class file at the time of compilation and provide higher efficiency in low resource environments. The Dalvik VM allows multiple instance of Virtual machine to be created simultaneously providing security, isolation, memory management and threading support. It is developed by Dan Bornstein of Google.

Core Java Libraries: These are different from Java SE and Java ME libraries. However these libraries provide most of the functionalities defined in the Java SE libraries.

Application Framework

These are the blocks that our application directly interacts with. These programs manage the basic functions of phone like resource management, voice call management etc. As a developer, you just consider these are some basic tools with which we are building our applications.

Important blocks of Application framework are:

Activity Manager: Manages the activity life cycle of applications

Content Providers: Manage the data sharing between applications

Telephony Manager: Manages all voice calls. We use telephony manager if we want to access voice calls in our application.

Location Manager: Location management, using GPS or cell tower

Resource Manager: Manage the various types of resources we use in our Application

Applications: Applications are the top layer in the Android architecture and this is where our applications are going to fit. Several standard applications come pre-installed with every device, such as:

- SMS client app
- Dialer
- Web browser
- Contact manager

As a developer we are able to write an app which replaces any existing system app. That is, you are not limited in accessing any particular feature. You are practically limitless and can whatever you want to do with the android (as long as the user of your app permits it). Thus Android is opening endless opportunities to the developer.

Android Runtime consists of Dalvik Virtual machine and Core Java libraries responsible for running the applications. It is the Application Framework that our application interacts with.

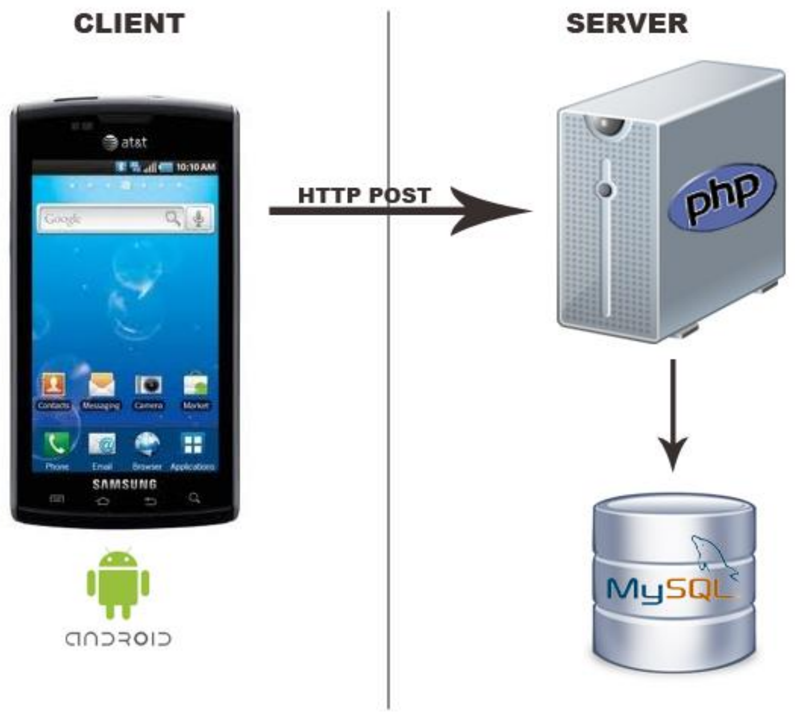


Figure: Connection between PHP (server) and Android (client) using HTTP and JSON

The architecture shown in above figure is used to show the search results, store the user details, user review and user invites. The data from android goes to webserver (PHP) to database server (MySQL). PHP is used here because of the interaction it can offer with the databases. On Android, HTTP protocol is used to connect with the webserver (PHP). JSON (JavaScript Object notation) is a lightweight text-based open standard designed for human-readable data interchange and it is used in this application to send data from Android device to PHP Script. When the application is executed, it connects the device to the PHP script on the server. PHP script fetches the response data which is encoded to JSON format and then sent back to the device. The data is parsed and displayed according to the requirement.

Android is a Linux-based operating system designed primarily for touchscreen mobile devices such as smartphones and tablet computers. Initially developed by Android, Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007 along with the founding of the Open Handset Alliance: a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. The first Android-powered phone was sold in October 2008.

Android's user interface is based on direct manipulation, using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching and reverse pinching to manipulate on-screen objects. The response to user input is designed to be immediate and provides a fluid touch interface, often using the vibration capabilities of the device to provide haptic feedback to the user. Internal hardware such as accelerometers, gyroscopes and proximity sensors are used by some applications to respond to additional user actions, for example adjusting the screen from portrait to landscape depending on how the device is oriented, or allowing the user to steer a vehicle in a racing game by rotating the device, simulating control of a steering wheel.

Android devices boot to the home screen, the primary navigation and information point on the device, which is similar to the desktop found on PCs. Android home screens are typically made up of app icons and widgets; app icons launch the associated app, whereas widgets display live, auto-updating content such as the weather forecast, the user's email inbox, or a news ticker directly on the home screen. A home screen may be made up of several pages that the user can swipe back and forth between, though Android's home screen interface is heavily customizable, allowing the user to adjust the look and feel of the device to their tastes. Third party apps available on Google Play and other app stores can extensively re-theme the home screen, and even mimic the look of other operating systems, such as Phone. Most manufacturers, and some wireless carriers, customize the look and feel of their Android devices to differentiate themselves from their competitors.

4.2: Android and Linux

Linux Android consists of a kernel based on Linux kernel version 3.x (version 2.6 prior to Android 4.0 *Ice Cream Sandwich*), with middleware, libraries and APIs written in C, and application software running on an application framework which includes Java-compatible libraries based on Apache Harmony. Android uses the Dalvik virtual machine with just-in-time compilation to run Dalvik 'dex-code' (Dalvik Executable), which is usually translated from Java byte code. The main hardware platform for Android is the ARM architecture. There is support for x86 from the Android-x86 project, and Google TV uses a special x86 version of Android. In 2013, Free scale announced Android on its i.MX processor, i.MX5X and i.MX6X series. In 2012 Intel processors began to appear on more mainstream Android platforms, such as phones.

Android's Linux kernel has further architecture changes by Google outside the typical Linux kernel development cycle. Android does not have a native X Window System by default nor does it support the full set of standard GNU libraries, and this makes it difficult to port existing Linux applications or libraries to Android. Support for simple C and SDL applications is possible by injection of a small Java shim and usage of the JNI like, for example, in the Jagged Alliance 2 port for Android.

Certain features that Google contributed back to the Linux kernel, notably a power management feature called "wake locks", were rejected by mainline kernel developers partly because they felt that Google did not show any intent to maintain its own code. Google announced in April 2010 that they would hire two employees to work with the Linux kernel community, but Greg Kroah-Hartman, the current Linux kernel maintainer for the stable branch, said in December 2010 that he was concerned that Google was no longer trying to get their code changes included in mainstream Linux. Some Google Android developers hinted that "the Android team was getting fed up with the process," because they were a small team and had more urgent work to do on Android.

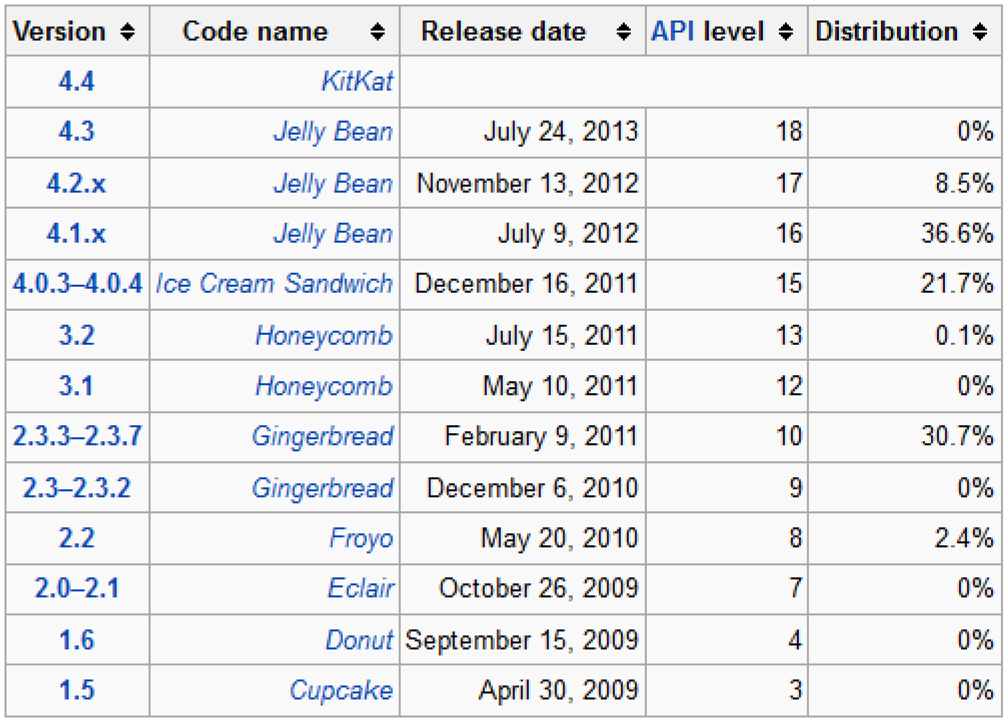
In August 2011, Linus Torvalds said that "eventually Android and Linux would come back to a common kernel, but it will probably not be for four to five years". In December 2011, Greg Kroah-Hartman announced the start of the Android Mainlining Project, which aims to put some Android drivers, patches and features back into the Linux kernel, starting in Linux 3.3.[81] Linux included the auto sleep and wake locks capabilities in the 3.5 kernel, after many previous attempts at merger. The interfaces are the same but the upstream Linux implementation allows for two different suspend modes: to memory (the traditional suspend that Android uses), and to disk (hibernate, as it is known on the desktop). Google maintains a public code repository that contains their experimental work to re-base Android off the latest stable Linux versions.

The flash storage on Android devices is split into several partitions, such as "system" for the operating system itself and "data" for user data and app installations. In contrast to desktop Linux distributions, Android device owners are not given root access to the operating system and sensitive partitions such as systems are read-only. However, root access can be obtained by exploiting security flaws in Android, which is used frequently by the open-source community to enhance the capabilities of their devices, but also by malicious parties to install viruses and malware.

Whether or not Android counts as a Linux distribution is a widely debated topic, with the Foundation and Chris DiBona,[89] Google's open-source chief, in favor. Others, such as Google engineer Patrick Brady disagree, noting the lack of support for many GNU tools, including glibc, in Android.

4.3: Platform usage

This chart provides data about the relative number of devices accessing the Play Store recently and running a given version of the Android platform as of September 4, 2013



Version	Code name	Release date	API level	Distribution
4.4	*KitKat*			
4.3	*Jelly Bean*	July 24, 2013	18	0%
4.2.x	*Jelly Bean*	November 13, 2012	17	8.5%
4.1.x	*Jelly Bean*	July 9, 2012	16	36.6%
4.0.3–4.0.4	*Ice Cream Sandwich*	December 16, 2011	15	21.7%
3.2	*Honeycomb*	July 15, 2011	13	0.1%
3.1	*Honeycomb*	May 10, 2011	12	0%
2.3.3–2.3.7	*Gingerbread*	February 9, 2011	10	30.7%
2.3–2.3.2	*Gingerbread*	December 6, 2010	9	0%
2.2	*Froyo*	May 20, 2010	8	2.4%
2.0–2.1	*Eclair*	October 26, 2009	7	0%
1.6	*Donut*	September 15, 2009	4	0%
1.5	*Cupcake*	April 30, 2009	3	0%

Android Application Development

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in the Java programming language using the Android Software Development Kit, but other development tools are available. As of October 2012, more than 700,000 applications have been developed for Android, with over 25 billion downloads. A June 2011 research indicated that over 67% of mobile developers used the platform, at the time of publication. In Q2 2012; around 105 million units of Android smartphones were shipped which acquires a total share of 68% in overall smartphones sale till Q2 2012.

Android is created by the Open Handset Alliance which is led by Google. The early feedback on developing applications for the Android platform was mixed. Issues cited include bugs, lack of documentation, inadequate QA infrastructure, and no public issue-tracking system. (Google announced an issue tracker on 18 January 2008.)In December 2007, Merge Lab mobile startup founder Adam Macbeth stated, ”Functionality is not there, is poorly documented or just doesn’t work. It’s clearly not ready for prime time.” Despite this, Android-targeted applications began to appear the week after the platform was announced. The first publicly available application was the Snake game. The Android Dev. Phone is a SIM-unlocked and hardware-unlocked device that is designed for advanced developers. While developers can use regular consumer devices purchased at retail to test and use their applications, some developers may choose not to use a retail device, preferring an unlocked or no-contract device.

A preview release of the Android SDK was released on 12 November 2007. On 15 July 2008, the Android Developer Challenge Team accidentally sent an email to all entrants in the Android Developer Challenge announcing that a new release of the SDK was available in a ”private” download area. The email was intended for winners of the first round of the Android Developer Challenge. The revelation that Google was supplying new SDK releases to some developers and not others (and keeping this arrangement private) led to widely reported frustration within the Android developer community at the time.On 18 August 2008, the Android 0.9 SDK beta was released. This release provided an updated and extended API, improved development tools and an updated design for the home screen. Detailed instructions for upgrading are available to those already working with an earlier release. On 23 September 2008, the Android 1.0 SDK (Release 1) was released. According to the release notes, it included ”mainly bug fixes, although some smaller features were added.” It also included several API changes from the 0.9 version. Multiple versions have been released since it was developed.

4.4: Android SDK

The Android software development kit (SDK) includes a comprehensive set of development tools. These includes a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, Windows XP or later; for the moment one can develop Android software on Android itself by using [AIDE - Android IDE - Java, C++] app and [Android java editor] app. The officially supported integrated development environment (IDE) is Eclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and NetBeans IDE also supports Android development via a plugin. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).

Enhancements to Android's SDK go hand in hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing.

Android applications are packaged in .apk format and stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains .dex files(compiled byte code files called Dalvik executable), resource files, etc.

Android Debug Bridge

The Android Debug Bridge (ADB) is a toolkit included in the Android SDK package. It consists of both client and server-side programs that communicate with one another. The ADB is typically accessed through the command-line interface.

The format for issuing commands through the ADB is typically:

```
adb [-d|-e|-s <serialNumber>] <command>
```

In a security issue reported in March 2011, ADB was targeted as a vector to attempt to install a rootkit on connected phones using a ”resource exhaustion attack”.

Fastboot

Fastboot is a diagnostic protocol included with the SDK package used primarily to modify the flash filesystem via a USB connection from host computer. It requires that the device be started in a boot loader or Second Program Loader mode in which only the most basic hardware initialization is performed. After enabling the protocol on the device itself, it will accept a specific set of commands sent to it via USB using a command line. Some of most commonly used fastboot commands include:

flash - Rewrites a partition with a binary image stored on the host computer.

erase - Erases a specific partition.

reboot - Reboots the device into either the main operating system, the system recovery partition or back into its bootloader.

devices - Displays a list of all devices (with the serial number) connected to the host computer.

format - Format a specific partition. The file system of the partition must be recognized by the device.

Native development kit

Libraries written in C and other languages can be compiled to ARM, MIPS or x86 native code and installed using the Android Native Development Kit. Native classes can be called from Java code running under the Dalvik VM using the System.loadLibrary call, which is part of the standard Android Java classes

Complete applications can be compiled and installed using traditional development tools.[16] The ADB debugger gives a root shell under the Android Emulator which allows native ARM, MIPS or x86 code to be uploaded and executed. Native code can be compiled using GCC on a standard PC.[16] Running native code is complicated by Android's use of a non-standard C library (libc, known as Bionic). The graphics library that Android uses to arbitrate and control access to this device is called the Skia Graphics Library (SGL), and it has been released under an open source license. Skia has back ends for both Win32 and Unix, allowing the development of cross-platform applications, and it is the graphics engine underlying the Google Chrome web browser.

Unlike Java application development based on the Eclipse IDE, the NDK is based on command-line tools and requires invoking them manually to build, deploy and debug the apps. Several third-party tools allow integrating the NDK into Eclipse and Visual Studio.

Android Open Accessory Development Kit

The Android 3.1 platform (also back ported to Android 2.3.4) introduces Android Open Accessory support, which allows external USB hardware (an Android USB accessory) to interact with an Android-powered device in a special "accessory" mode. When an Android-powered device is in accessory mode, the connected accessory acts as the USB host (powers the bus and enumerates devices) and the Android-powered device acts as the USB device. Android USB accessories are specifically designed to attach to Android-powered devices and adhere to a simple protocol (Android accessory protocol) that allows them to detect Android-powered devices that support accessory mode.

App Inventor for Android

On 12 July 2010, Google announced the availability of App Inventor for Android, a Web-based visual development environment for novice programmers, based on MIT's Open Blocks Java library and providing access to Android devices' GPS, accelerometer and orientation data, phone functions, text messaging, speech-to-text conversion, contact data, persistent storage, and Web services, initially including Amazon and Twitter. "We could only have done this because Android's architecture is so open," said the project director, MIT's Hal Abelson. Under development for over a year, the block-editing tool has been taught to non-majors in computer science at Harvard, MIT, Wellesley, Trinity College (Hartford,) and the University of San Francisco, where Professor David Wolber developed an introductory computer science course and tutorial book for non-computer science students based on App Inventor for Android.

HyperNext Android Creator

HyperNext Android Creator (HAC) is a software development system aimed at beginner programmers that can help them create their own Android apps without knowing Java and the Android SDK. It is based on HyperCard that treated software as a stack of cards with only one card being visible at any one time and so is well suited to mobile phone applications that have only one window visible at a time. HyperNext Android Creator's main programming language is simply called HyperNext and is loosely based on HyperCard's HyperTalk language. HyperNext is an interpreted English-like language and has many features that allow creation of Android applications. It supports a growing subset of the Android SDK including its own versions of the GUI control types and automatically runs its own background service so apps can continue to run and process information while in the background.

SDL

The SDL library offers also a development possibility beside Java, allowing the development with C and the simple porting of existing SDL and native C applications. By injection of a small Java shim and JNI the usage of native SDL code is possible, allowing Android ports like e.g. the Jagged Alliance 2 video game.

The Simple project

The goal of Simple is to bring an easy-to-learn-and-use language to the Android platform. Simple is a BASIC dialect for developing Android applications. It targets professional and non-professional programmers alike in that it allows programmers to quickly write Android applications that use the Android runtime components.

Similar to Microsoft Visual Basic 6, Simple programs are form definitions (which contain components) and code (which contains the program logic). The interaction between the components and the program logic happens through events triggered by the components. The program logic consists of event handlers which contain code reacting to the events.

The Simple project is not very active, the last source code update being in August 2009.

RFO Basic

RFO Basic is an on-device interpreter which provides simple access to hardware, sensors, sound, graphics, multitouch, file system, SQLite, network sockets, FTP, HTTP, Bluetooth, HTML GUI, encryption, SMS, phone, email, text-to-speech, voice recognition, GPS, math, string functions, list functions, and other essentials. It is an open source project which can produce full-fledged Android APK files. Development of RFO Basic is active, and there is a strong online community of RFO Basic! developers.

Basic4android

Basic4android is a commercial product similar to Simple. It is inspired by Microsoft Visual Basic 6 and Microsoft Visual Studio. It makes android programming much simpler for regular Visual Basic programmers who find coding in Java difficult. Basic4android is very active, and there is a strong online community of Basic4android developers.

Android APIMiner

Android APIMiner is a platform that automatically instruments the Javadoc documentation of the Android API with examples of usage, extracted from real open-source Android applications. To improve the quality of the extracted examples, APIMiner relies on an intra-procedural static slicing algorithm.

Android Developer Challenge

The Android Developer Challenge was for the most innovative application for Android. Google offered prizes totaling 10 million US dollars, distributed between ADC I and ADC II. ADC I accepted submissions from 2 January to 14 April 2008. The 50 most promising entries, announced on 12 May 2008, each received a $25,000 award to further development. It ended in early September with the announcement of ten teams that received $275,000 each, and ten teams that received $100,000 each. ADC II was announced on 27 May 2009. The first round of the ADC II closed on 6 October 2009. The first-round winners of ADC II comprising the top 200 applications were announced on 5 November 2009. Voting for the second round also opened on the same day and ended on November 25. Google announced the top winners of ADC II on November 30, with SweetDreams, What the Doodle!? and WaveSecure being nominated the overall winners of the challenge.

Community-based firmware

There is a community of open-source enthusiasts that build and share Android-based firmware with a number of customizations and additional features, such as FLAC lossless audio support and the ability to store downloaded applications on the microSD card. This usually involves rooting the device. Rooting allows users root access to the operating system, enabling full control of the phone. In order to use custom firmwares the device's bootloader must be unlocked. Rooting alone does not allow the flashing of custom firmware. Modified firmwares allow users of older phones to use applications available only on newer releases.

Those firmware packages are updated frequently, incorporate elements of Android functionality that haven't yet been officially released within a carrier-sanctioned firmware, and tend to have fewer limitations. CyanogenMod and OMFGB are examples of such firmware.

On 24 September 2009, Google issued a cease and desist letter to the modder Cyanogen, citing issues with the re-distribution of Google's closed-source applications within the custom firmware. Even though most of Android OS is open source, phones come packaged with closed-source Google applications for functionality such as the Android Market and GPS navigation. Google has asserted that these applications can only be provided through approved distribution channels by licensed distributors. Cyanogen has complied with Google's wishes and is continuing to distribute this mod without the proprietary software. It has provided a method to back up licensed Google applications during the mod's install process and restore them when the process is complete.

Java standards

Obstacles to development include the fact that Android does not use established Java standards, that is, Java SE and ME. This prevents compatibility between Java applications written for those platforms and those written for the Android platform. Android only reuses the Java language syntax and semantics, but it does not provide the full class libraries and APIs bundled with Java SE or ME. However, there are multiple tools in the market from companies such as Myriad Group and UpOnTek that provide Java ME to Android conversion services.

Figure: Android Development Tools

Chapter 5: Google Maps

Google Maps is a web mapping service application and technology provided by Google, that powers many map-based services, including the Google Maps website, Google Ride Finder, Google Transit, and maps embedded on third-party websites via the Google Maps API. It offers street maps, a route planner for traveling by foot, car, bike (beta), or with public transportation and a locator for urban businesses in numerous countries around the world. Google Maps satellite images are not updated in real time, but rather they are several months or years old. Google Maps uses a close variant of the Mercator projection, so it cannot show areas around the poles. A related product is Google Earth, a stand-alone program which offers more globe-viewing features, including showing polar areas. Google Maps is the actively used app on smartphones with over 54% of users using it between April-June 2013.

Figure : Google Maps Logo

5.1: Contiguous regions of Google Maps

Google Maps provides a route planner under "Get Directions". Up to four modes of transportation are available depending on the area: driving, public transit walking, and bicycling. In combination with Google Street View, issues like parking, turning lanes and one-way streets can be viewed before travelling. Driving directions are covered as follows:

Most countries of mainland Eurasia and Africa are covered contiguously, including the United Kingdom, Ireland, the Canary Islands, Cyprus, Malta, Sri Lanka, most of Indonesia and Timor-Leste. China mainland, Hong Kong, Macau, Israel (including parts of the West Bank), Jordan, Lebanon and North Korea have directions available without connection to other states. Only public transit directions are provided for South Korea. All countries of mainland North and Central America are covered contiguously.

All countries of mainland South America are covered. Argentina, Bolivia, Brazil, Chile, Ecuador, Paraguay, Peru, Trinidad and Tobago, Uruguay and Venezuela are treated contiguously, whereas Colombia, French Guiana, Guyana and Suriname are not connected to other states.

All inhabited countries and territories in the Caribbean are covered, though in general there are no connections between islands.

Additionally, American Samoa, Australia, the Azores, Brunei, Cape Verde, The Comoros, The Cook Islands, the Faroe Islands, The Federated States of Micronesia, Fiji, French Polynesia, Guam, Hawaii, Iceland, Japan, Madagascar, the Maldives, Mauritius, Mayotte, New Caledonia, New Zealand, Niue, Northern Mariana Islands, Palau, the Philippines, Réunion, São Tomé and Príncipe, the Seychelles, Samoa, Taiwan, Tonga, Vanuatu, Wallis and Futuna are covered as stand-alone regions, as are Nuuk in Greenland, Sabah in Malaysia, parts of Papua New Guinea, parts of Solomon Islands and Socotra in Yemen.

5.2: Implementation

Like many other Google web applications, Google Maps uses JavaScript extensively. As the user drags the map, the grid squares are downloaded from the server and inserted into the page. When a user searches for a business, the results are downloaded in the background for insertion into the side panel and map; the page is not reloaded. Locations are drawn dynamically by positioning a red pin (composed of several partially transparent PNGs) on top of the map images. A hidden IFrame with form submission is used because it preserves browser history. The site also uses JSON for data transfer rather than XML, for performance reasons. These techniques both fall under the broad Ajax umbrella. The result is termed a slippy map and is implemented elsewhere in projects like OpenLayers.In October 2011, Google announced MapsGL, a WebGL version of Maps with better renderings and smoother transitions.

5.3: Extensibility and customization

As Google Maps is coded almost entirely in JavaScript and XML, some end users have reverse-engineered the tool and produced client-side scripts and server-side hooks which allowed a user or website to introduce expanded or customized features into the Google Maps interface.

Using the core engine and the map/satellite images hosted by Google, such tools can introduce custom location icons, location coordinates and metadata, and even custom map image sources into the Google Maps interface. The script-insertion tool Grease monkey provides a large number of client-side scripts to customize Google Maps data.

Combinations with photo sharing websites, such as Flickr, are used to create "memory maps". Using copies of the Keyhole satellite photos, users have taken advantage of image annotation features to provide personal histories and information regarding particular points of the area.

5.4: Google Maps API

After the success of reverse-engineered mashups such as chicagocrime.org and housingmaps.com, Google launched the Google Maps API in June 2005 to allow developers to integrate Google Maps into their websites. It is a free service, and currently does not contain ads, but Google states in their terms of use that they reserve the right to display ads in the future.

By using the Google Maps API, it is possible to embed Google Maps site into an external website, on to which site specific data can be overlaid. Although initially only a JavaScript API, the Maps API was expanded to include an API for Adobe Flash applications (but this has been deprecated), a service for retrieving static map images, and web services for performing geocoding, generating driving directions, and obtaining elevation profiles. Over 350,000 web sites use the Google Maps API, making it the most heavily used web application development API.

The Google Maps API is free for commercial use, provided that the site on which it is being used is publicly accessible and does not charge for access, and is not generating more than 25 000 map accesses a day. Sites that do not meet these requirements can purchase the Google Maps API for Business.

The success of the Google Maps API has spawned a number of competing alternatives, including the Yahoo! Maps API, Bing Maps Platform, MapQuest Development Platform, and OpenLayers.

In September 2011, Google announced it would discontinue a number of its products, including Google Maps API for Flash.

5.5: Google Maps for Mobile

In October 2005, Google introduced a Java application called Google Maps for Mobile, intended to run on any Java-based phone or mobile device. Many of the web-based site's features are provided in the application.

On November 28, 2007, Google Maps for Mobile 2.0 was released. Its location service can work with or without a GPS receiver. The ”my location” feature uses the GPS / Assisted GPS location of the mobile device, if available, supplemented by determining the nearest wireless networks and cell sites. The software looks up the location of the cell site using a database of known wireless networks and sites. By triangulating the different signal strengths from different cell transmitters and then using their location property (retrieved from the online cell site database), My Location determines the user's current location. Wireless network location method is calculated by discovering the nearby Wi-Fi hotspots and using their location property (retrieved from the online Wi-Fi database, in the same way as the cell site database) to further discover the user's location. The order in which these take precedence is:

GPS-based services

WLAN-, Wi-Fi-based services

Cell transmitter-based services

The software plots the streets in blue that are available with a yellow icon and a green circle around the estimated range of the cell site based on the transmitter's rated power (among other variables). The estimate is refined using the strength of the cell phone signal to estimate how close to the cell site the mobile device is.

As of December 15, 2008, this service was available for the following platforms:

Android

iOS

PlayStation Vita system software

Windows Mobile

Symbian OS

BlackBerry OS

Palm OS

Palm webOS

On November 4, 2009, Google Maps Navigation was released in conjunction with Google Android OS 2.0 Eclair on the Motorola Droid, adding voice commands, traffic reports, and street view support. The initial release was limited to the United States. The service was launched in the UK on 20 April 2010 and in large parts of continental Europe on June 9, 2010 (including Austria, Belgium, Canada, Denmark, France, Germany, Italy, the Netherlands, Portugal, Spain, and Switzerland).In March 2011 Google Vice President of Location Service, Marissa Mayer said that Google provided map services to 150 million users.

It provided voice guidance and live traffic information in the Indian cities of Bengaluru, Mumbai, New Delhi, Chennai, Pune and Hyderabad from September 5, 2012.

In June 2012, Apple announced that they would replace Google Maps with their own maps service from iOS 6. The application software and maps that replaced Google Maps came under fierce criticism for several problems and errors. However, on December 13, 2012, Google announced the availability of Google Maps in the Apple App Store, starting with the iPhone version. Just hours after the Google Maps iOS app was released, it became the top free app in the App Store. While Google maps as an app no longer enjoys the tight integration into the iOS as it had as a system service, the new version offers some features said to be easier to use and more intuitive than even the Android service.

It was announced on December 6, 2012 that Google Maps would make its way to the Wii U, Nintendo's new gaming system. Accessibility to Google Maps on the Wii U would come via a downloadable app.

5.6: Google Maps Android 2.0

Cell phones are being increasingly used for navigation assistance. Google Maps Navigation for Android 2.0 is free.

Features provided in the application:

- Search in plain English
- Search by voice
- Search along route
- Satellite view
- Street View

5.7: Use

Maps and related information are not included in the installed Google Maps for Android file; an Internet connection is required, as for iPhone's Google Maps application. An automatic map caching feature temporarily stores recently viewed areas, reducing the amount of data to be downloaded. There is also a 'Download Map Area' feature in the Labs section of Google Maps which enables the user to download the basic road map and landmark data for an area of 10 square miles (26 km2) around any point; but even after a map of an area is downloaded, a data connection is still needed to "see satellite view and 3D buildings, search for Places and get directions.". The user can download limited areas several times to cover the total area desired. Google suggests that users can make use of this feature to download the map of a foreign city before traveling to visit it, to avoid the need for expensive international roaming data downloading.

Offline version

As of version 6.9 Google Maps offers offline access to downloaded maps of certain countries, a feature only offered in Labs of previous versions. 150 countries are available offline. For example one country that is not available offline is Colombia.

Google Maps parameters

In Google Maps, URL parameters are sometimes data-driven in their limits and the user interface presented by the web may or may not reflect those limits. In particular, the zoom level (denoted by the z parameter) supported varies. In less populated regions, the supported zoom levels might stop at around 18. In earlier versions of the API, specifying these higher values might result in no image being displayed. In Western cities, the supported zoom level generally stops at about 20. In some isolated cases, the data supports up to 23 or greater, as in these elephants or this view of people at a well in Chad, Africa. Different versions of the API and web interfaces may or may not fully support these higher levels.

As of October 2010, the Google map viewer updates its zoom bar to allow the user to zoom all the way when centered over areas that support higher zoom levels.

Map projection

Google Maps is based on a close variant of the Mercator projection. If the Earth were perfectly spherical, the projection would be the same as the Mercator. Google Maps uses the formula for the spherical Mercator, but the coordinates of features on Google Maps are the GPS coordinates based on the WGS 84 datum. The difference between a sphere and the WGS 84 ellipsoid causes the resultant projection not to be precisely conformal. The discrepancy is imperceptible at the global scale but causes maps of local areas to deviate slightly from true ellipsoidal Mercator maps at the same scale.

Assuming that dE and dN are the components of infinitesimal local ENU coordinates, their breadth and length projected on the map are described as follows:

$$dx = a_{\mathrm{map}} \left(\frac{a}{\sqrt{1 - e^2 \sin^2 \varphi}} \right)^{-1} \sec \varphi \, dE,$$

$$dy = a_{\mathrm{map}} \left(\frac{a(1 - e^2)}{\left(1 - e^2 \sin^2 \varphi\right)^{3/2}} \right)^{-1} \sec \varphi \, dN,$$

5.8: Google Maps Android API v2

The new Google Maps Android API v2 allows you to offer interactive, feature-rich maps to users of your Android application. This version offers the following improvements:

- The API is now distributed as part of the Google Play services SDK, which you can download with the Android SDK Manager. To learn how to install the package, see Installing the Maps API SDK.
- Maps are now encapsulated in the MapFragment class, an extension of Android's Fragment class. Now you can add a map

as a piece of a larger Activity. With a MapFragment object, you can show a map by itself on smaller screens, such as mobilephones, or as a part of a more complex UI on larger-screen devices, such as tablets.

- Because maps are encapsulated in the MapFragment class, you can implement them by extending the Android standard Activity class, rather than extending the MapActivity used in version 1.

- The Maps API now uses vector tiles. Their data representation is smaller, so maps appear in your apps faster, and use less bandwidth.

- Caching is improved, so users will typically see a map without empty areas.

- Maps are now 3D. By moving the user's viewpoint, you can show the map with perspective.

Chapter 6

Near Hand: An overview

We have named our Google Maps Client as "Near Hand". It will fetch the Google Maps through Wi-Fi or EDGE or any other means of data connections. After loading the map, it will directly locate the user's location through GPS and will start to show nearby available jobs and House rent advertisements. Anyone using the application will be able to upload information about job description, its nature, salary etc. and house rent information such as rent, address, contact information etc. We have also implemented a voice based search option. The user will be able to search for jobs or house rent advertisements through his/her voice. The application will take the voice as the input and will return with its matched results.

Our intention was to focus on the easiness of the application and from our point of view perhaps we are successful. There is also no option for pop-up advertisements. So the user of the application will also not be disappointed as because there are no pop-up advertisements. We have strictly followed the open source license in order to ensure the application's legal validity.

6.1: AndroidManifest.xml file:

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

<uses-sdk
android:minSdkVersion="8"
android:targetSdkVersion="18" />

<permission
android:name="in.wptrafficanalyzer.locationremovesinglemarker.permission
.MAPS_RECEIVE"
 android:protectionLevel="signature" />
```

```
<uses-permission
android:name="in.wptrafficanalyzer.locationremovesinglemarker.permission
.MAPS_RECEIVE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES
" />
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />

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

<application
android:allowBackup="true"
android:icon="@drawable/ic_launcher"
android:label="@string/app_name"
android:theme="@style/AppTheme" >
<activity
android:name="com.sa.nearhand.MainActivity"
android:configChanges="orientation|keyboardHidden"
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="com.sa.nearhand.MapActivity"
android:configChanges="orientation|keyboardHidden" >
</activity>

<meta-data
android:name="com.google.android.maps.v2.API_KEY"
android:value="AIzaSyCs1DqvzdPT2ELvIk4fszYbHCmY7vgmDc0" />

<activity
android:name="com.sa.nearhand.InsertData"
android:label="@string/title_activity_insertdata" >
</activity>
</application>

</manifest>
```

6.2: Activities:

An activity provides a means of interaction to the user. It provides a window where the UI can be designed according to the window. Almost all activities interact with the users.

This application has the following activities: AddItemAsyncTask Activity, DashboardLayout Activity, GlobalConstant Activity, GMapV2GetRouteDirection Activity, InsertData Activity, JSONParser Activity, LoadNearPlace Activity, MainActivity, MapActivity, NearPlaceItem Activity and Network Activity.

6.3: Intents:

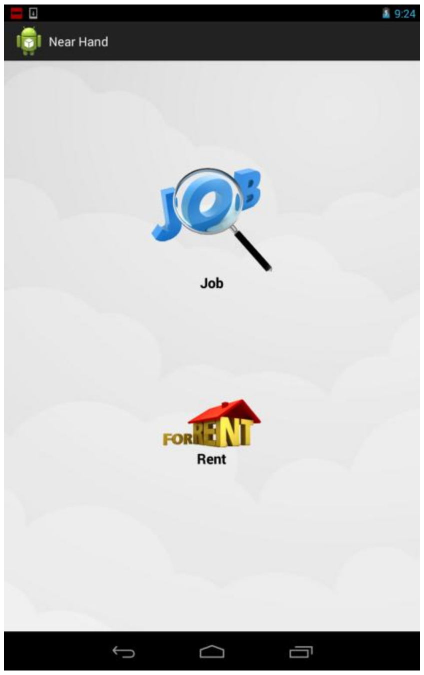
The activities are activated through messages called intents. It basically has
Information that is required by the component. It helps in launching the activity or to do something with the existing activity. An intent passed to startActivity() is delivered only to an activity. The intent object holds the name of the component that handles the event, the action to be performed, data on which the operation is performed, the
key value pairs to send some additional information. The “Near Hand” Application makes use of intents to send the users’ desired data after user’s choice through executing different activities.

Chapter 7

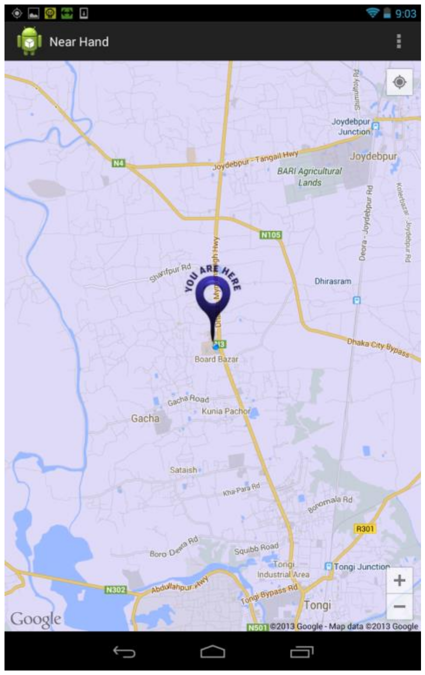
Graphical User Interface

Some screenshots of the Near Hand applications are provided below:

Start Screen



Locator (By GPS)



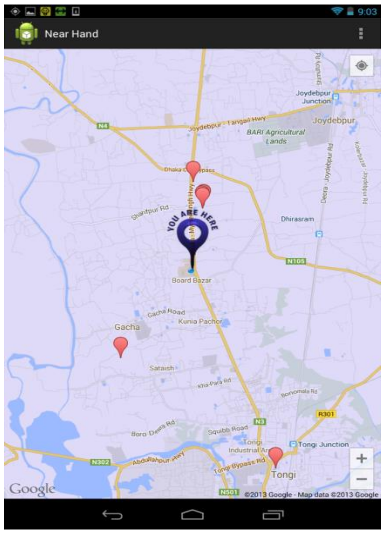
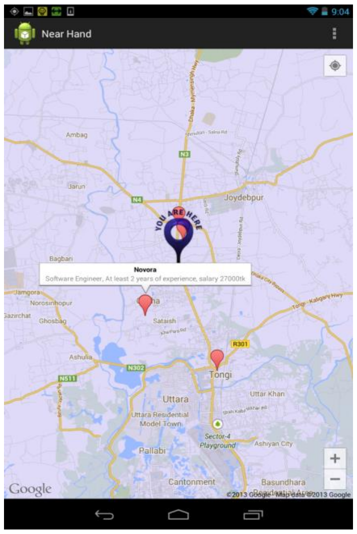
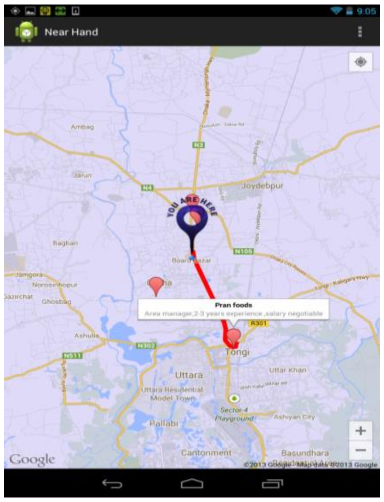


Fig:showing nearer Jobs



Novora
Software Engineer, At least 2 years of experience, salary 27000tk

Fig: Displaying job details



Pran foods
Area manager,2-3 years experience ,salary negotiable

Fig:Showing Route to the workplace

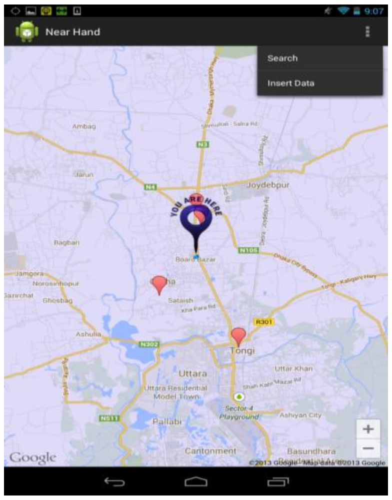


Fig:Displaying nearer houses for rent

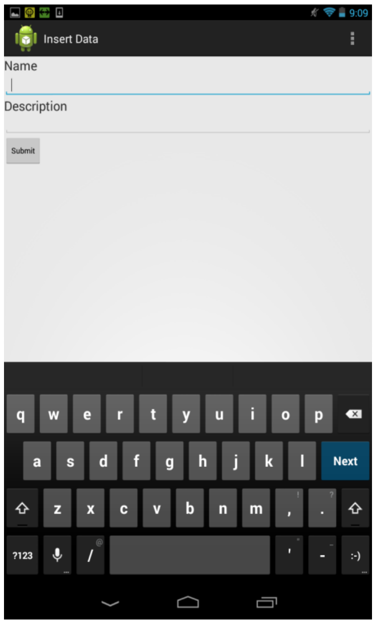
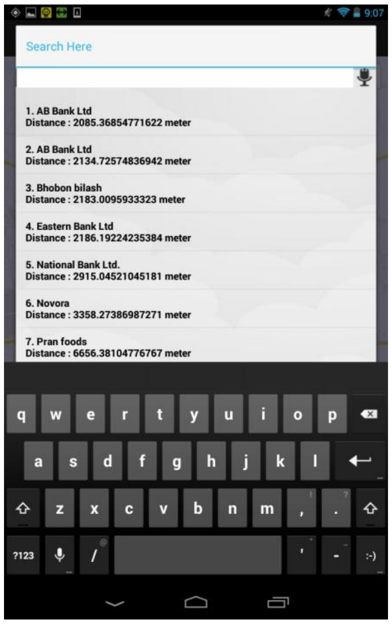


Fig: Data insertion



1. AB Bank Ltd
Distance : 2085.36854771622 meter

2. AB Bank Ltd
Distance : 2134.72574836942 meter

3. Bhobon bilash
Distance : 2183.0095933323 meter

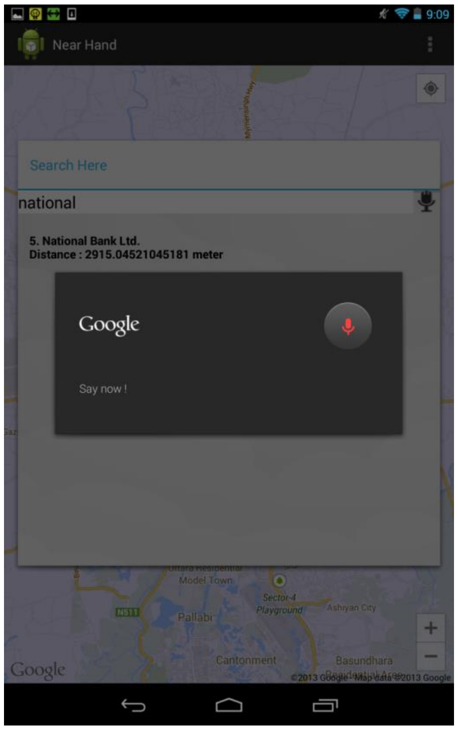
4. Eastern Bank Ltd
Distance : 2186.19224235384 meter

5. National Bank Ltd.
Distance : 2915.04521045181 meter

6. Novora
Distance : 3358.27386987271 meter

7. Pran foods
Distance : 6656.38104776767 meter

Fig: Showing search results



national

5. National Bank Ltd.
Distance : 2915.04521045181 meter

Fig: Voice search interface

Chapter 8

Conclusion

This is our first attempt in developing a mobile application which gave us a basic understanding of development and challenges of mobile application development. The main aim of the project is to provide an easy to use application for searching the nearby jobs and house rent advertisements. The application has been implemented and tested on real devices. We are again grateful to Mr. Mahmud Hasan for immensely helping us.

References:

[1] Android Architecture

http://en.androidwiki.com/wiki/Diagram_of_the_Android_architecture

[2] Android Architecture Description

http://www.android-app-market.com/android-architecture.html

[3] Connection between PHP (server) and Android (client) Using HTTP and JSON

http://fahmirahman.wordpress.com/2011/04/21/connection-between-php-server-and-android-client-using-http-and-json/

[4] Android Developer Guide

http://developer.android.com/

[5] Android Coding

http://stackoverflow.com/

[6] Crawling and Parsing

http://www.oracle.com/technetwork/java/index.html

[7] Android

http://wikipedia.com/android