



EVITE MOBILE APPLICATION

A PROJECT REPORT

Submitted by

M. SANTHOSH RAJ

212711205085

in partial fulfillment for the award of the

degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

SRI VENKATESWARA COLLEGE OF ENGINEERING

SRIPERUMBUDUR

ANNA UNIVERSITY::CHENNAI 600 025

APRIL 2015



EVITE MOBILE APPLICATION

A PROJECT REPORT

Submitted by

M. SANTHOSH RAJ

212711205085

in partial fulfillment for the award of the

degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

SRI VENKATESWARA COLLEGE OF ENGINEERING

SRIPERUMBUDUR

ANNA UNIVERSITY::CHENNAI 600 025

APRIL 2015

BONAFIDE CERTIFICATE

Certified that this project report titled “EVITE MOBILE APPLICATION” is the bonafide work of "SANTHOSH RAJ M" who carried out the project work under my supervision, for the partial fulfilment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in INFORMATION TECHNOLOGY.

SIGNATURE

Dr. D. BALASUBRAMANIAN, Ph.D,
HEAD OF THE DEPARTMENT ,

Department of Information Technology,
Sri Venkateswara College of
Engineering,
Sriperumbudur 602 117,
India.

SIGNATURE

Mr. T. SUKUMAR, M. Tech
SUPERVISOR,
Assistant Professor,

Department of Information Technology,
Sri Venkateswara College of
Engineering,
Sriperumbudur 602 117,
India.

Place : Chennai

Date:

INTERNAL EXAMINER

EXTERNAL EXAMINE

ACKNOWLEDGEMENT

We express our sincere thanks to **Dr. R.Parthiban Ph.D**, Principal, Sri Venkateswara College Of Engineering, for being a source of inspiration throughout the study in this college.

We express our sincere thanks to **Dr.D.Balasubramanian, Ph.D**, Head Of The Department, Information Technology, for his permission and encouragement accorded to carry out this project. We also thank **Ms.Swarna Parvathi S, M.E.** ,Associate Professor, for her continuous guidance and support towards completion of the project.

With profound respect, we express our deep sense of gratitude and sincere thanks to our guide, **Mr. Sukumar T, M. Tech** Assistant Professor, Department of Information Technology , for her valuable guidance and motivation throughout this project which helped us finding new approaches towards problem solving.

We express our heart-felt gratitude to **Mr. Venkata Raman S S, Senior Technical Lead, Vulcantech Software India**, and his colleagues for their constant support and invaluable guidance throughout the project.

We are also thankful to **Ms.Indumathi A, M.Tech**, Assistant Professor, Department of Information Technology, our Project Coordinator, who proved to be a pillar of support for us as she was very accepting of our success and setbacks during the project in equal measure. We convey our sincere thanks to all the faculty members of the department who guided us and provided timely assistance and guidance.

ABSTRACT:

Evite is a social-planning Application for creating, sending, and managing online invitations. It is nothing but a mobile application which has several in built features that makes it quite easy for a person to invite all his near and dear ones. The user first registers his number through the app and an auto-generated One Time Password (OTP) is sent by the server as a response to his mobile number as a pin for authentication. Then he enters the same and verifies his phone number. Then he is asked to verify himself with the help of Finger Print and is checked with the fingerprint registered on the device so that in the future he can verify himself with only the fingerprint and only he can create new event and sees the list of events he has to attend. Then, he also can create his own events and include a list of people whom he wishes to invite with the help of his contacts app which is inbuilt in almost every phone and select the location of the event with the help of the Google maps api and fix the place of the event as he touches the location of the event on the map and also the date and time with the help of a date and time picker which is a feature of Android. Those who have the app installed in their phone will be receiving a push notification.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	5
	LIST OF TABES	9
	LIST OF FIGURES	10
	LIST OF ABBREVIATIONS	11
1.	EVITE	12
	1.1. What is Evite?	12
	1.2. Purpose and scope of the project	13
	1.3. Literature survey	13
	1.4. Existing system and its limitations	14
	1.5. Proposed system	15
2.	SYSTEM REQUIREMENTS SPECIFICATION	16
	2.1. System Requirements Specifications	16
	2.2. System Requirements	16
	2.2.1. Hardware Requirements	16
	2.2.2. Software Requirements	17

3.	DESIGN	19
	3.1. Blocks	19
	3.1.1. Types of Blocks	19
	3.1.1.1. Registration Block	19
	3.1.1.2. Verification Block	20
	3.1.1.3. Authentication Block	20
	3.1.1.4. Events Block	20
	3.2. Architecture Diagram	21
4.	BLOCKS	22
	4.1. Registration Block	22
	4.2. Verification Block	24
	4.3. Authentication Block	25
	4.4. Events Block	27
	4.4.1. Events List	28
	4.4.2. Add Events	29
	4.4.2.1. Date and Time Picker	30
	4.4.2.2. Location Picker	32

	4.4.2.3. Contact Picker	35
	4.4.2.4. Events Detail Block	36
5.	Conclusion and Enhancements	39
	References	40
	Appendices	41
	A. Android	41

LIST OF TABLES

TABLE NO.	TITLE	PAGE NO.
2.1	Hardware Requirement	17
2.2	Software Requirement	18

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
1.1	Evite Reviews	14
2.1	Samsung s5 with finger Print	17
2.2	Android O.S	18
3.1	Block Diagram	21
4.1	Registration Screen	23
4.2	Verification Screen	25
4.3	Finger Print Screen	27
4.4	Events List Screen	29
4.5	Date Picker	31
4.6	Time Picker	32
4.7	Location Picker	34
4.8	Contact Picker	35
4.9	Event Detail Screen	37
4.10	Event Location Screen	38

LIST OF ABBREVIATIONS

RB	Registration Block
VB	Verification Block
AB	Authentication Block
EB	Events Block
AE	Add Events
EL	Events List
EDB	Event Details Block

Chapter 1

EVITE

Evite is a social-planning website for creating, sending, and managing online invitations. Evite was launched in 1998. The website is a free, advertisement-supported service. It was acquired by conglomerate IAC/InterActiveCorp in 2001. Liberty Media acquired it in 2010.

An event organizer creates an online invitation through the website via a simple interface. This online invitation is colloquially referred to as "an Evite". The host enters e-mail addresses of prospective guests and Evite sends e-mails to the guests. Each guest is in one of four categories: "Attending", "Not Attending", "Maybe", and "Not yet replied". Guests move themselves from the "Not yet replied" category to one of the three "replied" categories and can write additional comments. Evite is a mobile application for android operating system. Evite is also available on iOS which is basically a social planning application. Evite was initially a software.

1.1. What is Evite?

Evite is a social-planning Application for creating, sending, and managing online invitations. It is nothing but a mobile application which has several in built features that makes it quite easy for a person to invite all his near and dear ones. The user first registers his number through the app and an auto-generated One Time Password (OTP) is sent by the server as a response to his mobile number as a pin for authentication. Then he enters the same and verifies his phone number. Then he is asked to verify himself with the help of Finger Print and is checked with the

fingerprint registered on the device so that in the future he can verify himself with only the fingerprint and only he can create new event and sees the list of events he has to attend. Then, he also can create his own events and include a list of people whom he wishes to invite with the help of his contacts app which is inbuilt in almost every phone and select the location of the event with the help of the Google maps api and fix the place of the event as he touches the location of the event on the map and also the date and time with the help of a date and time picker which is a feature of Android. Those who has the app installed in their phone will be receiving a push notification.

1.2. Purpose and scope of the project:

As mentioned above, the purpose of Evite is to send invitations about the events you host to all others whom you want to invite. You can have any number of events and any number of people for that event.

You have to specify the location of the event and select the type of event from a predefined set of events or you can also create a new kind of event and press send button to send it to the group of contacts you wish to send. If the members in your list have the evite app installed in their mobile, they will get a push notification or else they will be sent an e-mail regarding the event.

The evite software has been in existence and been used by many customers all over the world. As the mobile applications are fast replacing the PC software, this app will be of greater use to people.

1.3.Literature Survey

Reference: “Xiping Hu, Victor C.M. Leung”., “Multidimensional Context-

Aware Social Network Architecture for Mobile Crowdsensing. I had made a survey of the Evite software which has been in existence. Evite was launched in 1998. The website is a free, advertisement-supported service. It was acquired by conglomerate IAC/InterActiveCorp in 2001. Liberty Media acquired it in 2010. The purpose of it is the same which is used for sending invitation about the events to the people you wish to. It has been a greatly useful software in the western countries which avoids the extra overhead of ringing up your colleagues for the event which you want them to be. According to the survey 7 out of every 10 people use this software and has been proven to be greatly useful to people using it.

1.4. Existing system and its limitations:

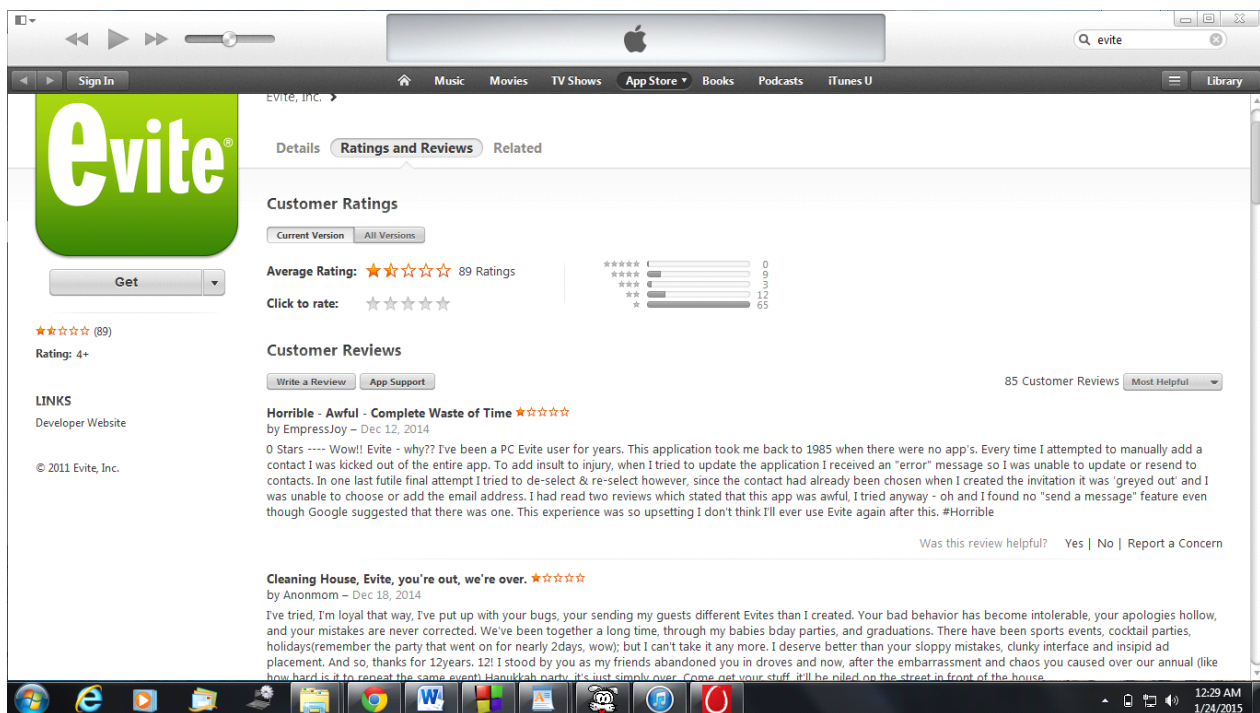


Fig. 1.1: Evite Reviews

As you can see in the image, from the customer reviews it is clear that the app

crashes often which makes it annoying to use. This is a very major problem in an application where the user is forcibly terminated from where he wants or more accurately what he wants to be. There are certain conditions and exceptions that are not handled properly in the application which makes it to inconvenient and annoying to use. Moreover the existing system does not allow the users to select the location of the event by a touch on the map. Evite software has been in existence for a while and has been friendlier to use and has been widely used software in many parts of the world. As we all are more addicted to use mobiles than laptops or PC's, every software is transformed into a mobile application and same is the case with evite also. All that the users want is a good mobile application of evite which does not lag or hand or forcibly exit.

1.5. Proposed system:

The proposed system has been designed in such a way that it overcomes all these limitations and prevents the app from crashing. Along with that we have also added some optimization feature which makes the app function faster and perform better.

Optional invitation features include:

Adding polls for your guests to answer — find out what movie, food or music everybody prefers

Letting guests select items to bring — great for potlucks and other group-run events.

Chapter 2

System Requirements Specification

2.1. System Requirements Specification

Here, we will be discussing in detail about the system requirements. We have specified the system requirements below.

2.2. System Requirements

Basically, there are two kinds of Requirements called as

- Hardware Requirements
- Software Requirements

2.2.1. Hardware Requirements

- The first and foremost requirement is a mobile phone which could run that app.
- Smart phones with an inbuilt fingerprint scanner should be used to run the app because we are using the feature of fingerprint authentication used to run the application.
- They must have a minimum of 512MB RAM. They must also have 20MB of free space in their memory as memory requirement for the app is about 17.5MB.

Table 2.1: Hardware Requirements

PART	SPECIFICATION
Processor	Quad-core 2.5 GHz
RAM	512MB
Memory	20MB
Finger Print Scanner	Yes



Fig. 2.1: Samsung s5 with finger print

2.2.1. Software Requirements

- The Software requirement is quite simple when compared to the hardware requirements.
- All we need is an Android Operating System. The minimum version of which should be 3.0.

Table 2.2: Software Requirements

PART	SPECIFICATION
Operating System	Android
Version	3.0



Fig. 2.2: **Android O.S**

Chapter 3

DESIGN

In this Chapter we will be discussing in detail about the system design and the Interfaces.

3.1. Blocks

The functionality of Evite is mainly divided into many blocks. The following chapters give a description about them.

3.1.1. Types of Blocks

As mentioned above the different functioning blocks of evite are:

- Registration Block (RB)
- Verification Block (VB)
- Authentication block (AB)
- Events Block (EB)
 - i) Add Event (AE)
 - ii) Event list (EL)
- Event Details Block (EDB)

3.1.1.1. Registration Block:

This is the first block in the evite. So, any user who wishes to use evite should cross this block to move on to the next block which is the verification block in the next step.

3.1.1.2. Verification Block:

This is the Second block coming next to the registration block in the evite. Here, the user enters the Verification code (pin) sent to him via sms to identify his phone number.

3.1.1.3. Authentication Block:

This is the next block coming next to the Verification Block in the evite block diagram. Here, the user verifies his identity with the help of his finger print which is more secure than any other authentication mechanisms.

3.1.1.4. Events Block:

This block is pre final block in the evite block diagram. It is further sub divided into the following 2 blocks:

- 1) Add Event (AE)
- 2) Events list (EL)

3.1.1.5 .Events Details Block:

This block is the final block where the user can see the details about the event such as date, time and place of the event.

3.2. Architecture Diagram

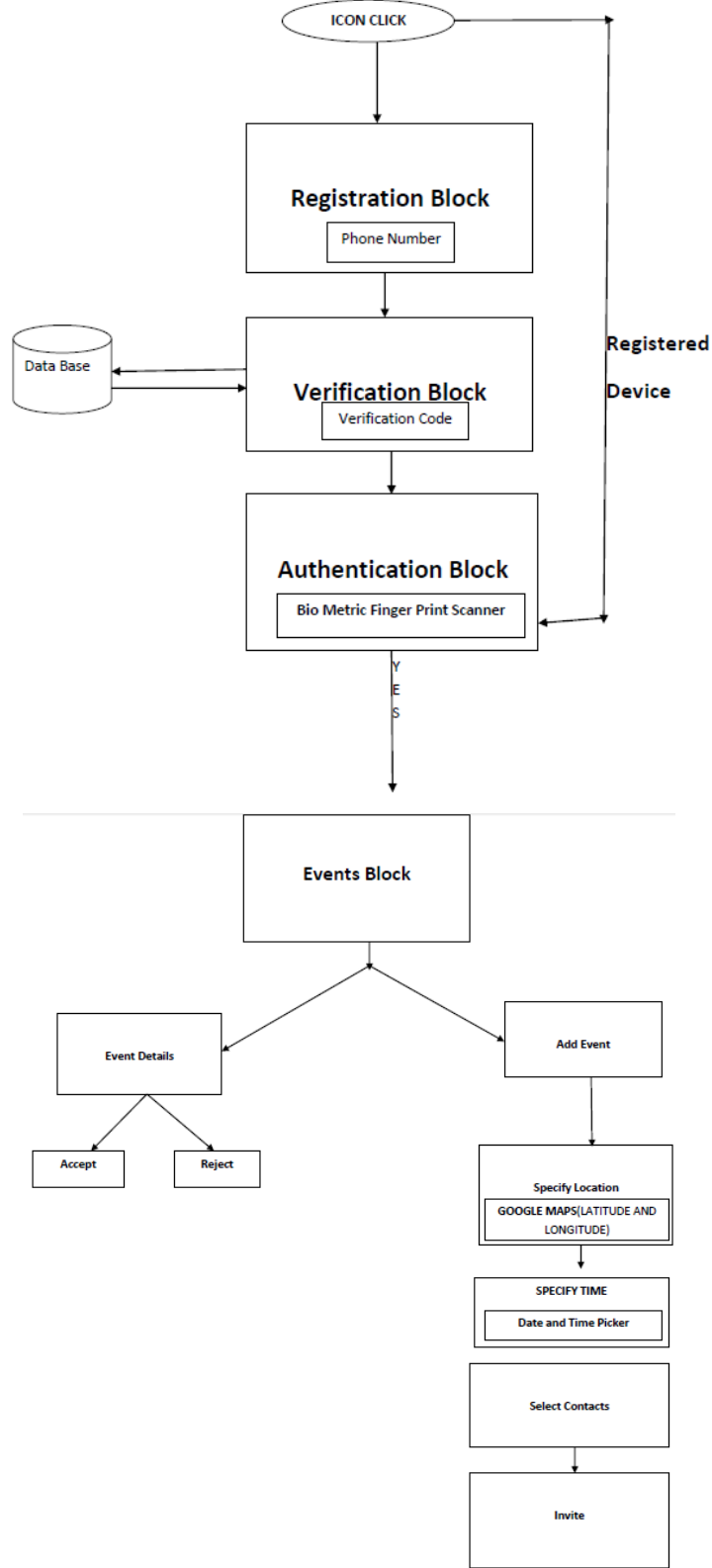


Fig. 3.1: Block Diagram

Chapter 4

BLOCKS

4. Blocks

Here in this chapter we will be discussing in detail about the various blocks involved in evite.

4.1. Registration Block (RB)

This is the first block when the user opens the app. It is also abbreviated as RB. Here, he is asked to register his phone number for future use. The first screen asks the user to enter his phone number in the text field and also there is a button provided for him to press when he enters his number. He can enter only his number and not any alphabets because provision is made so that only the number pad appears on the screen and therefore he is forced to enter only phone number.

Once the user enters the number he has press the register button and the request is being sent to the server using the json and HTTP post method. Till the response is received a progress dialog appears on the screen and if the response code is 200 and the result has a success he will be redirected to the next screen. The screen shot of the registration block is shown below.

When the user opens the Evite for the second time he is automatically redirected to the 3rd screen which is the Authentication Screen. We are actually creating a Boolean variable named hasPhoneNumber and setting it to true once the user crosses the Verification Block which is the next block. We are searching for the status of the variable when ever the app opens and if it is set to true we move on the third screen (kruskal's algorithm) there by finding the shortest path.

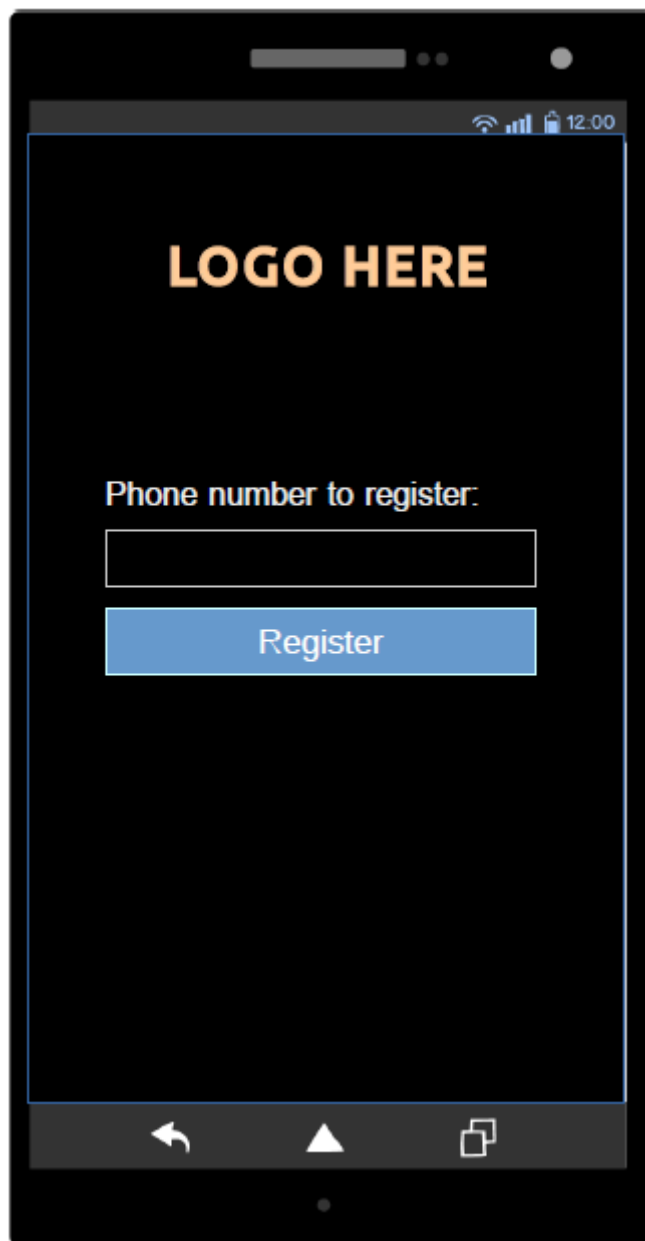


Fig. 4.1: *Registration Screen*

4.2. Verification Block (VB)

This block is one of the key blocks in the evite related with the security. This is also the second block in the Evite. It is also abbreviated as VB. Here, the user is asked to enter the verification PIN (stored in the hash table) which has been sent to him from the server with the help of rand() function. The user after receiving this sms, has to enter the same in the space provided in the verification block. Here also the provision is made so that the user can enter only the numbers as only the number pad appears on the screen and once he enters the verification pin he presses the verify button and a progress dialog appears on the screen as the request is sent to the server with the help of the HTTP post method using the json . Till the response is received a progress dialog appears on the screen and if the response code is 200 and the result has a success he will be redirected to the next screen.

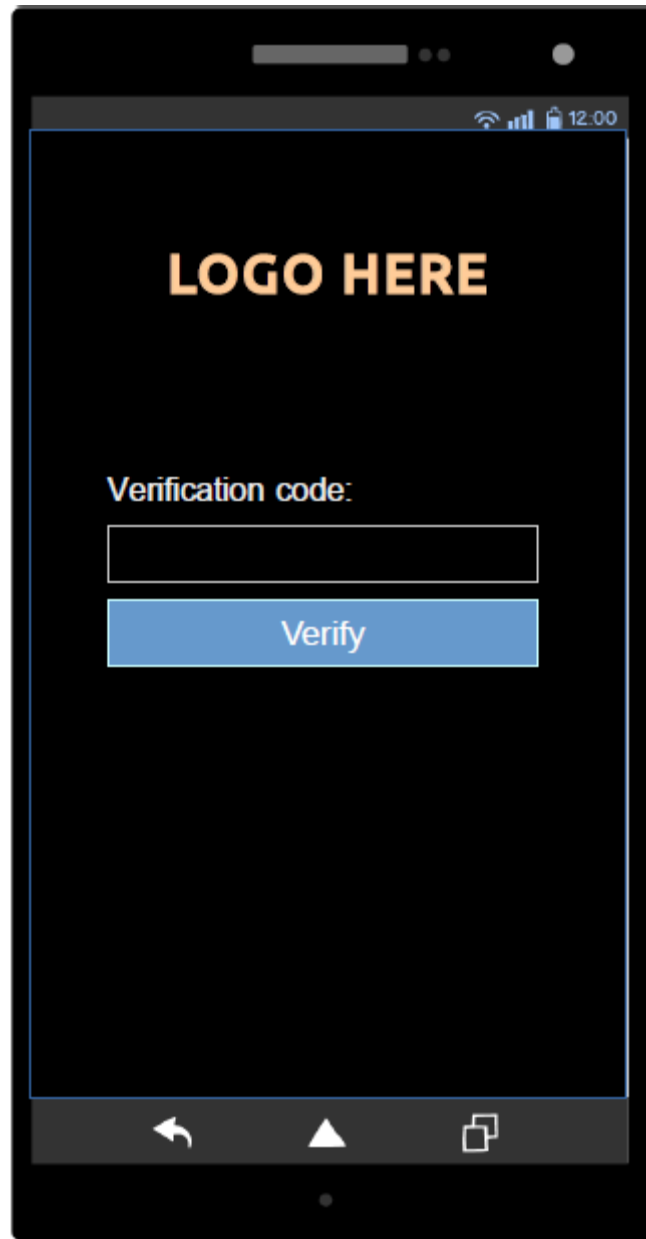


Fig. 4.2: *Verification Screen*

4.3. AUTHENTICATION BLOCK (AB)

This is the third block falling next to the Verification Block where the user is asked to identify himself with the help of his finger print where the user's fingerprint is scanned with the help of Bio Metric Finger Print Scanner and is checked against

the finger print registered on the device and if both of them matches and the result is a success, then the user is being redirected to the next screen which is the Events Block where the list of events for which he has been invited appears. In this screen, the user is given a max time limit of 10s before which he has to place his finger on the scanner, otherwise the user has to follow the steps from the first again. This way we can able to add privacy to the user and thus he feels secure.

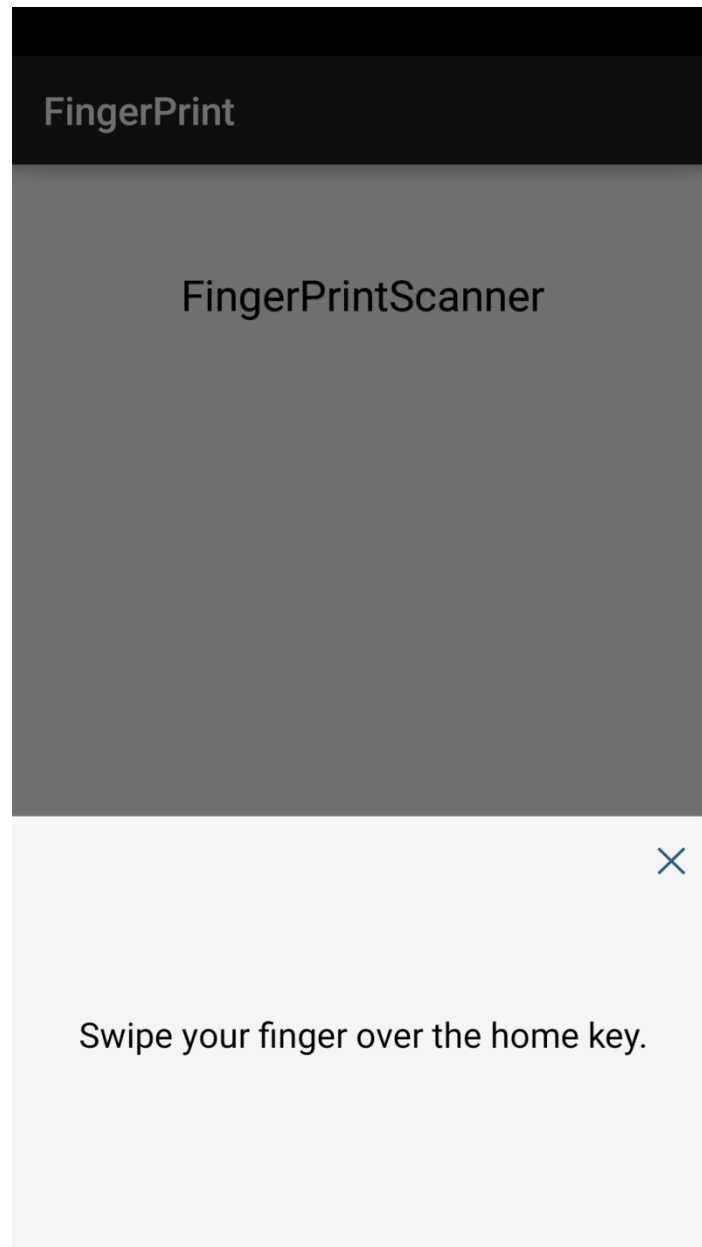


Fig. 4.3: **Finger Print Screen**

4.4. EVENTS BLOCK

This is the last block which is next to the Authentication Block. This is the main or very important block of Evite. This block is basically made up of two sub-blocks. They are:

- Events List (EL)
- Add Events (AE)

4.4.1.Events List:

This is the first sub block of the Events Block. Here, the list of events for which the user has been invited by others will be listed. It is displayed in a list view. We have used the help of a special feature of android which is called as adapters to perform this task. The adapter gets the http response and the sets it to a list datatype. We also have an adapter class which has getter and setter methods each for each variable received as response such as start time, end time, location , contact number of the person invited the user for the event. The getters and setters are used to set the values for the adapter and also we get an automatic scroll bar (vertical) when the list of events exceeded the screen size. The events are listed in this screen and this screen also has a provision for creating a new event and it is denoted with the help of a + symbol. The diagram that shows the Events List screen is as shown below.



Fig. 4.4: *Events List Screen*

4.4.2. Add Events

This is the second sub block of the Events Block. Here, we have actually exploited some of the best available features of android. Here, in this screen the user is asked

to Enter the Name of the event and the optional message if any in the text field provided and the he has to set the date and time of the event with the help of a Date picker and Time picker which is one of the features of android. The user is prompted to select a valid date and time. Secondly we also make use of the google maps api in our application which is one of the highlighting features of this app which prompts the user to set the location of the event by just tapping on the map, the location of the event. Therefore, this Add Events block consists of two sub-blocks. They are:

- Date and Time Picker
- Location Picker
- Contact Picker

4.4.2.1. Date and Time Picker

Initially, there is a text field which shows the text “When” and when the user clicks on this an `onClick`Listener associated with it gets called and the date picker initially pops up on the screen with a dialog interface. The date picker looks similar to a calendar with the current date highlighted on the calendar. The date picker currently is set to the current date on the calendar. The user is asked to pick a date from the forth coming days in the calendar. The user can just simply tap on the screen at the point where the date is present to pick a date from it. After selecting a date he has to click on the set button from the dialog box to set it as the date of the event. Then comes the time picker which automatically pops up on the screen with a dialog interface. The time picker is scrollable and is in a 24 hour format and as in date picker the user has to scroll on to the time of the event and press set and then once this is done, the date and time are displayed on the screen in the place of “When”.

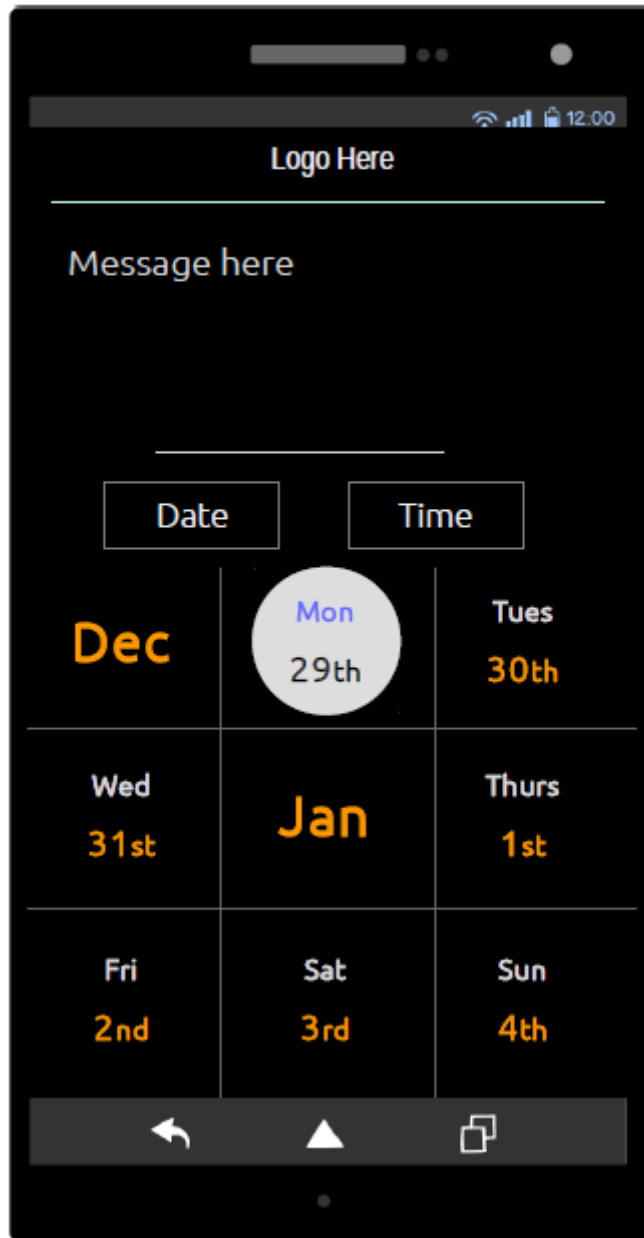


Fig. 4.5: *DATE PICKER*

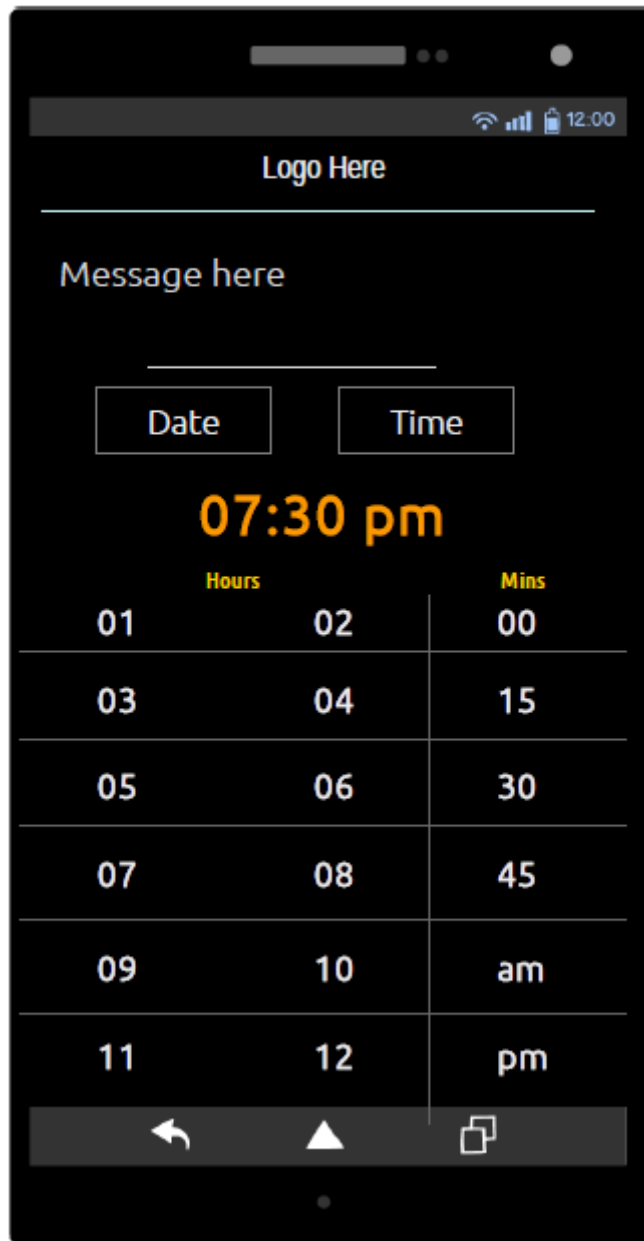


Fig. 4.6: *TIME PICKER*

4.4.2.2. Location Picker:

This is the predominant feature of Evite which helps us to pick the location of the event from the google maps. As, Maps are being used frequently by many to travel to a location or find a locality very easily we provide the convenience to the user of picking the location of the event from the map. Here, we sense the location of the

place the user clicks with the help of an `onClickListener` and get the location of the place with the help of the latitude and longitude of the place the user clicks. Now the latitude and the longitude are displayed on the top of the screen in a textbox which is automatically set with the touch on the map. Also we have used a marker which is displayed on the map where the user clicks to indicate the location of the touch on the map. We have used a geocoder which automatically senses the name of the location and a few other details of the position at which the user clicks on the map. Now, what happens if the user clicks on a wrong place on the map? Nothing to worry, the user can now select the proper location on the map which will place a new marker at that point and remove of the old marker from the point where the user had wrongly placed earlier. Now, the user after fixing the correct location on the map, can press the set button at the bottom of the window which redirects him to the previous page that is the Add Events page and location that he clicked on the map appears at the place of “Where” on the screen.



Fig. 4.7: *LOCTION PICKER*

4.4.2.3. Contact Picker

The contact Picker is the next and final step of Evite where we have to invite the name of the contact we want to invite from the contacts default app which is present in the android device and then we have to just click the button invite contacts to invite them all to the event.

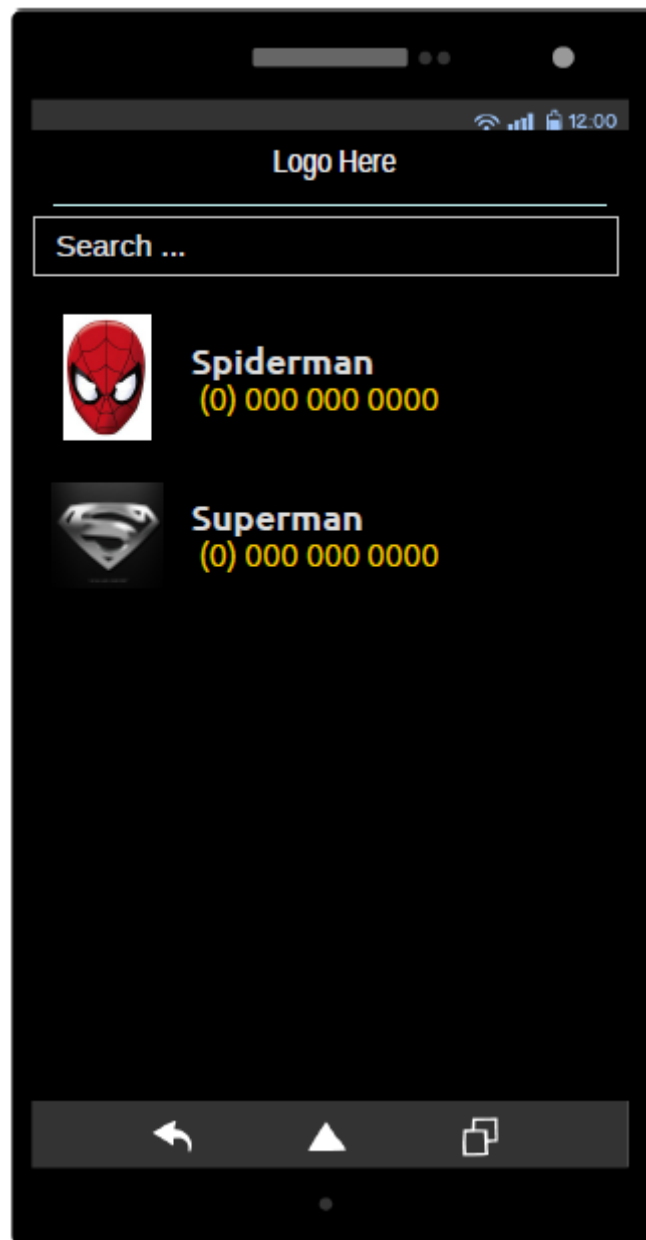


Fig. 4.8: *CONTACT PICKER*

4.5. Event Details Block

This is the final block of Evite. In this, the user can see the details of the event such as the year, month and date of the event, location of the event and other such things. Here there is also a navigate icon on the screen. The user can simply touch the icon to get the exact location of the event on the map. Here, we make use of the latitude and longitude of the location of the event in order to get the marker(a red balloon) on the map. You also get to know where you location on the map is with the help of a small blue colored circle on the map. The image of which is shown below.

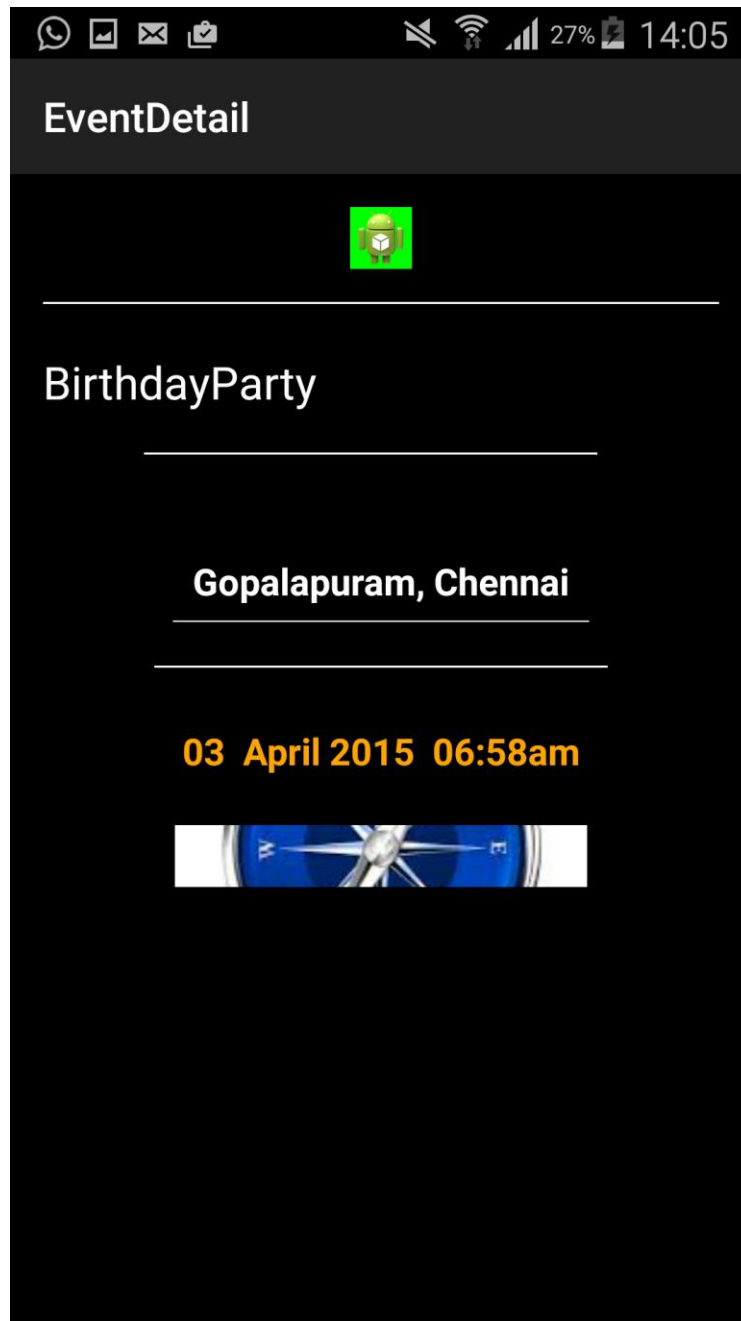


Fig. 4.9: *Event Detail Screen*

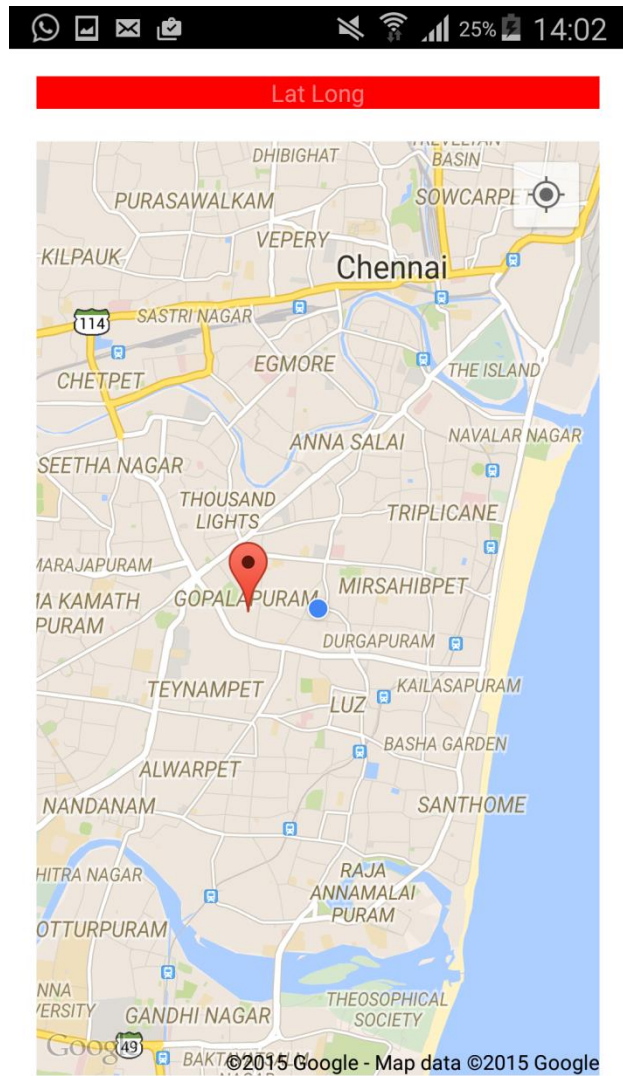


Fig. 4.10: *Event Location Screen*

Chapter 5

Conclusion and Enhancements

Thus, as far as we have seen the Evite for Android is trying to provide customer satisfaction to the utmost extent in all aspects. Now, the user gets a specialized application for sending the invitations on the go to many without the need of many different applications such as what'sapp, google maps, contacts as everything has been merged with Evite itself.

In future, we can make enhancements such as introducing pin unlock for the authentication block which is now in need of a finger print scanner which one must cross in order to use the Evite application. Also, we need to introduce features to invite multiple contacts simultaneously and also bring the contacts list into the Evite so that we don't have the overhead of choosing the contacts from the contacts app just like whatsapp. We also can include features so that the people who are invited have options to accept or decline the invite and thereby the organizer gets notified about the number of people accepted and rejected the invitation.

Now, Evite for ios has been in development stages and will soon be launched to facilitate the users who are having the iphone.

References:

- [1] **Xiping Hu, Xitong Li, Edith C.-H. Ngai, Victor C.M. Leung, and Philippe Kruchten.**, “Multidimensional Context-Aware Social network Architecture for Mobile Crowdsensing”, IEEE Communications Magazine • June 2014
- [2] **Thomas Cormen.**, “Introduction to Algorithms”, 2nd edition , July 14, 1990.
- [3] <http://www.developer.android.com> (Android official site).

Appendices

Android

1. INTRODUCTION

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Android is a software platform and operating system for mobile devices based on the Linux operating system and developed by Google and the Open Handset Alliance. It allows developers to write managed code in a Java-like language that utilizes Google-developed Java libraries, but does not support programs developed in native code.

The unveiling of the Android platform on 5 November 2007 was announced with the founding of the Open Handset Alliance, a consortium of 34 hardware, software and telecom companies devoted to advancing open standards for mobile devices. When released in 2008, most of the Android platform will be made available under the Apache free-software and open-source license.

1.1.THE BIRTH OF ANDROID

1.1.1. Google Acquires Android Inc.

In July 2005, Google acquired Android Inc., a small startup company based in Palo Alto, CA. Android's co-founders who went to work at Google included Andy Rubin (co-founder of Danger), Rich Miner (co-founder of Wildfire Communications, Inc), Nick Sears (once VP at T-Mobile), and Chris White (one of the first engineers at WebTV). At the time, little was known about the functions of Android Inc. other than they made software for mobile phones.

At Google, the team, led by Rubin, developed a Linux-based mobile device OS which they marketed to handset makers and carriers on the premise of providing a flexible, upgradeable system. It was reported that Google had already lined up a series of hardware component and software partners and signaled to carriers that it was open to various degrees of cooperation on their part.

1.1.2. Open Handset Alliance Founded

On 5 November 2007, the Open Handset Alliance, a consortium of several companies which include Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, Sprint Nextel and NVIDIA, was unveiled with the goal to develop open standards for mobile devices. Along with the formation of the Open Handset Alliance, the OHA also unveiled their first product, Android, an open source mobile device platform based on the Linux operating system.

1.1.3. Hardware

Google has unveiled at least three prototypes for Android, at the Mobile World Congress on February 12, 2008. One prototype at the ARM booth displayed several basic Google applications. A 'd-pad' control zooming of items in the dock with a relatively quick response.

A prototype at the Google IO conference on May 28, 2008 had a 528 MHz Qualcomm processor and a Synaptics capacitive touch screen, and used the UMTS cellular standard. It had 128 MB of RAM and 256 MB of flash, showing that Android's memory requirements are reasonable. The demo was carried out using a 3.6 Mbit/s HSDPA connection.

1.2.FEATURES

1.2.1. Application Framework

It is used to write applications for Android. Unlike other embedded mobile environments, Android applications are all equal, for instance, an applications which come with the phone are no different than those that any developer writes. The framework is supported by numerous open source libraries such as openssl, SQLite and libc. It is also supported by the Android core libraries. From the point of security, the framework is based on UNIX file system permissions that assure applications have only those abilities that mobile phone owner gave them at install time.

1.2.2. Dalvik Virtual Machine

It is extremely low-memory based virtual machine, which was designed especially for Android to run on embedded systems and work well in low power situations. It is also tuned to the CPU attributes. The Dalvik VM creates a special file format (.DEX) that is created through build time post processing. Conversion between Java classes and .DEX format is done by included “dx” tool.

1.2.3. Integrated Browser

Google made a right choice on choosing WebKit as open source web browser. They added a two pass layout and frame flattening. Two pass layout loads a page without waiting for blocking elements, such as external CSS or external JavaScript and after a while renders again with all resources downloaded to the device. Frame flattening converts founded

frames into single one and loads into the browser. These features increase speed and usability browsing the internet via mobile phone.

1.2.4. Optimized Graphics

As Android has 2D graphics library and 3D graphics based on OpenGL ES 1.0, possibly we will see great applications like Google Earth and spectacular games like Second Life, which come on Linux version. At this moment, the shooting legendary 3D game Doom was presented using Android on the mobile phone.

1.2.5. SQLite

Extremely small (~500kb) relational database management system, which is integrated in Android. It is based on function calls and single file, where all definitions, tables and data are stored. This simple design is more than suitable for a platform such as Android.

1.2.6. Handset Layouts

The platform is adaptable to both larger, VGA, 2D graphics library, 3D graphics library based on OpenGL ES 1.0 specifications, traditional smart phone layouts. An underlying 2D graphics engine is also included. Surface Manager manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications

1.2.7. Data Storage

SQLite is used for structured data storage .SQLite is a powerful and lightweight relational database engine available to all applications.

1.2.8. Connectivity

Android supports a wide variety of connectivity technologies including GSM, CDMA, Bluetooth, EDGE, EVDO, 3G and Wi-Fi.

1.2.9. Messaging

SMS, MMS, and XMPP are available forms of messaging including threaded text messaging.

1.2.10. Web Browser

The web browser available in Android is based on the open-source WebKit application framework. It includes LibWebCore which is a modern web browser engine which powers both the Android browser and an embeddable web view.

1.2.11. Java Virtual Machine

Software written in Java can be compiled into Dalvik bytecodes and executed in the Dalvik virtual machine, which is a specialized VM implementation designed for mobile device use, although not technically a standard Java Virtual Machine.

1.2.12. Media Support

Android will support advanced audio/video/still media formats such as MPEG-4, H.264, MP3, and AAC, AMR, JPEG, PNG, GIF.

1.2.13. Additional Hardware Support

Android is fully capable of utilizing video/still cameras, touchscreens, GPS, compasses, accelerometers, and accelerated 3D graphics.

1.2.14. Development Environment

Includes a device emulator, tools for debugging, memory and performance profiling, a plugin for the Eclipse IDE. There are a number of hardware dependent features, for instance, a huge media and connections support, GPS, improved support for Camera and simply GSM telephony. A great work was done for the developers to start work with Android using device emulator, tools for debugging and plugin for Eclipse IDE.