Message Scheduler

A Mini-Project Report
Under
Project Workshop

Submitted by

Aman Agarwal (B004)
Tanmay Agnani (B005)
Neelam Babel (B010)
Surbhi Chaplot (B016)

Under The Guidance Of

Prof. Abhay Kolhe

In partial fulfillment for the award of the degree
Of

Bachelor of Technology
IN
Computer Engineering

At
NMIMS' MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT AND ENGINEERING, MUMBAI

April 2014

CERTIFICATE

This is to certify that the project entitled “Message Scheduler” is the bonafide work carried out by Aman Agrawal, Tanmay Agnani, Neelam Babel, Surbhi Chaplot of B.Tech(Computer Engineering), MPSTME (NMIMS), Mumbai, during the VI semester of the academic year 2013-14, in partial fulfillment of the requirements for the award of the Degree of Bachelors of Technology as per the norms prescribed by NMIMS. The mini-project work has been assessed and found to be satisfactory.

Prof. Abhay Kolhe

Internal Mentor

Examiner 1

Examiner 2

Dean

Dr. S. Y. Mhaiskar

DECLARATION

We, Aman Agrawal (B004), Tanmay Agnani (B005), Neelam Babel (B010), Surbhi Chaplot (B016) B.Tech (Computer Engineering), VI semester understand that plagiarism is defined as anyone or combination of the following:

1. Un-credited verbatim copying of individual sentences, paragraphs or illustration (such as graphs, diagrams, etc.) from any source, published or unpublished, including the internet.
2. Un-credited improper paraphrasing of pages paragraphs (changing a few words phrases, or rearranging the original sentence order)
3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did wrote what. (Source:IEEE, The institute, Dec. 2004)
4. I have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of my work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.
5. I affirm that no portion of my work can be considered as plagiarism and I take full responsibility if such a complaint occurs. I understand fully well that the guide of the seminar/ project report may not be in a position to check for the possibility of such incidences of plagiarism in this body of work.

Signature of the Student	Signature of the Student	Signature of the Student	Signature of the Student
Aman Agrawal Roll No. B004 Place: Mumbai Date:	Tanmay Agnani Roll No. B005 Place: Mumbai Date:	Neelam Babel Roll No. B010 Place: Mumbai Date:	Surbhi Chaplot Roll No. B016 Place: Mumbai Date:

Table of contents

List of Figures

Abstract

Our project is an Android Application called Message Scheduler. This application will automate the messages with chosen frequency on user’s behalf. The application will allow the user to create new text messages and specify when to send them. The user just needs to specify the message, contact, and his date and time requirements and then simply forget about it. Also the messages can be set to repeat. Message Scheduler shall provide the user the ability to add or view any event while allowing a concise and structured way of selecting the event information. The app has been set up in a way that makes it user-friendly and approachable to a larger audience.

The app helps the user by reducing his job of memorizing the dates of the important events by automating the messages when a particular event is triggered. In future, the app can be made a free app by providing free messaging.

1. INTRODUCTION

1.1 Project Overview

This software system will be an Android Application that allows the users to schedule messages to be sent in future. The user just needs to specify the message, contact, and his date and time requirements and then simply forget about it. Also the messages can be set to repeat. Message Scheduler shall provide the user the ability to add or view any event while allowing a concise and structured way of selecting the event information. The app has been set up in a way that makes it user-friendly and approachable to a larger audience.
This is a relatively new project. There are a few simple scheduling applications available on the android market but most of them are not user-friendly and do not allow recurring of messages. Our aim is to create a free and useful app. It will be available for everyone with Android 3.0 (Honeycomb) or higher; with all basic requirements fulfilled by every mobile phone (RAM, memory, etc.)
This is a very small application that takes less than 1.50 Megabytes when installed on the device. Also it is not a processor intensive application and thus can be used by any person with any android device no matter what its hardware specifications as long as it is running on Android 3.0 (Honeycomb) or higher.
There will most probably be no updates available to the App unless some bugs are brought to notice.

1.2 Hardware Requirements

Recommend 2GB memory for IDE and 2GB if running server locally

1.3 Software Requirements

Operating Systems

- Windows XP (32-bit), Vista (32- or 64-bit), or Windows 7 (32- or 64-bit)
- Mac OS X 10.5.8 or later (x86 only)
- Linux (tested on Ubuntu Linux, Lucid Lynx)
- GNU C Library (glibc) 2.7 or later is required.
- On Ubuntu Linux, version 8.04 or later is required.
- 64-bit distributions must be capable of running 32-bit applications.

Eclipse IDE

- Eclipse 3.7.2 (Indigo) or greater

Note: Eclipse 3.6 (Helios) is no longer supported with the latest version of ADT.

- Eclipse JDT plugin (included in most Eclipse IDE packages)
- JDK 6 (JRE alone is not sufficient)
- Android Development Tools plugin (recommended)
- Not compatible with GNU Compiler for Java (gcj)

Other development environments

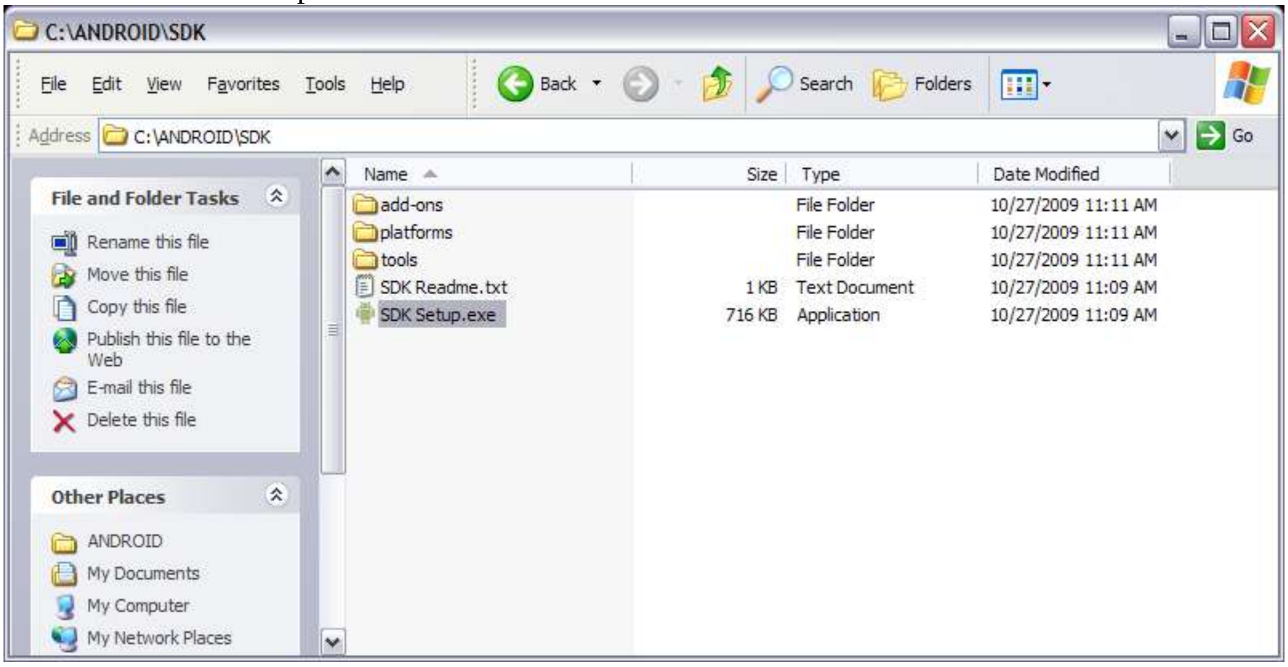
- JDK 6 (JRE alone is not sufficient)
- Apache Ant 1.8 or later
- Not compatible with Gnu Compiler for Java (gcj)

Note: Some Linux distributions may include JDK 1.4 or Gnu Compiler for Java, both of which are not supported for Android development.

2. INSTALLATION AND SETUP

2.1 Installing ADT Bundle (Android SDK + Eclipse IDE)

First run the SDK Setup File

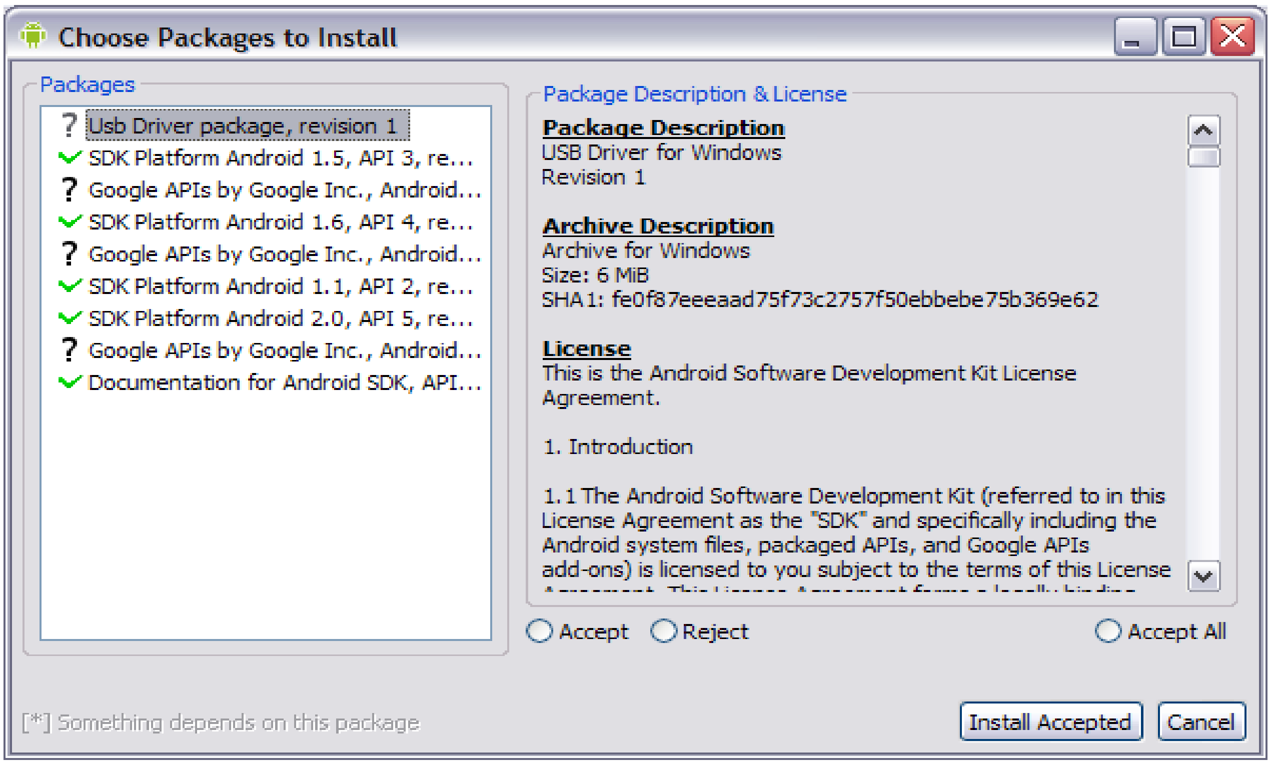


Address: C:\ANDROID\SDK

Name	Size	Type	Date Modified
add-ons		File Folder	10/27/2009 11:11 AM
platforms		File Folder	10/27/2009 11:11 AM
tools		File Folder	10/27/2009 11:11 AM
SDK Readme.txt	1 KB	Text Document	10/27/2009 11:09 AM
SDK Setup.exe	716 KB	Application	10/27/2009 11:09 AM

Figure 2.1.1: Directory

Then select the packages that you wish to install. We installed all packages


Choose Packages to Install
Packages
Package Description & License
Accept
Reject
Accept All
[*] Something depends on this package
Install Accepted
Cancel

- ? Usb Driver package, revision 1
- ✓ SDK Platform Android 1.5, API 3, re...
- ? Google APIs by Google Inc., Android...
- ✓ SDK Platform Android 1.6, API 4, re...
- ? Google APIs by Google Inc., Android...
- ✓ SDK Platform Android 1.1, API 2, re...
- ✓ SDK Platform Android 2.0, API 5, re...
- ? Google APIs by Google Inc., Android...
- ✓ Documentation for Android SDK, API...

Package Description
USB Driver for Windows
Revision 1

Archive Description
Archive for Windows
Size: 6 MiB
SHA1: fe0f87eeeaad75f73c2757f50ebbebe75b369e62

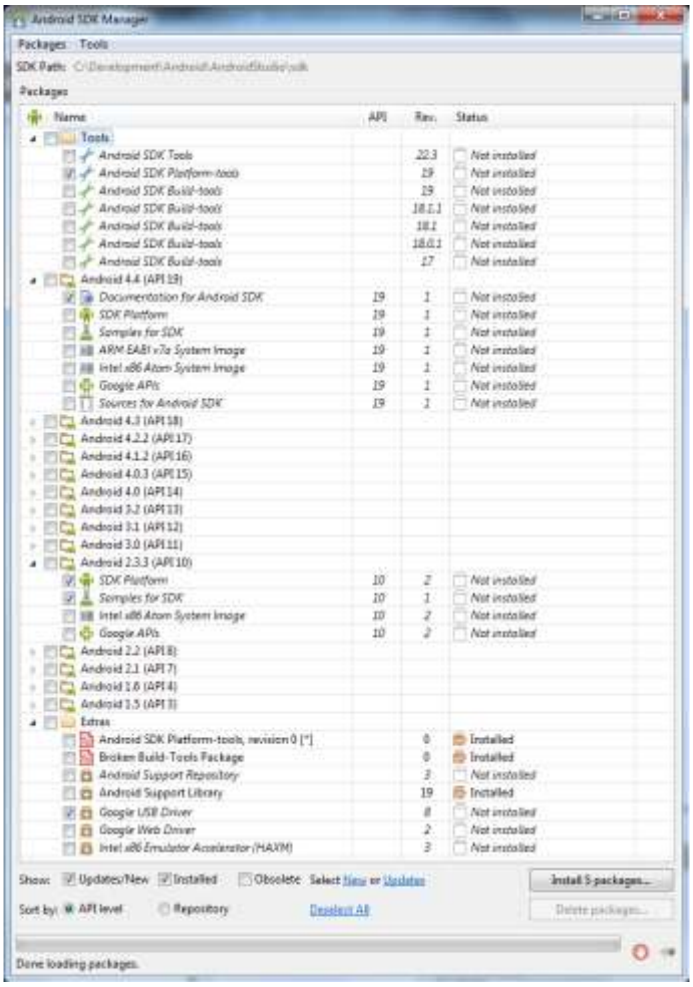
License
This is the Android Software Development Kit License Agreement.

1. Introduction

1.1 The Android Software Development Kit (referred to in this License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of this License

Figure 2.1.2 : Window displaying the packages to be installed

Then open the Android SDK Manager and install the packages that you require (like Android Documentation, Samples for various android versions, SDK platform, etc.)


Android SDK Manager

Packages Tools

SDK Path: C:\Development\Android\AndroidStudio\sdk

Packages

Name	API	Rev.	Status
Tools			
Android SDK Tools		22.3	Not installed
✓ Android SDK Platform-tools		19	Not installed
Android SDK Build-tools		19	Not installed
Android SDK Build-tools		18.1.1	Not installed
Android SDK Build-tools		18.1	Not installed
Android SDK Build-tools		18.0.1	Not installed
Android SDK Build-tools		17	Not installed
Android 4.4 (API 19)			
✓ Documentation for Android SDK	19	1	Not installed
SDK Platform	19	1	Not installed
Samples for SDK	19	1	Not installed
ARM EABI v7a System Image	19	1	Not installed
Intel x86 Atom System Image	19	1	Not installed
Google APIs	19	1	Not installed
Sources for Android SDK	19	1	Not installed
Android 4.3 (API 18)			
Android 4.2.2 (API 17)			
Android 4.1.2 (API 16)			
Android 4.0.3 (API 15)			
Android 4.0 (API 14)			
Android 3.2 (API 13)			
Android 3.1 (API 12)			
Android 3.0 (API 11)			
Android 2.3.3 (API 10)			
✓ SDK Platform	10	2	Not installed
✓ Samples for SDK	10	1	Not installed
Intel x86 Atom System Image	10	2	Not installed
Google APIs	10	2	Not installed
Android 2.2 (API 8)			
Android 2.1 (API 7)			
Android 1.6 (API 4)			
Android 1.5 (API 3)			
Extras			
Android SDK Platform-tools, revision 0 [*]		0	Installed
Broken Build-Tools Package		0	Installed
Android Support Repository		3	Not installed
Android Support Library		19	Installed
✓ Google USB Driver		8	Not installed
Google Web Driver		2	Not installed
Intel x86 Emulator Accelerator (HAXM)		3	Not installed

Show: ✓ Updates/New ✓ Installed ☐ Obsolete Select New or Updates

Sort by: ● API level ○ Repository Deselect All

Done loading packages.

Figure 2.1.3: Android SDK manager

After this open Eclipse (it will have automatically been installed when you unpack the ADT bundle) and it will automatically load in the ADT mode.

2.2 Set-up (Running Test App)

Steps of creating applications

- Start Eclipse and select your workspace

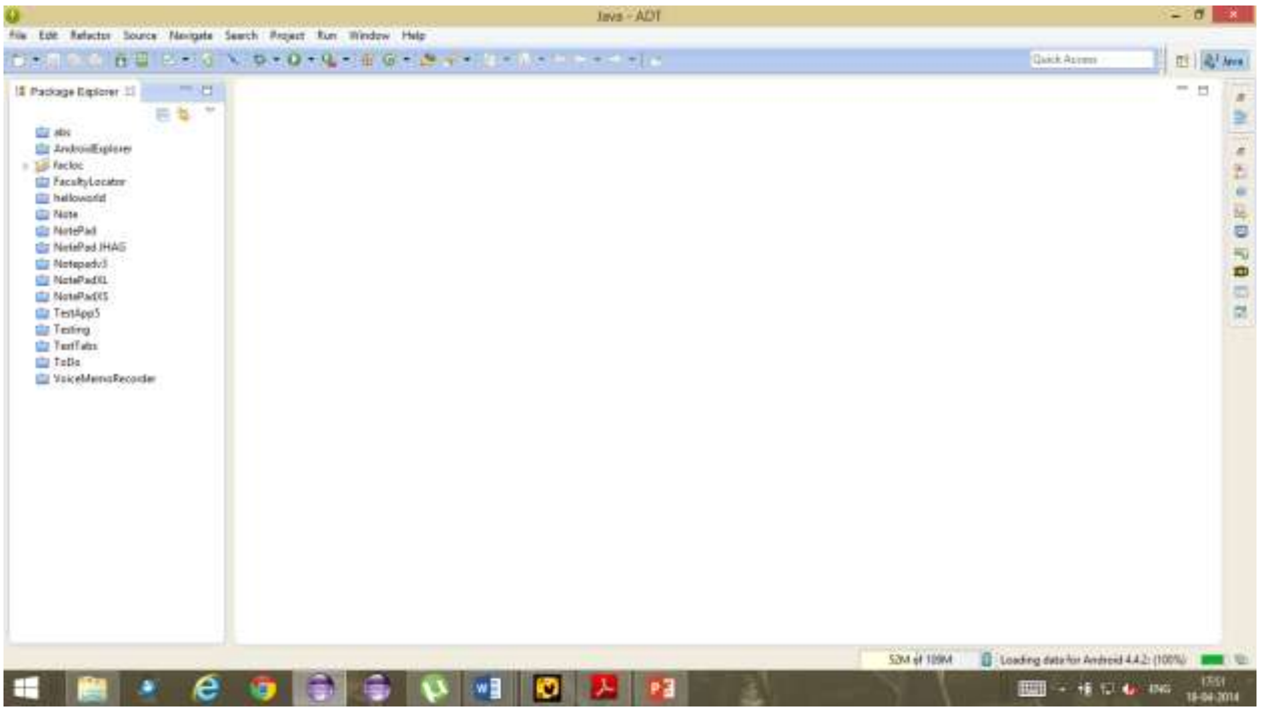
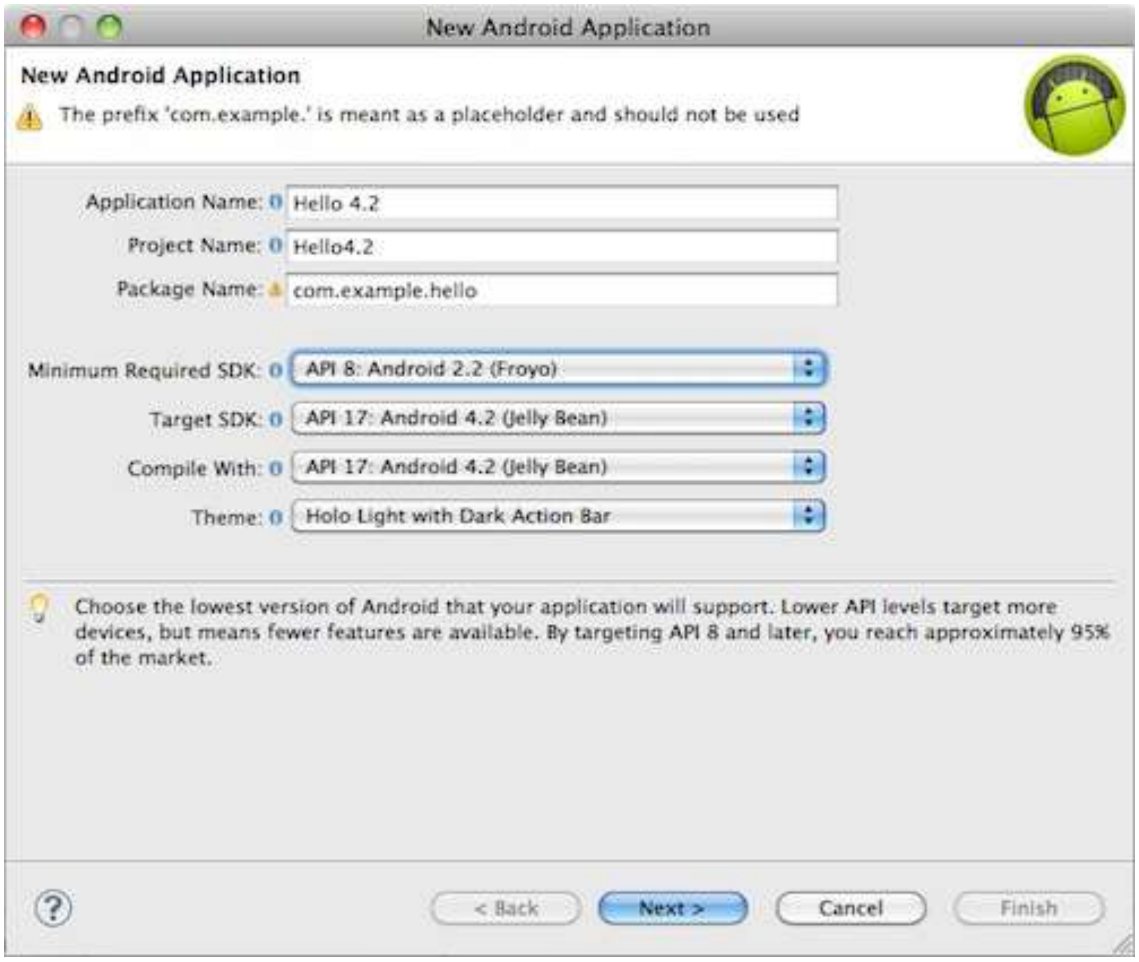


Figure 2.2.1: Java-ADT Window

- Then, go to File -> New -> Android Application Project. Name your Application and select the version like this:


New Android Application

New Android Application

The prefix 'com.example.' is meant as a placeholder and should not be used

Application Name: Hello 4.2

Project Name: Hello4.2

Package Name: com.example.hello

Minimum Required SDK: API 8: Android 2.2 (Froyo)

Target SDK: API 17: Android 4.2 (Jelly Bean)

Compile With: API 17: Android 4.2 (Jelly Bean)

Theme: Holo Light with Dark Action Bar

Choose the lowest version of Android that your application will support. Lower API levels target more devices, but means fewer features are available. By targeting API 8 and later, you reach approximately 95% of the market.

Figure 2.2.2: New Android Application Window

- By default there will be a program to print hello world. We can run it on either a USB connected Android Device or the AVM emulator.
- The Emulator home screen looks like this:

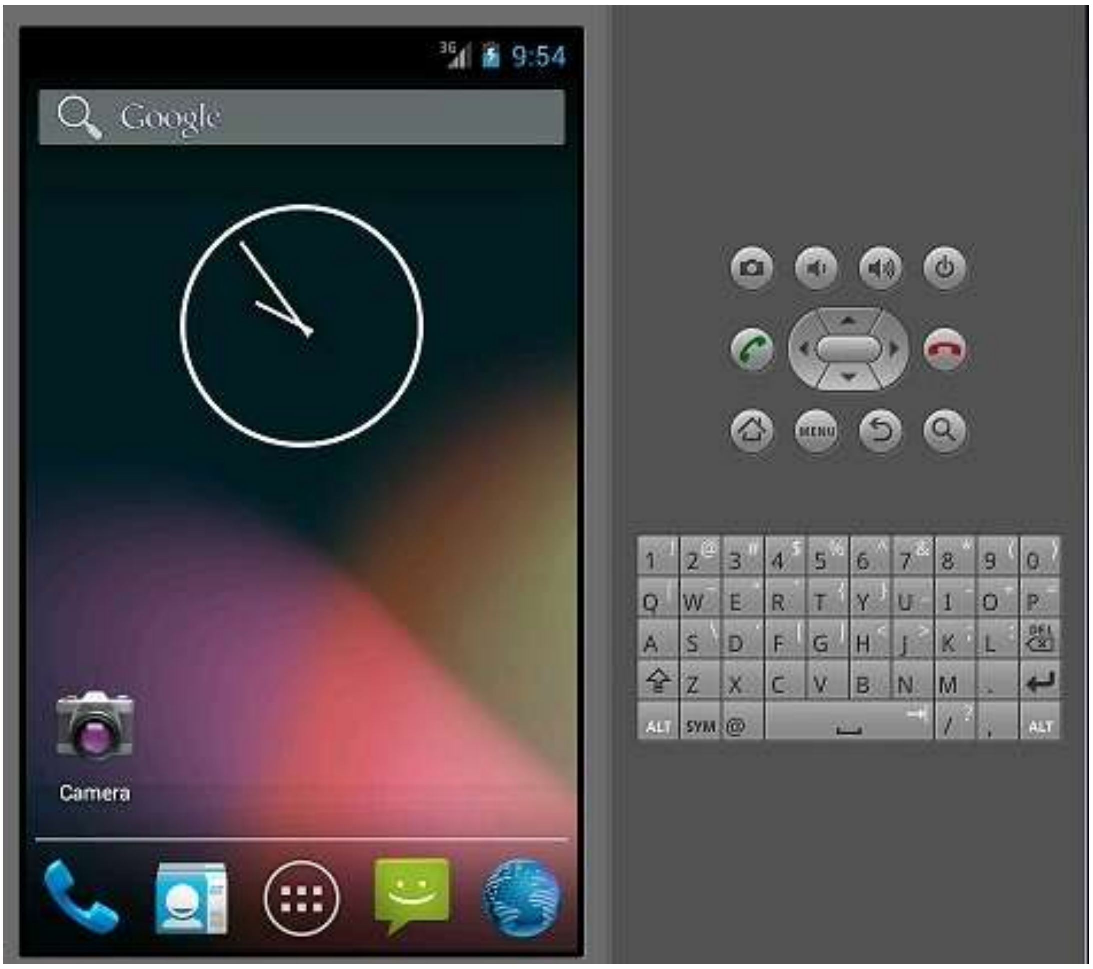


Figure 2.2.3: Emulator

- When we run the Hello World program on it, it looks like:

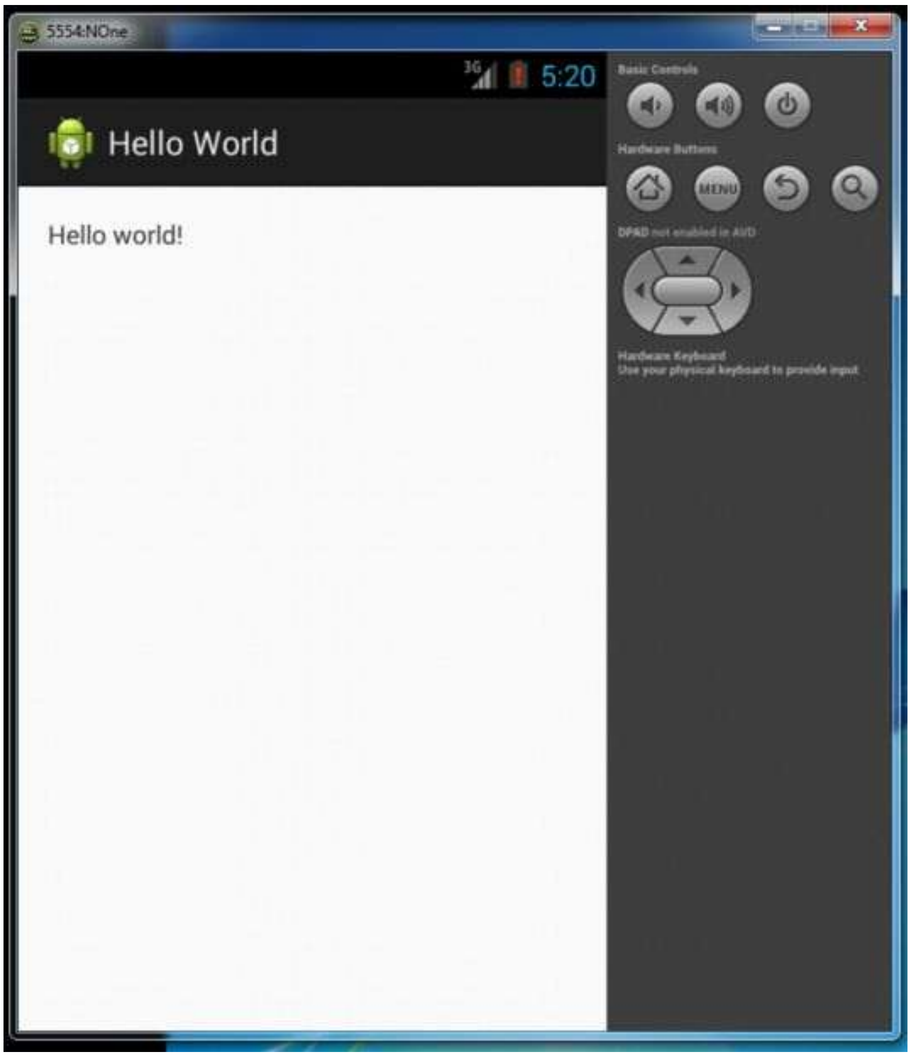


Figure 2.2.4: Emulator displaying the running application

3. ANALYSIS & DESIGN

3.1 ANALYSIS

3.1.1 External Interface Requirements

Contacts:
The user should be able to access the contact list of the phone book.

Notification:
The application should be allowed to broadcast the notification.

Time:
The application should be able to receive the time from the system.

Date
The application should be able to receive the date from the system.

3.1.2 Hardware Interfaces

Since the app is an Android app, Android hardware will be a requirement to run the app.

For example any Android device like Aakash Tablet, Nexus 5, etc. The hardware should also be touch enabled.

3.1.3 Software Interfaces

Message Scheduler runs on every android phone available so it does not need any specific software from the user's side. However, Message Scheduler depends on some libraries and tools for its function. These packages are already pre-included in the .apk file downloaded.

3.1.4 Communications Interface

Internet connection is not required for the working for the application. User needs to have a minimum balance in order to send the message.

3.1.5 Functional Requirements

Time:
Trigger - Touching the "Time Picker" Component.
Function – User has to select the time at which he wants to send the message.
Error Handling – None.

Date:
Trigger - Touching the "Date Picker" Component.
Function – User has to select the day on which he wants to send the message.
Error Handling – None.

Number:
Trigger – Text box
Function – Will take the phone no. on which the user wants to send the message.
Error Handling – The user needs to enter the correct phone number.

To who?:
Trigger - Touching the “To Who?” Button.
Function – Asks the user, contact on which he wants to send the message from his phone directory in case he doesn’t want to enter it manually.
Error Handling – The no. need to be stored on the directory and must be a valid no.

Message:
Trigger – Text Box.
Function – Will take the message from the user regarding what to send.
Error Handling – No special error handling is applied on this textbox.

3.1.6 Use Cases

Use Case #1

Add event:

Purpose: Add event function allows the user to add a new event.

Steps:

- The user opens the app.
- The user opens the Home page.
- The user enters the required event information.
- The user clicks the “Send” button.
- The user opens the app.
- The user views the event by clicking on the queue option in the action bar and selects the event information.

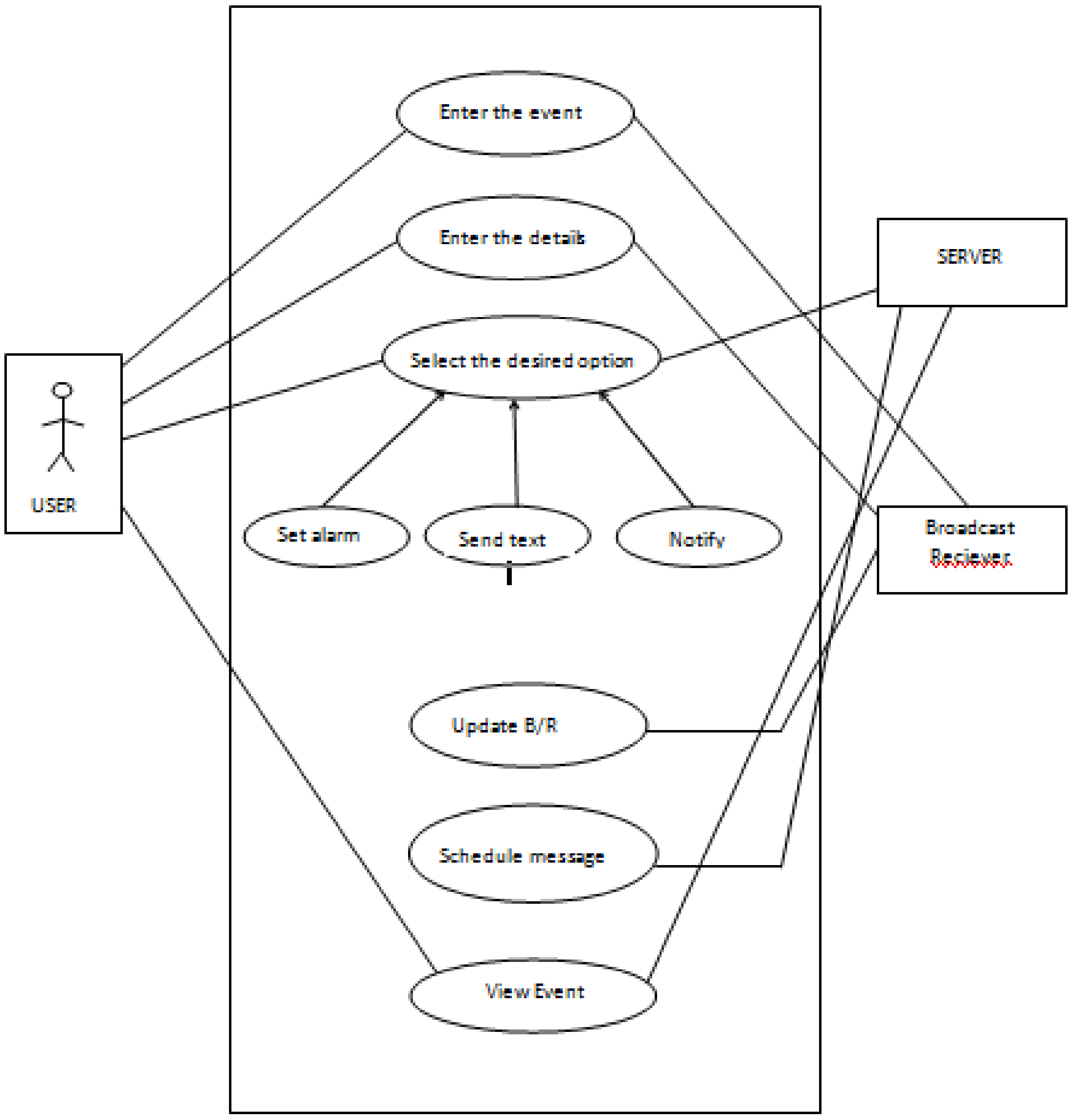


Figure 3.1.1: Use case Diagram

3.1.7 Non-Functional Requirements

Performance

The application will have a good performance on both high end as well as low end android smartphones because the app is not very resource intensive.

Reliability

The application will have the functionality to create a backup of the database from time to time and hence increasing the reliability of the data stored.

Availability

The application will be available in the form of ".apk" initially and once approved will be available on the Google Play Store once approved by Google.

Security

Security will be an issue because of the possibility of personal information of the clients stored. Because of these issues the application will have various security features to enable to user to know that all action performed on the application will be safe.

Maintainability

Continuous application updates and maintenance releases will be offered to keep the application running smoothly over time.

Portability

The application will of course be portable because it is a mobile application.

3.1.8 Design Constraints

- The Android design guide was followed to make use of the recommended design language.
- Software interfaces involved in designing the application was constrained to Android SDK.
- The program size can't be very large as it is a mobile application which have a limited amount of storage space.
- The hardware has to be an android phone with touch screen capabilities.

3.1.9 Other Requirements

- The user does not need any special training in using the application as it's a very simple and easy-to-use application.
- When it comes to reusability the source code will be open source so anyone can work on it and improve it.

\

Analysis Models

1. Sequence Diagram

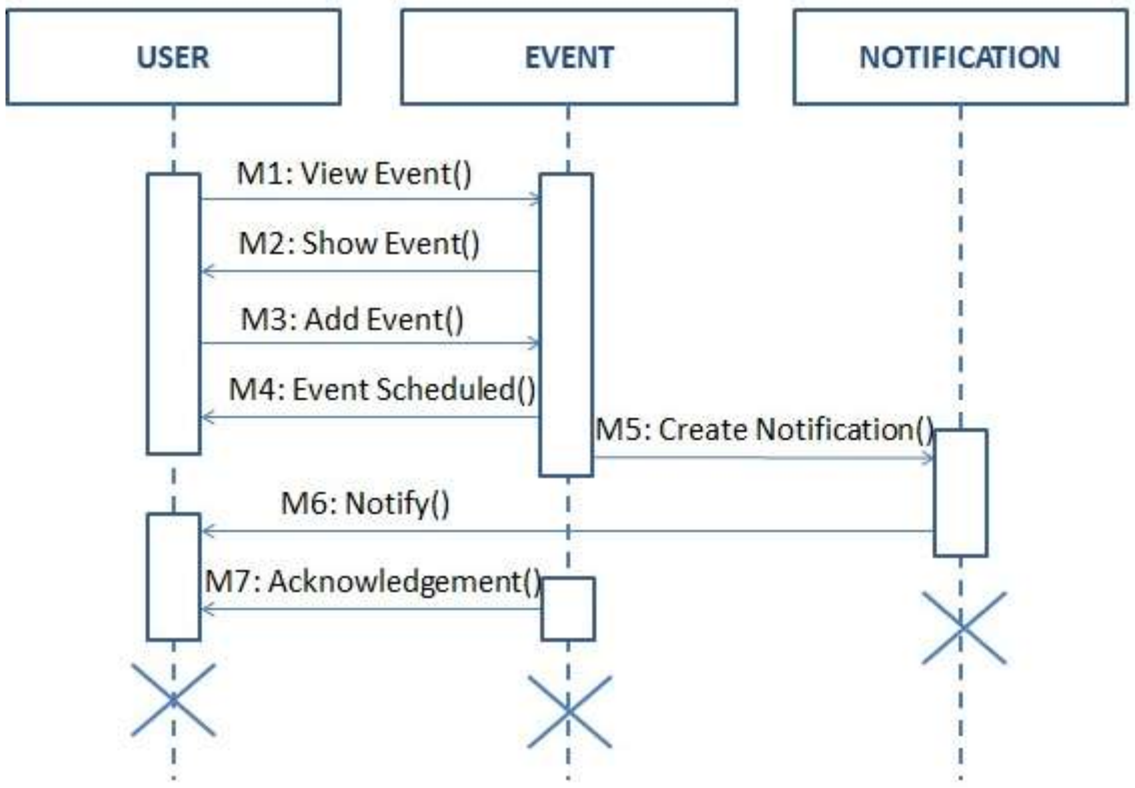


Figure 3.1.2: Sequence Diagram

2. Data Flow Diagram (DFD)

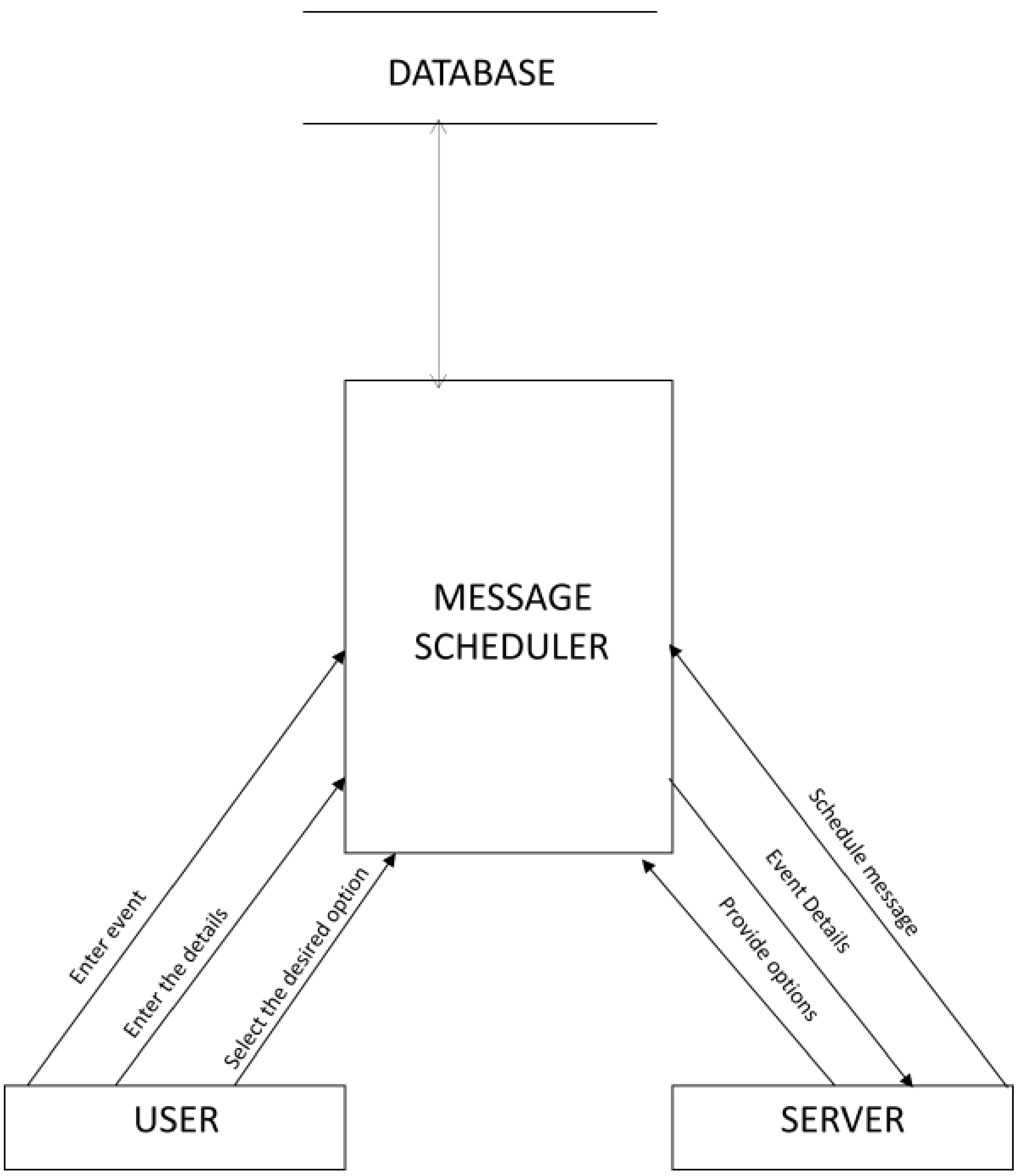


Figure 3.1.3: Data Flow Diagram

4. PROJECT IN DETAIL

Here we will describe the working of the project including the working of the code given above in detail

4.1 Splash Screen

A splash screen is an image that appears while a game or program is loading. The term may also be used to describe an introduction page on a website or an application. Splash screens cover the entire screen or simply a rectangle near the center of the screen. Our slash screen is displayed for 5 seconds. For the app to display the splash screen, we declare the SplashScreen activity with intent.action.MAIN to that it is opened first in AndroidManifest.xml.

Once the timer expires we will be redirected to the Main Activity

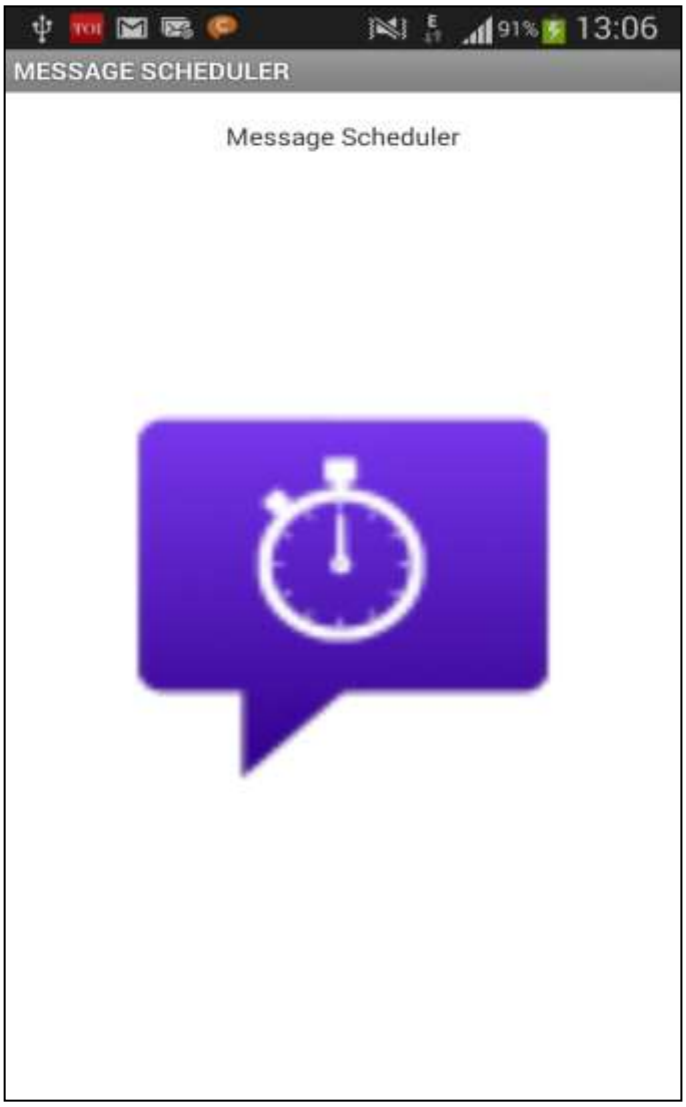


Figure 4.1: Splash screen

4.2 Main Layout

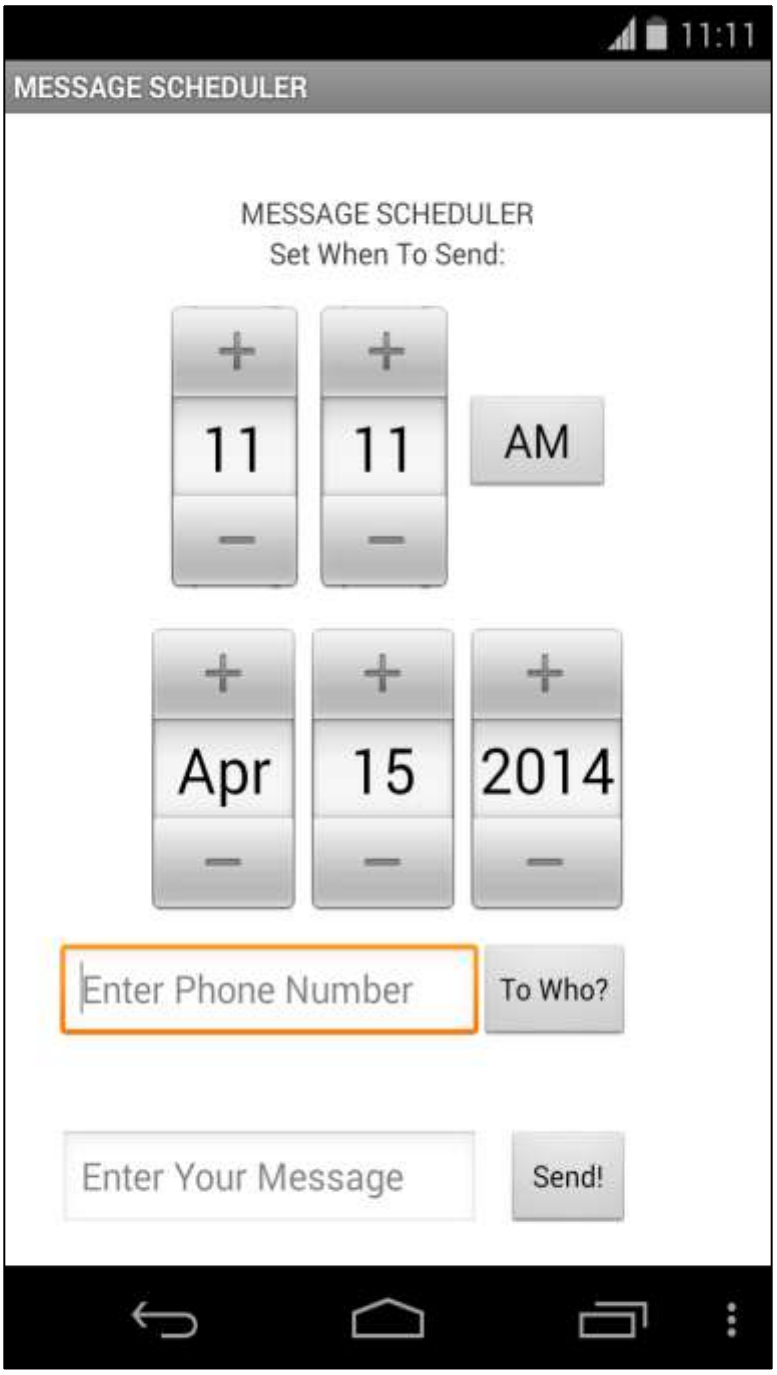


Figure 4.2: Main activity

This is the main activity that is the home screen of outr application Message Scheduler wherein the user is provided with various options to schedule the message which are as follows:

Time Picker – User has to select the time at which he wants to send the message.

Date Picker – User has to select the day on which he wants to send the message.

Number: – Will take the phone no. on which the user wants to send the message.

Contact Select – Asks the user, contact on which he wants to send the message from his phone directory in case he doesn’t want to enter it manually.

Message: Will take the message from the user regarding what to send.

4.3 Options Available

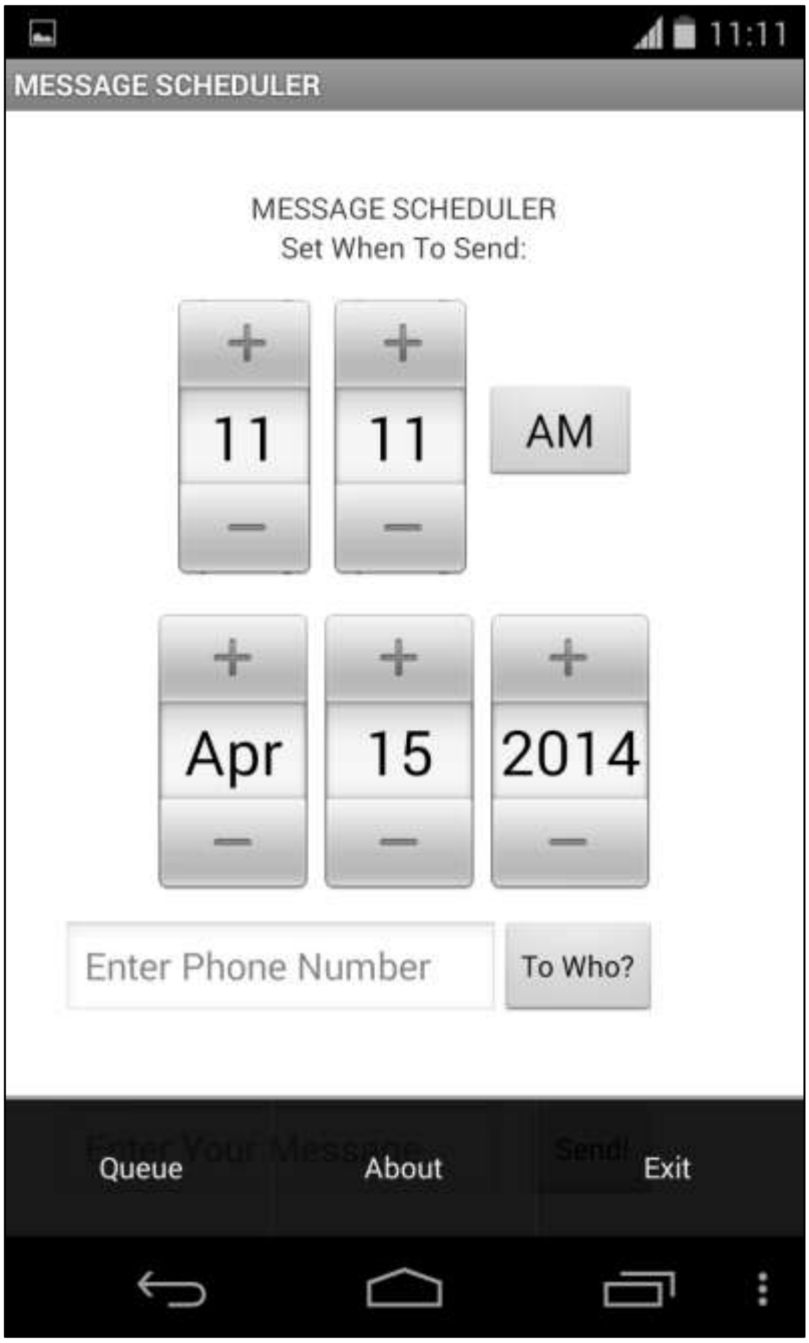
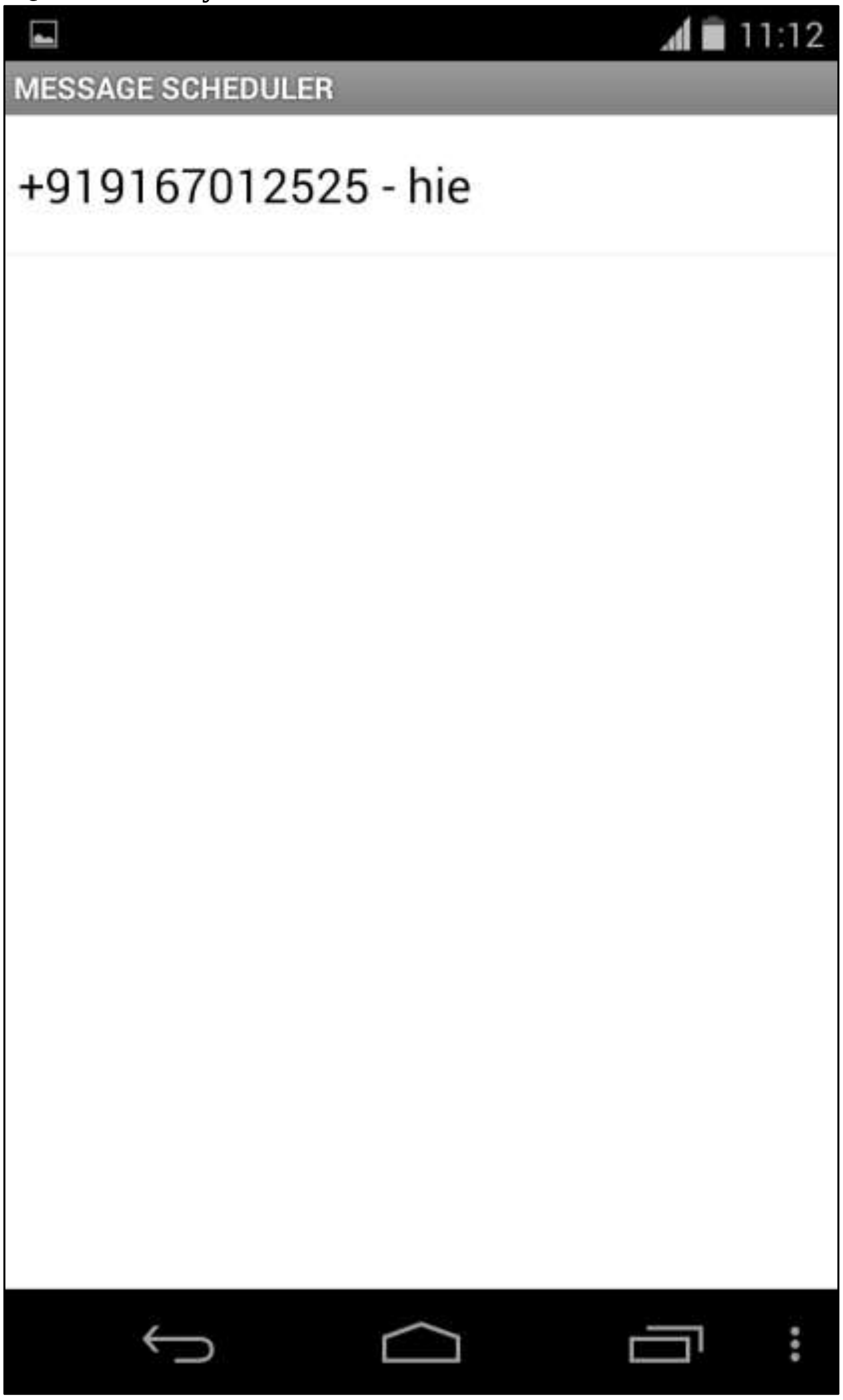


Figure 4.3: main Activity with action bar

It has three options:

- Queue: Displays the list of all scheduled messages.
- About: Tells about the app and the developers.
- Exit: To exit to the main screen.

4.4 Queue Activity

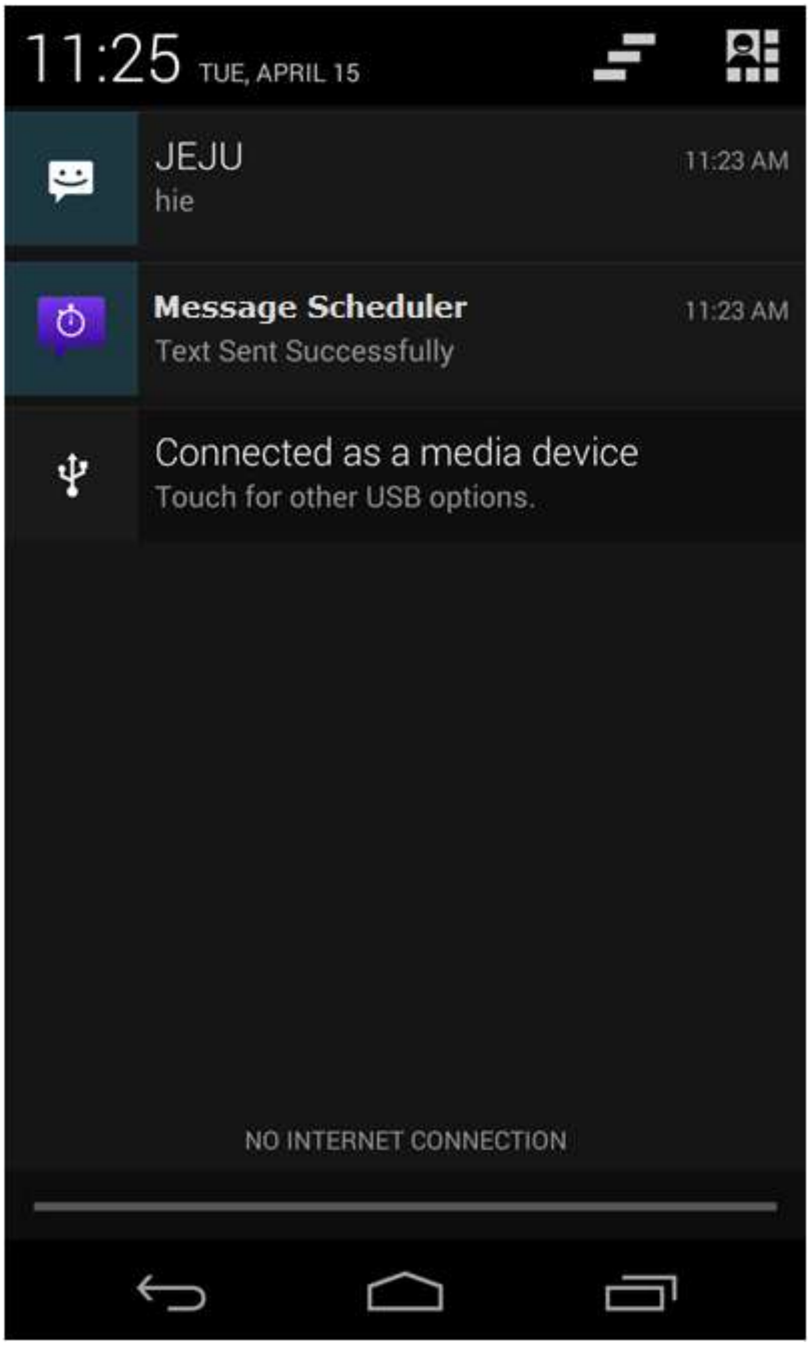


+919167012525 - hie

Figure 4.4: Queue activity

This is the screen having a list of all the message scheduled.
It displays the no. and the text of the message.
Entry from the list is removed when the message is delivered.

4.5 Notifications


11:25 TUE, APRIL 15

JEJU 11:23 AM
hie

Message Scheduler 11:23 AM
Text Sent Successfully

Connected as a media device
Touch for other USB options.

NO INTERNET CONNECTION

Figure 4.5: Notification screen

The screenshot displays the notification received to the user when the task is completed.

5. CONCLUSION & FUTURE SCOPE

5.1 CONCLUSION

Thus we have made an Android Application, a Message Scheduler which can be used to schedule messages based on user’s date and time requirements. The user can add a new event and can also view the existing events.

The app gives a very user-friendly interface where the user can choose the event and contact, followed by date and time details. Then the rest of the work will be done by the app and the message will be sent to the specified person on due date which can be confirmed via push notifications.

5.2 FUTURE SCOPE

This is not the final polished Application. There are a number of things that can be added to better the experience of using this App. Some of them are:

- The ability to send message for free.
- The ability to send animation greetings.
- The ability to send message to multiple recipients.

Other amazing features as and when suggested will also be incorporated to the best of our abilities in the Application.

REFERENCES

[1]. http://developer.android.com/training/index.html

[2]. http://www.stackoverflow.com

[3]. http://www.thenewboston.com/videos

[4]. http://www.lynda.com