

TASK SCHEDULER APPLICATION ON ANDROID

A Thesis

Presented to the

Faculty of

San Diego State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

by

Naga Santhoshi Bokka

Spring 2012

SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Naga Santhoshi Bokka:

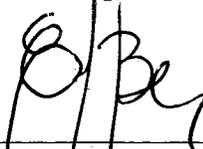
Task Scheduler Application on Android



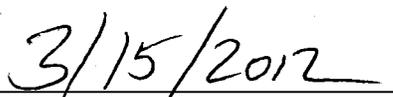
Joseph Lewis, Chair
Department of Computer Science



Carl Eckberg
Department of Computer Science



Carlos Baran
Department of Computational Science



Approval Date

Copyright © 2012
by
Naga Santhoshi Bokka
All Rights Reserved

ABSTRACT OF THE THESIS

Task Scheduler Application on Android

by

Naga Santhoshi Bokka

Master of Science in Computer Science

San Diego State University, 2012

Smartphones has become an important part of modern life. In day to day life people often use and carry them where ever they we go. So getting reminders for the important tasks you wish to complete through mobile devices is an efficient and effective way. With advancement of technology, smartphones today have many capabilities to provide user interfaces with its copious features.

The task scheduler application done for this thesis is an android based application for students and business people to schedule and full fill prioritized tasks in their daily life and get email notifications to provide quick alerts to users. The application helps to create and schedule tasks automatically based upon date and time, view a list of tasks, edit and delete tasks whenever we want. Tasks which are scheduled will be stored in the application calendar and the application provides a quick view of tasks. The task scheduler application stores all the scheduled tasks in the application calendar and gives email reminders in the Gmail, Yahoo mail and corporate email account that can be set up on the phone and as well as alarm reminders. In this application, when the user selects gmail account as sender and type any recipient email address as recipient, it will send the email notifications to the particular recipient email address and at some later time notifies user through the alarm notification on the phone. This application will send email notifications to the preferred user from the host user which will act like an invitation for any business meeting. In summary, the application helps reminds the user of the task schedule he makes and the priorities he gives to the tasks. The applicaton can send email notifications or invitations to the arbitrary recipients.

This application is developed to provide a many featured interface, easy to access and user friendly application. The application notifies the user through both alarm and email notifications to complete the tasks as per priority and helps for better time management. The application is developed on Android OS and provides flexibility and user friendly feature. The smartphones used to develop and test the project are built of Android platform.

TABLE OF CONTENTS

	PAGE
ABSTRACT.....	iv
LIST OF TABLES.....	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS.....	xi
CHAPTER	
1 INTRODUCTION	1
1.1 Emerging Mobile Platforms.....	1
1.2 Mobile Operating System	2
1.3 Existing ToDo Android Applications	3
1.3.1 ToDo Application	3
1.3.2 Brief Discussion About an Existing App (Quick List)	3
1.4 My Application.....	5
1.5 Research and Study.....	6
1.6 Research Objective	7
2 MOBILE OPERATING SYSTEMS.....	10
2.1 Mobile Operating Systems Overview	10
2.1.1 Iphone OS	10
2.1.2 Windows Mobile.....	11
2.1.3 Symbian OS	11
2.1.4 Android OS	12
2.2 Why is Android More Popular Than Other Smartphones?.....	13
2.3 Smartphone Market Share.....	14
2.4 Application Resources for Various Smartphones	16
2.5 Android Mobile Platform.....	16
2.6 Introduction to Android	17
2.7 Android Platform Features.....	17
2.8 Android Architecture	18

2.8.1 Applications	19
2.8.2 Application Framework	19
2.8.3 Libraries	20
2.8.4 Android Runtime	21
2.8.5 Linux Kernel	21
3 TASK SCHEDULER THROUGH ANDROID.....	22
3.1 Task Scheduler - Overview.....	22
3.2 Components Overview.....	22
3.3 Core Components.....	24
3.3.1 Overview	24
3.3.2 User Interface.....	25
3.3.3 Services	27
3.3.4 Content Providers.....	28
3.3.5 Application Resources	29
3.3.6 Data Storage.....	30
4 MODULES	32
4.1 Overview.....	32
4.2 Application Home Screen – Layout 1.....	33
4.2.1 Enter a Task – Layout 2.....	33
4.2.2 Editing the Event – Layout 3	35
4.2.3 Deleting the Event.....	36
4.2.4 Show Task Summary – Layout 4.....	36
4.2.5 Delete Tasks – Layout 5	38
4.2.6 Incomplete Tasks – Layout 6.....	38
4.2.7 Notifications Shown.....	40
5 USER’S MANUAL	41
5.1 Overview.....	41
5.2 Development Environment for Andriod	41
5.3 Installing Application on the Andriod Device	41
5.4 Task Scheduler on Android Device	42
5.4.1 Creating a Task	44
5.4.2 Task Summary Layout.....	44

5.4.3 Editing the Task	46
5.4.4 High, Medium and Low Priorities on Calendar	47
5.4.5 Delete Task	48
5.4.6 Incomplete Tasks	48
5.4.7 Notifications.....	50
6 CONCLUSION.....	52
6.1 Future Enhancements.....	52
6.2 Conclusion	52
6.3 Future Research	53
BIBLIOGRAPHY.....	54

LIST OF TABLES

	PAGE
Table 2.1. IDC: Android Market Share to Top 40% in 2011 - Share (%) of Smartphone Subscribers.....	16

LIST OF FIGURES

	PAGE
Figure 1.1. The main screen of the Quick List app.....	4
Figure 1.2. This figure shows five different colors. Setting task in one of the labels. Entryscreen of Quick List app	4
Figure 1.3. The app is being designed mainly based on the priority basis. It is a more user-friendly app than the existing ones. This figure shows how the app looks	6
Figure 2.1. North America smart phone unit sales by quarter.	14
Figure 2.2. North America smart phone OS market share, Q3 2010.....	15
Figure 2.3. Android architecture.....	19
Figure 3.1. Applications overview design	22
Figure 3.2. Application’s overview design with core components.....	25
Figure 3.3. View hierarchy with View and ViewGroup classes.....	26
Figure 4.1. Displaying work flow of Taskscheduler application.....	32
Figure 4.2. Initial home screen layout – Layout 1.	34
Figure 4.3. Enter a task – Layout 2.....	34
Figure 4.4. Edit event activity.....	36
Figure 4.5. Deleting a task	37
Figure 4.6. Show task summary – Layout 3.	37
Figure 4.7. Delete task – Layout 5.....	39
Figure 4.8. Incomplete tasks – Layout 6.....	39
Figure 5.1. Installation of the android application on device from “unknown source”	42
Figure 5.2. Initial layout displays all buttons and calendar (left) and creating a task with high priority on layout (right).	43
Figure 5.3. An activity showing task created (left) and an activity showing task created with created with high priority medium priority (right).....	45
Figure 5.4. An activity showing task created with low priority.....	45
Figure 5.5. Show task summary layout.....	46
Figure 5.6. Edit a task layout (left), and updated task on task summary layout (right).....	47
Figure 5.7. Layout showing high, medium, low priorities on calendar.....	48

Figure 5.8. Delete tasks layout (left), and a layout showing no tasks to be deleted (right).	49
Figure 5.9. Incomplete tasks layout.	49
Figure 5.10. Alarm notification on Android device.....	50
Figure 5.11. Email notification on yahoo account.....	51

LIST OF ABBREVIATIONS

UI	USER INTERFACE
WAP	WIRELESS ACCESS POINT
IOS	IPHONE OPERATING SYSTEM
GSM	GLOBAL SYSTEM FOR MOBILE COMMUNICATIONS
CDMA	CODE DIVISION MULTIPLE ACCESS
API	APPLICATION PROGRAMMING INTERFACE
ARM	ADVANCED RISC MACHINE
GUI	GRAPHICAL USER INTERFACE
GPRS	GENERAL PACKET RADIO SERVICE
UMTS	UNIVERSAL MOBILE TELECOMMUNICATION SYSTEMS
SDK	SOFTWARE DEVELOPMENT KIT
JVM	JAVA VIRTUAL MACHINE
DVM	DALVIK VIRTUAL MACHINE
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
XML	EXTENDED MARKUP LANGUAGE
IPC	INTERPROCESS COMMUNICATION
JDK	JAVA DEVELOPMENT KIT
ADT	ANDROID DEVELOPER TOOL
AVD	ANDROID VIRTUAL DEVICE

CHAPTER 1

INTRODUCTION

1.1 EMERGING MOBILE PLATFORMS

With changing times, mobile technology has developed a lot and during the last few years we have seen innovative smartphones and operating systems. Mobile operating systems have evolved rapidly. Popular vendors like AT&T, T-mobile, Verizon etc. have all participated competitive vendors. These competitive vendors have expanded roles, features and functionality of smartphone models. Each different mobile platform has a unique execution environment and functionality. Mobile platforms are developed in terms of a software stack with functional layers like application execution environment, kernel, middleware layer, UI framework, application suite etc. All these layers contribute to the design of a proper mobile handset for the current competitive market around the world. Software vendors are expanding their market by releasing different versions of mobile platforms every year with innovative and distinct applications on the devices. Mobile platforms like Google Android, iPhone OS, Symbian etc are evolving and experiencing heavy competition in the market.

Each platform has its own style of UI framework, application execution environment and features. This improvement & competition has resulted in heavy usage of smartphones. Increased sales lead to more applications, which lead to more sales, a pleasant cycle for the industry [6].

The recent and rapid application development in mobile technology has increased the demand for sophisticated software applications for mobile phones. This technological- driven market evolution offers new opportunities and challenges for developers to create innovative applications, which will fully exploit mobile's new and expanding capabilities. In this chapter we will investigate what are the existing mobile applications, how they are developed, what are their potential future directions. The purpose is to highlight any important aspects of mobile application development which should be considered by developers in unleashing the great potential of mobile technologies. Mobile applications are developed for social and commercial purposes which attract many consumers. The concept

of being connected anytime and anywhere leads to various mobile applications which aim at satisfying diverse user needs. Development of mobile applications mainly depends on three factors:

Maturity of mobile network.

Advanced mobile hardware.

Increasing demand for mobile applications/services.

The recent revolution in the mobile technology 4G has become a new trend with increasing bandwidth and user friendly hardware and software features. These evolutions in mobile technologies have renewed an interest in the developers to create innovative applications. According to statistics, the total number of mobile smartphone users around worldwide will reach 975 million by 2012 [7].

1.2 MOBILE OPERATING SYSTEM

A mobile operating system is also known as mobile OS, a mobile platform or hand held operating system .An operating system is one which controls the mobile device.

Similarly, the desktop operating system is one which controls a desktop computer or laptop such as Mac OS X, Windows, Linux etc.

Other programs run on top of mobile platform. Operating system deals with broadband wireless versions, local connectivity and mobile multi-media formats. Examples of devices running a mobile operating system are smart phones, personal digital assistants (PDAs), tablet computers and information appliances.

Each mobile device has some specific operating system, which is chosen by the manufacturer. The operating system determines the functions, and features available on mobile device such as keyboards, WAP, synchronization, thumbwheel with applications, e-mail, text messaging and much more.

The mobile operating also influences which third party applications should be used on your device.

There are various mobile operating systems like Google Android, Apple iPhone OS, Symbian, windows Mobile etc. Challenges for mobile application development include power consumption, cross-platform compatibility and software quality. Our primary focus is on the android platform and its applications. We will now discuss existing android

applications. Android platform has applications like Google latitude, Google maps, which gather real time information concerning location, android market etc.

Let's talk about To Do apps which are currently in the market, because our research deals with developing a To Do app with email synchronization. The android market holds thousands of apps in the market. All the apps of the android market are differentiated in to different categories. Some of the To-Do list Android applications of the android market are Do List, GTasks, Astrid Task, Quick ToDo and ToDo Today with various features.

1.3 EXISTING TODO ANDROID APPLICATIONS

This section gives the overview of existing ToDo application in the android market. Features and flaws of the application will be discussed below in detail.

1.3.1 ToDo Application

Android market has thousands of applications. Android has a number of ToDo applications which are very much useful for our day- day life. Let's discuss an existing ToDo application and its flaws. My research focuses on fixing these flaws with task priorities, email synchronization, alarm etc. The existing ToDo application in the android market is Quicklist.

1.3.2 Brief Discussion About an Existing App (Quick List)

Quick List is an app that allows you to create quick and instant lists on your android phone. We usually need to create lists in our daily lives to make our tasks easier and get things done. Quick List is an app for the Android platform that makes it easy for you to create tasks on the go. (See Figure 1.1 [3]).

You can create and clean up items. There aren't many options to configure in this app, but you can make it your own. If you'd like to, you can actually rename them as per your needs. The colors should be used in their default order for various tasks that you set up [12]. (See Figure 1.2 [3]).

After you set up tasks, you can easily start striking out the tasks you complete. This makes it easy to keep track of what you've finished. You can also clear out all of the items that you've completed. Five different colors are available in this app to differentiate items on

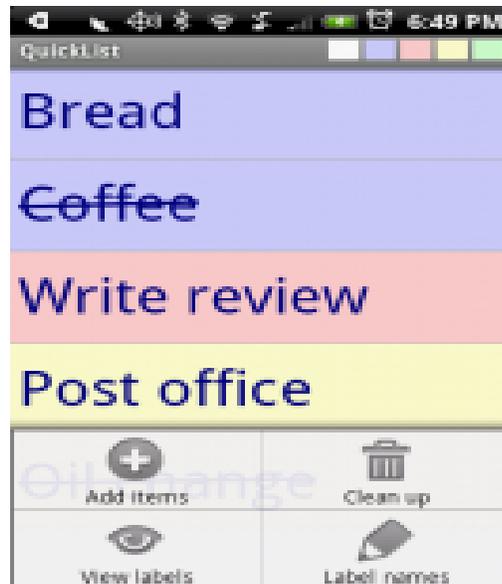


Figure 1.1. The main screen of the Quick List app. It adds items, clean up, view labels and label names. Source: ANDROID TAPP, *QuickList*. Android Tapp, <http://www.androidtapp.com/quicklist/>, accessed July 2011, n.d.



Figure 1.2. This figure shows five different colors. Setting task in one of the labels. Entryscreen of Quick List app. Source: ANDROID TAPP, *QuickList*. Android Tapp, <http://www.androidtapp.com/quicklist/>, accessed July 2011, n.d.

your list; this reduces the confusion in finding the specific task. You can also rename items on your list which are important since they are named after their color by default. The colors are just visual aids.

Some flaws:

1. This application does not allow you to rearrange the order of colors on your list. You have to follow the default arrangement of colors on your list.
2. It doesn't list the tasks according to a priority basis.
3. It won't be offering the user to add a subtask within an existing task.
4. This isn't a note taking app by any means. Anything more than one or two words will go off the screen with no way to scroll to see the rest of it.
5. It doesn't have any task reminder of any sort.

Pros & Cons – Pros:

- Simple to use.
- Fast way to stay organized.
- No need to carry around separate lists of what needs to get done.

Cons:

- No scribbling with your finger.
- No organized way of tasks.
- No notifications.
- Not able to add some extra notes to the existing tasks.
- No date and time setup.

This app isn't robust and it really doesn't have to be. The Android platform has need for all these little apps that offer something new to the user. As you can see the app is simple but not perfect it lacks so many features which are valuable but the idea is simple [3].

So my goal is to improve this app by fixing many of the flaws.

1.4 MY APPLICATION

The basic idea is to create a user-friendly app, also focusing on the flaws of the existing apps in the relevant area to bring up a more sophisticated design.

Key Features of my app:

1. The first and foremost thing to be done is the layout, which is designed to display 4 sections:

- a. Enter a task. This segment displays an android activity to enter all task details like title, description, priority, calendar etc.
 - b. Show task summary. This section displays an android activity which lists all tasks saved under each of its priority levels like high-level, medium-level, lower level priority along with unprioritized section.
 - c. Delete task. This section displays an android activity which shows a list of tasks to be deleted.
 - d. Incomplete tasks. This section displays a list of incomplete tasks. The layout also displays a calendar which shows all saved tasks that are saved on particular days.
2. Users can delete, edit and clear the tasks any time.
 3. Users will be notified with email notifications such as Gmail notification, Yahoo mail notification and organization mail account. All Tasks will be synced with Gmail account when you configure the Gmail account. (See Figure 1.3).

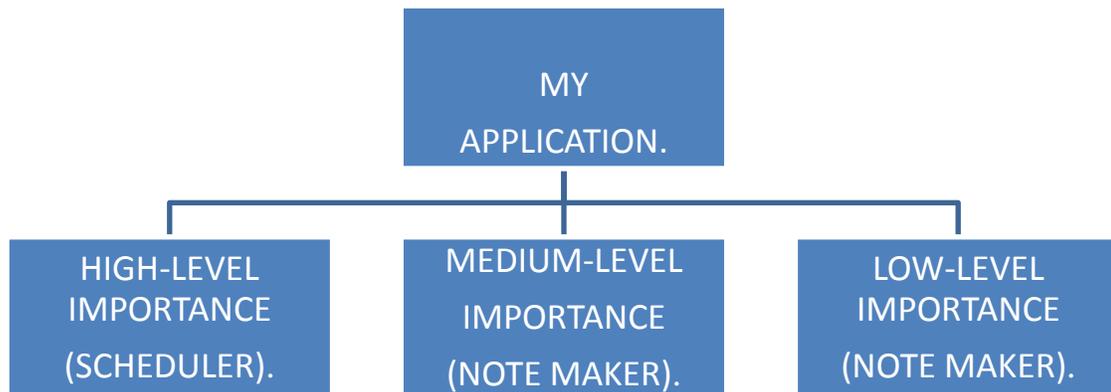


Figure 1.3. The app is being designed mainly based on the priority basis. It is a more user-friendly app than the existing ones. This figure shows how the app looks. As mentioned earlier it has got 3 categories where the user can schedule the tasks according to their priority and preference for the tasks.

1.5 RESEARCH AND STUDY

To Do App lets you keep track of everything you need to do as per date and time. There are many To-Do Apps in the android market with various features. To Do apps of android market have basic features like backup/restore, task reminders, repeating tasks, edit and delete tasks, set alerts and make calls, speech to Text. So almost all the To Do applications have these basic features, But do all of these To-Do apps have email synchronization? Getting email notifications for the tasks you schedule? Do these apps support email reminders in multiple email accounts whether it could be Gmail, Yahoo mail and any Company email accounts? Do any of the apps have priority task setting?

Setting the priority of the task with priority level such as high-level, medium level and low level priority basis with specific date and time and notify the user about the task with priority basis as an email notification. This research will categorize tasks into 3 priorities, (1) High level, (2) Medium level, and (3) Low level importance for the tasks. The tasks will be given priorities based on date and time so the users will know the priority of the each task. The user Interface will have a calendar in it which displays all tasks that are created on a particular date with a color indicating its significance. A task will be easily identified with a color coding to figure out if it is high level, medium level or low level.

Getting notifications in email will help to notify the user and finish up the tasks. Even when we forget the mobile, we should be notified and reminded by email notification. After a quick research none of the apps have multiple email notifications feature. Because most of the android devices have Gmail account app on the phone with calendar sync by default so it will be more flexible for user to quickly schedule and get notified through it.

So from the default Gmail application, we should be able to send email reminders which can be used as an invitation for a business meeting to another person in any email account other than Gmail. An application should have sender email id and receiver email id for the purpose of sending an invite to another user. Similarly, business people should be benefited by using company email accounts to store the tasks from their device. In this way, user like students and business people will be benefited by getting notified through email notifications in multiple email accounts like Gmail, Yahoo mail and corporate email accounts on the web as well through Alarm notification on a phone. As a result the user can send the notification email from Gmail account to other email accounts like Gmail, Yahoo, corporate email accounts providing more flexibility to users to get notified for their tasks. So my research focuses on the key aspects of the application so the users will get access to Gmail, Yahoo mail and corporate email accounts to receive email notifications as well as through alarm notifications on phone for their priority based day-day tasks.

1.6 RESEARCH OBJECTIVE

This research develops an android ToDo app named as “Task Scheduler for Android”. This app schedule priorities to tasks such as ‘High level’, ‘Medium level’ and ‘Low level’ based on date and time and gets notification emails in Gmail, Yahoo, corporate

email accounts and also alarm reminders on phone. There will be Gmail calendar sync up such that it will sync the tasks that are created on android phone to the Gmail calendar and stores in it when Gmail id is selected as sender. There will be calendar within the home screen layout such that all the tasks that are created will be stored in the calendar. Each task will have color significance to represent its priority in calendar. In General, Most people uses Gmail account by default on the android device so it is more flexible for the user to send a notification email or invitation from Gmail to other accounts. With the sender and receiver email address option we can be able to send the tasks as email notification as well as an invitation to the receiver to remind about tasks. That's how users will be notified easily through email notifications as well as through alarm. This application will keep track of list of Incomplete tasks, task summary and tasks to be deleted in different layouts.

For business people, During the work they often use corporate email account, through this application, users can receive email notification reminder in their corporate mail account from Gmail account in order to quickly get notified even through corporate mail account when there is situation they don't have access to other email accounts in a company. So throughout application Gmail account will serve as a host to send email notifications to other email accounts providing the easy access to accounts to get notified apart from alarm notification. This research helps to get notification emails to any user account from Gmail also works for the purpose of sending invitation to any users for events, group meeting etc. This research provides much more flexibility to the user to give the quick notice for prioritized tasks in two different ways such as Alarm and Email notifications. Goal of the research is to provide simple and easy to use Android application to sort out the tasks according to their importance with respect to date and time and gets notified through Alarm and Email notifications with calendar sync up option on the device. So in that way users will be greatly benefited to finish up the important tasks in their daily life and have good time management. The user itself can assign the priority to the tasks and can keep track of various tasks with different priority levels and gets notified on time to accomplish those tasks.

This application is more user friendly with minimum number of layouts unlike other applications. The usability of the application is even easy and simple because users who are new to Android OS can easily understand the usability of this application makes them feel comfortable as very easy to use application. This application is definitely the best application

with simple functionality and user interface among other existing applications. The layout of application gives clear idea with various importance levels of tasks and memory used by application is even less. This is similar to other applications but learning the techniques used behind such an application and such learning process is one of my research objectives. This application with its user friendly interface will be definitely be best and useful application among others apps.

CHAPTER 2

MOBILE OPERATING SYSTEMS

In this chapter we will discuss about different types of mobile operating systems and their development.

2.1 MOBILE OPERATING SYSTEMS OVERVIEW

This section gives introduction and overview of various mobile operating systems. Following are the mobile operating systems to be discussed about growth, market statistics, advantages etc.

2.1.1 Iphone OS

Iphone OS is also known as IOS. It is an Apple's mobile operating system, which is originally developed for iphone. IOS also supports other apple devices such as iPod touch, iPad and Apple TV. Apple Inc does not provide license to install its OS on any third party hardware devices. An interesting thing about Apple Inc, App store has more than 500,000 iOS applications. All the applications are handy and easy to use. In year 2010, it had a 26% Share of the smartphone operating systems market behind Google's Android and Nokia's Symbian.

iOS is derived from Mac OS X, with which it shares the Darwin foundation, so therefore it is almost like Unix like operating system. The operating system was announced officially with iPhone at the Macworld conference and Expo on January 2007. iPhone applications like iOS and Mac OS X are written in Objective-C [14].

The operating system also features new applications such as "Reminders" app and "Newsstand" an application is all about Folders and iBooks. Besides the features and advantages I phone has some Limitations too such as single Vendor/Hardware availability through Apple Inc, the single provider of service through AT&T, the lack of external memory expansion and many more.

Sometimes user needs to handle two tasks at a time for example switching between email browser and any other app. Users can't get back to the place of previous app while

switching back from another app. This was not properly handled well by previous versions of iPhone OS until the release of iPhone 4 OS. IOS needs license to install software on any hardware whereas Android is open source and free to download on any hardware. This is a disadvantage over iPhone OS might lead users to attract more towards Android OS [10].

2.1.2 Windows Mobile

Windows mobile was introduced by Microsoft. Windows OS is available on smartphones and mobile devices. Windows mobile is based on Windows CE 5.2 kernel, and first appeared as the Pocket PC 2000 operating system. Windows Mobile powered devices are available on GSM or CDMA networks. It is supplied with basic suite of applications developed with Microsoft windows API. It is designed in a way to have features and appearance similar to windows desktop versions. Windows Mobile's share of the smartphone market has been in decline year-on-year.

Gartner research provided data that total smartphone share grew 27% between 2008 and 2009 where as Windows Mobile market share fell 2.7% in the same year.

The applications for Windows Mobile are written using native code with Visual C++, writing managed code which is compatible with .NET Compact Framework or server-side code. The .NET Framework offers narrowed functionality of the framework. Windows applications should be optimized for touch screen user interfaces in order to be convenient for the users.

Windows CE Operating system is device specific and may use generic functions to expand the functionality. Windows mobile runs on .NET virtual machine programmed in VB.NET and C#.NET. Windows mobile are not as popular as Java based mobiles. Windows mobile has quite a few limitations, they are development complexity is medium or low, profit is medium or low, no multitasking , no video calls, no internet tethering support, no Bluetooth file transfers, no handwriting recognition support etc . These limitations can be a good cause for the users to attract more towards Android OS phones having wide range of efficient functionalities [16].

2.1.3 Symbian OS

Symbian OS is a mobile platform introduced by Nokia for smartphones. It has become a standard operating system and is licensed by more than 85 percent of the world's

handset manufacturers. Symbian OS is originally developed by Symbian Ltd. It is a successor to Symbian OS and Nokia Series 60. The kernel type of Symbian OS is Microkernel and it is descendent of Psion's EPOC. The supported platforms for Symbian OS are ARM and x86. The applications for Symbian OS are written in C++. The default user interface for Symbian OS is based on S60 platform and later to use GUI.

Nokia announced to migrate from Symbian to Windows Phone 7. Symbian was formed by merging and integrating software assets contributed by Nokia, NTT DoCoMo, Sony Ericsson and Symbian Ltd. As of June 2011, over 39% of mobile developers were using Symbian, were planning to abandon the platform. When talking about user interface for Symbian, It has a native graphics toolkit which is known as AVKON and also known as Series 60. S60 is a 15key augmented telephone keypad or mini-QWERTY keyboard. Full Virtual machine for Symbian OS is not implementable. Symbian supports for multihoming capabilities and link layer QoS on GPRS/UMTS [15].

The limitation about Symbian OS is its UI framework is fragmented but later it was resolved by new version of Symbian OS and OS is not available to PC's and not possible to develop Symbian applications in native environment. Besides this, Android is open source and developers are free to develop any kind of applications with Android platform and environment. This is why Android is been attracted by many users and holding first place in the market [11].

2.1.4 Android OS

Android is a mobile operating system introduced in November 2007 by Google for mobile devices such as smartphones and tablet computers. Initially developer of software is Android Inc. Later Google owned it in 2005. Android OS is based on Linux kernel and with founding of open source handset alliance. The Android Open Source Project (AOSP) is targeted with maintenance and further development of its OS. API's, middleware and libraries are written in C and application software. Android uses Dalvik virtual machine and its applications are written in Java. Because its an open source it attracts many developers come forward to contribute and show their innovation towards it.

The hardware platform for android is ARM architecture. The android OS is used on multiple devices like smartphones, netbooks, tablet computers, Google TV and other devices.

Android applications are developed in Java programming language using Android Software Development Kit, i.e., Android SDK.

Android OS came up with initial OS version as 2.0 (Éclair), 2.1 (Froyo), 2.3 (Gingerbread), 3.0 (Honeycomb) and upcoming OS version is 4.0 (Ice cream sandwich) is said to be the combination of Gingerbread and Honeycomb. It will most likely to be released in Q4 2011. Android Kernel is derived from Linux kernel. The notable features of Android are handset Layouts, Storage, Connectivity, Messaging, Multiple Language support, Web browser, Java support, Media support, Streaming media support, Additional hardware support, Multi-touch, Blue tooth, Video calling, Multi-tasking, Voice calling, Voice based features, Tethering, Screen capture etc. [13].

Android's hardware platform is the ARM architecture. The other Android supported platforms are ARM, MIPS, x86. According to research analysis, as of Q2 2009, android had a 2.8% of worldwide market share for smartphones. By Q4 2010, market share had grown to 33% becoming top-selling mobile platform. As of August 2011, Android has about 48% of smartphone market share. Android Green is color of Android Robot which represents Android Operating system.

2.2 WHY IS ANDROID MORE POPULAR THAN OTHER SMARTPHONES?

The Android is more popular than other Smartphones for the following reasons:

- Google is offering free to use open source versatile OS which is built on Linux kernel. It makes software developers to create exciting third-party applications.
- Android is more innovative OS that can match with other smartphone capabilities. These innovative features of Android are full filling user needs.
- Android is powerful software developed by many Google engineers that could power smartphone. On basis of market share and its applications Android is always considered as a success [8].
- Android phone has a larger display and more responsive swiping technology which can attract many users.
- Android has many 4G smartphones like HTC Thunderbolt. Now -a- days, Android phones are coming up with bigger RAM.
- Android OS comes up with variety of designs from many manufacturers like HTC, Motorola and Samsung. Android smartphones are powered by dual-core processor.

- Android is an Open Source OS, where we can get access to source of android. In market, there are many handsets that run Android OS with good camera, large screen, removable battery [9].

2.3 SMARTPHONE MARKET SHARE

Android Market Growth: Google offered an open source solution Android for mobile devices. Any handset manufacturer is free to use the platform for mobile application development. With reference to Figure 2.1 [5], Android has a constant growth in market sales over the other mobile platforms with its efficient technology and as an open handset alliance. Figure 2.1 [5] shows North America smart phone unit sales by quarter. Only Android's sales show significant growth each quarter for the past seven quarters. As of September 2010, Android has significant growth as Android sales numbered 2,00,000 to that of 80,000 iPhone sales per day, blackberry is steady over seven quarters like 5-6 million per quarter. For six quarters iPhone is about 3 million per quarter. From the market share statistics Android is the only phone has significant growth in Q2 2010, beating iPhone and tying with blackberry. In Q3 2010, Android and Iphone grew by 61 percent and Android became a market leader among other mobile platforms.

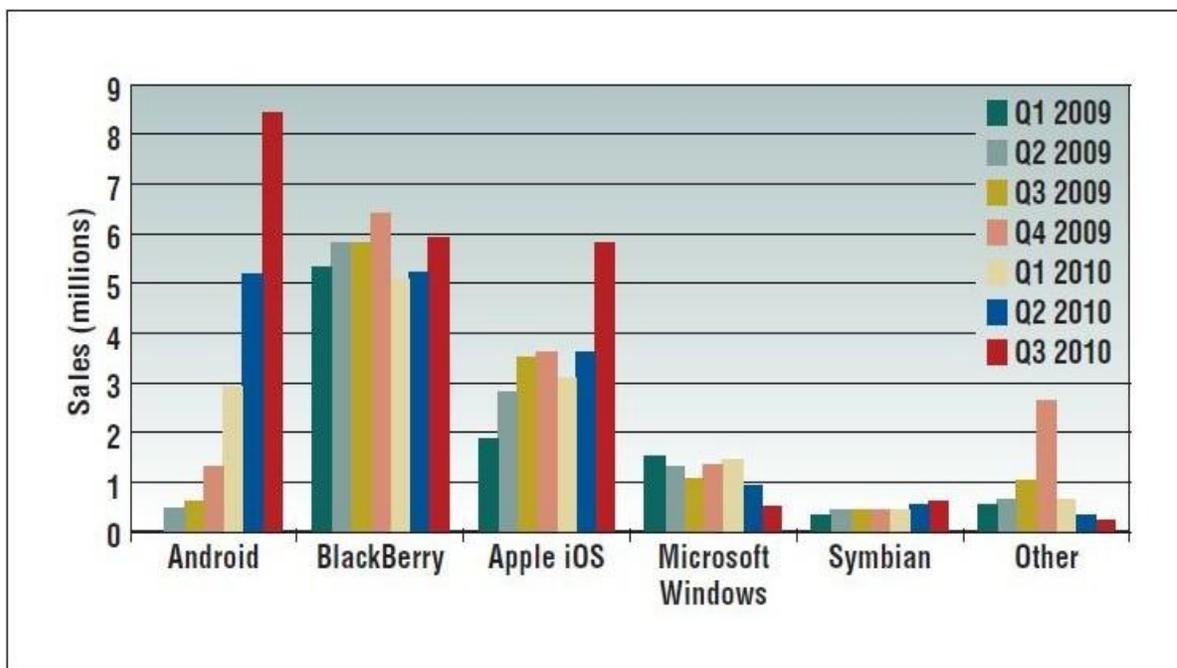


Figure 2.1. North America smart phone unit sales by quarter. Only Android's sales show significant growth each quarter for the past seven quarters. Source: M. BUTLER, *Android changing the mobile landscape*, IEEE Perv. Comp., 10 (2011), pp. 4-7.

From Figure 2.2 [5], In Q3 2010, with the growth and market share Android is a lead with 39 % market share where as Apple and Blackberry ended up with only 27% and 28 % market shares [5].

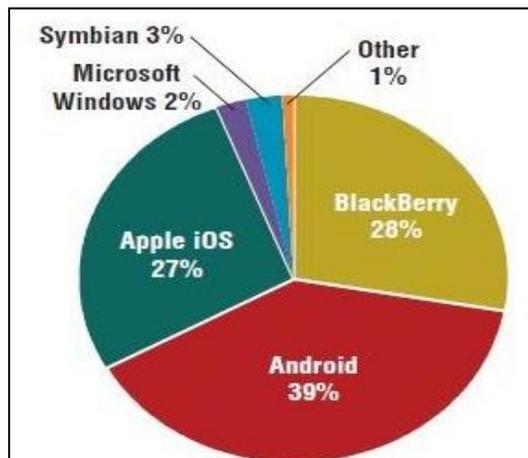


Figure 2.2. North America smart phone OS market share, Q3 2010. Android became the market leader, surpassing both iPhone and Blackberry, and now leads with a 39 percent market share. Source: M. BUTLER, *Android changing the mobile landscape*, IEEE Perv. Comp., 10 (2011), pp. 4-7.

Among the other smartphones, Google Android is greatly benefitted since last year Google Android predominantly dominated over other smartphone market shares. This market statistics shows as good example why Android is beating other smartphones in terms of applications, technology, resources etc. Google Android attracts many consumers with its innovative applications, multitasking and primarily being an open source operating system. The advantage of Android is coming with wide variety of smartphone models with different carriers like AT&T, Verizon, T-mobile etc.

Smartphone Market share in 2011: From the statistics found in Table 2.1 [4], Google Android ranked as top operating system with 38.1 market share compared to other smartphone OS such as Apple with 26.6%, RIM with 24.7%, Microsoft with 5.8% and palm with 2.4% as of May, 2011. From the above analysis, Google Android has been standing as popular and attractive OS to the consumers with its innovative features [4].

Table 2.1. IDC: Android Market Share to Top 40% in 2011 - Share (%) of Smartphone Subscribers

	Feb-11	May-11	Point change
Total Smartphone subscribers	100.0%	100.0%	N/A
Google	33.0%	38.1%	5.1
Apple	25.2%	26.6%	1.4
RIM	28.9%	24.7%	-4.2
Microsoft	7.7%	5.8%	-1.9
Palm	2.8%	2.4%	-0.4

Source: C. BOULTON, *Android market share to top 40% in 2011: IDC*. Eweek.com, <http://www.cioinsight.com/c/a/Latest-News/Android-to-Top-40-Market-Share-in-2011-IDC-664673/>, accessed September 2011, Jun. 2011.

2.4 APPLICATION RESOURCES FOR VARIOUS SMARTPHONES

Google Launched an app called “Android market” where there is catalog of application services and tools to download the various apps. Similarly, Apple launched an application for downloading the apps called “Iphone App store”, it has more than 195,000 applications and blackberry launched an app called “Blackberry app world” for downloading the apps.

This chapter introduces Android mobile OS, Android SDK, Android features, brief history of Android OS versions and its architecture will be explained in detail. Before going to brief explanation of all these there is basic terminology to be mentioned about Android.

2.5 ANDROID MOBILE PLATFORM

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides necessary tools and API’s to get started on developing applications, features on the Android platform using Java programming language and it runs on an x86 CPU. So that is the reason Android is considered as Server platform because it can also run on x86 platform servers along with smartphones.

Android is mobile operating system for smartphones and tablet devices. It is an open handset alliance led by Google [2].

2.6 INTRODUCTION TO ANDROID

With the introduction of Android, Mobile phone market has changed widely in terms of growth and attracted many consumers. Android is always biggest threat to iPhone market as it aimed to win hearts of mobile application developers since developers are free to develop Java applications on its open source mobile platform. Android is an open source operating system for mobile phones and tablets led by Google. It was started by small software firm called Android Inc, Later Google owned it in year 2005. Android is basically a modified version of Linux Kernel. Google announced android distribution with founding of open handset alliance on November 5, 2007. A group of hardware, software, telecom companies were devoted and collaborated to advancing open standards for mobile devices. Google released most of the Android code under free software license called “Apache License”. Main purpose of AOSP (Android Open Source Project) is to develop Android platform and better maintenance of it. Android uses Linux kernel with middleware, libraries and API's. These API's are written in C and application software runs on a application framework includes java compatible libraries.

Android market has more than 2,50,000 android apps . All applications are written in java programming language and it uses custom Java Virtual Machine (JVM) which runs on top of Linux kernel. All android applications are in .apk format which contains compiled Java code for Dalvik Virtual Machine, which executes files in .dex format which is a machine understandable code.

From the past two years, Android is listed as best selling phone in smartphone market share.

2.7 ANDROID PLATFORM FEATURES

Android platform features include the following:

- **Application framework:** It is designed to reuse and replacement of components by the user. Android offers developers to build innovative and rich applications and it provides to access device hardware, access location information, run background services, set alarms, add notifications to status bar etc. Developers can access same

framework API's used by core applications. It uses power management framework which can save power in mobile devices.

- **Dalvik Virtual Machine:** It is an Android platform's virtual machine. It is an interpreter only virtual machine which executes files in Dalvik executable format called .dex format. This format is optimized for efficient storage and memory-mappable execution. The virtual machine is register based, and it can run classes compiled by Java language compiler that have been transformed into its native format using "dx" tool.
- **Integrated browser:** Android has a web browser and it is based on open source webkit layout engine and it is collaborated with Chrome's V8 JavaScript engine.
- **Optimized graphics:** Android platform is powered by 2D graphics library and 3D graphics. 3D graphics, it is based on OpenGL ES 1.0 specification. We can use this OpenGL Es for fast and complex 3D images when developing applications on Android.
- **SQLite:** SQLite database manages private database in any application. All Android applications use SQLite to manage their own private database. Android platform uses SQLite version 3.4.0. It is powerful and reliable database engine is available to all android applications.
- **Media support:** Android's media support supports various audio, video and still image formats such as MPEG4, H.263, H.264, MP3, AAC, MIDI, Ogg Vorbis, FLAC, WAV, AMR, BMP, JPG, PNG, GIF.
- **GSM Telephony:** It is hardware dependent. Provides GSM telephony like text/data/PDU SMS messages.
- **Bluetooth, EDGE, 3G and WiFi:** It is hardware dependent. Bluetooth supports some specific features like AVRCP, sending files(OPP), A2DP, accessing phone book(PBAP), voice dialing and sending contacts between phone and in addition it has features like EDGE, 3G and WiFi.
- **Camera, GPS, compass and accelerometer:** These features are hardware dependent. Also Android platform supports other hardware support such as gyroscopes, magnetometer, dedicated gaming controls, proximity, thermometers, accelerated 2D bit and accelerated 3D graphics.
- **Rich development environment:** Android has a rich development environment which includes a device emulator, tools for debugging, memory and performance profiling and a plugin for Eclipse IDE, Android SDK, AVD manager and other additional tools available to develop applications more efficiently.

2.8 ANDROID ARCHITECTURE

Android Operating system is a complete mobile Operating system which supports middleware, key applications and other tools. The basic Android architecture includes Applications, Application framework, Libraries, Android run time, Linux kernel

(see Figure 2.3 [2]). All parts of the Android architecture will be explained in detail in the following sections.

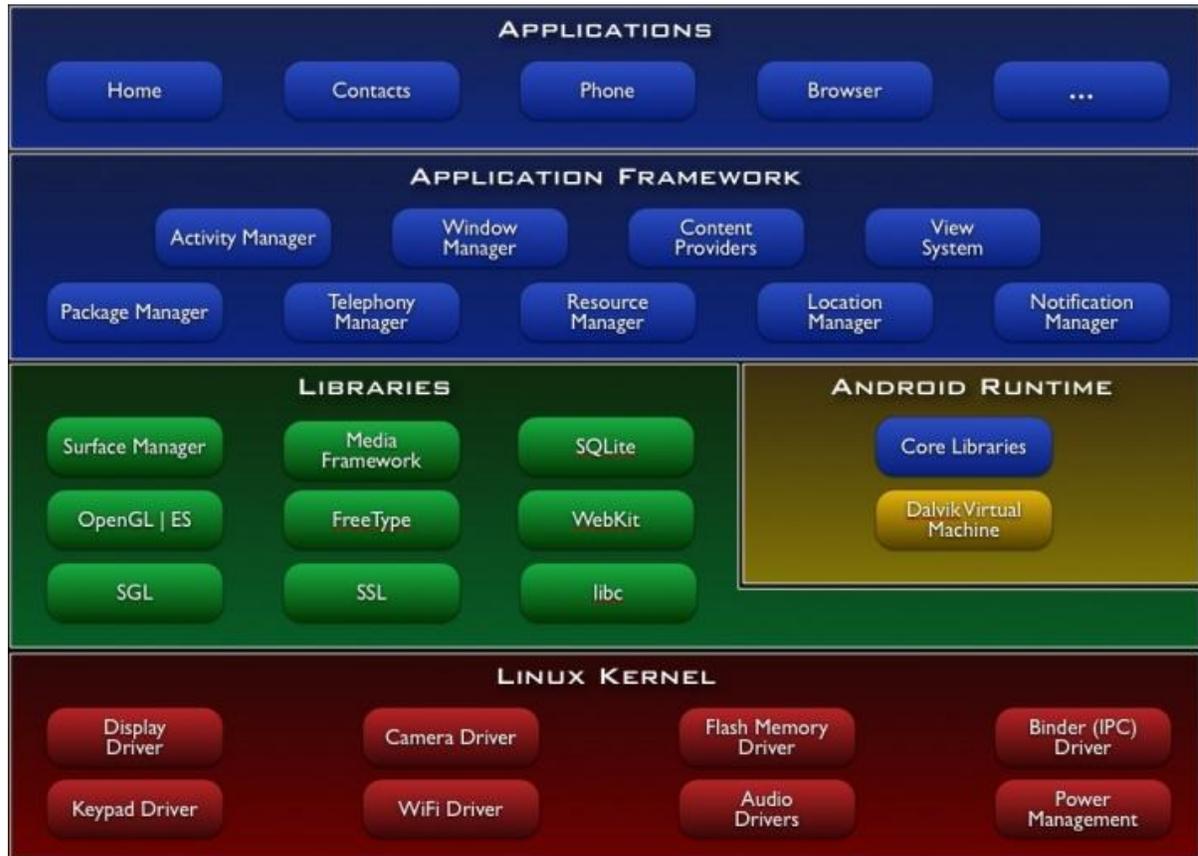


Figure 2.3. Android architecture. Source: ANDROID DEVELOPERS, *What is Android 2011? Android Developers*, <http://developer.android.com/guide/basics/what-is-android.html>, accessed October 2011, n.d.

2.8.1 Applications

Android is shipped with basic applications like home, contacts, phone, Browser etc. All the applications are written in java programming language and can run simultaneously.

2.8.2 Application Framework

The Android framework consists of the following:

- Android consist of set of core apps including an email client, SMS program, calendar, maps, browser, contact and others. All applications are written using Java programming language.
- Android Application framework consists of activity manager, window manager, content providers, view system, package manager, telephony manager, resource manager, location manager, notification manager.

- People developing android applications will have complete permissions to access application framework APIs used by core applications. Any application can reuse the components to publish each of its capabilities:
 - A rich and extensible set of views that can be used to build an complete android app, including lists, grids, text boxes, buttons, and even an embedded web browser. Content providers provides data access from other applications such as contacts, email etc.
 - Resource manager enables access to non-code resources such as localized strings, graphics and layout files.
 - Notification manager enables all applications to provide notifications and alerts in the status bar.
 - Activity manager manages lifecycle of activities in applications and provides common navigation back stack. Android applications are composed with no of activities.

2.8.3 Libraries

The libraries consist of the following:

- Android platform is a set of C/C++ libraries used used by various components of Android application system. Application framework provides access to developers to use all resources and library components in order to develop applications. Following are the Android core libraries.
- Android architecture consists of various libraries. They are System C library, Media libraries, Surface manager, LibWebCore, SGL, 3D libraries, Free Type, SQLite.
 - System C library – It is BSD derived implementation of standard C system library (libc), designed for embedded Linux based system.
 - Media libraries – Packet video OpenCore system provides playback and recording of audio/video formats and also static image files such as MPEG4, H.264, MP3, AAC, AMR JPG and PNG. Media libraries are responsible for entire audio and video playback.
 - Surface Manager – It is basically composites of 2D and 3D graphic layers from various applications so as to access to display system.
 - LibWebCore – Android browser and embeddable web system are powered by modern web browser engine called LibWebCore.
 - SGL – SGL provides entire 2D graphic features for the android applications. It is taken care by 2D graphics engine.
 - 3D libraries – 3D libraries are based on the OpenGL ES 1.0 APIs, this is basically a optimized 3D software.
 - Free type – Free type of library provides bit map and vector font rendering system.

- SQLite – This is android database where any private database from any android application can be stored. This is a powerful and lightweight relational database engine to store all android databases.

2.8.4 Android Runtime

Android runtime consists of the following:

- The two most important parts are Core libraries and Dalvik virtual machine. Android uses functionalities that are available in core libraries of the java programming language.
- Dalvik virtual machine executes all android applications in dalvik executable format called .dex format. It runs classes of all android applications compiled by java language compiler and translates to .dex format so that machine OS understands it.
- Multiple VMs can run parallel without affecting one another. Dalvik acts as mediator between application and the OS. Dalvik virtual machine relies on linux kernel for underlying functionality.

2.8.5 Linux Kernel

It provides services like security, memory management, process management, network stack and driver model. Linux kernel acts as abstraction layer between and hardware and rest of the software stack. Android uses several key components in linux kernel such as power management driver, binder IPC driver, kernel debugger, low memory killer and shared memory driver to support low memory devices [2].

CHAPTER 3

TASK SCHEDULER THROUGH ANDROID

3.1 TASK SCHEDULER - OVERVIEW

Task scheduler is an application built using Android framework for Android mobile phone. The application is used to keep track of our day-day tasks with due date and time and gives a reminder like email notification and alarm.

3.2 COMPONENTS OVERVIEW

Task scheduler is an Android application to create tasks and set up with due date and time. These tasks will be differentiated with priority basis such high level, medium level and low level priority basis based on date and time. User can assign the priority to each task, will be notified with email notifications in our email accounts and also with Alarm. Figure 3.1 shows the Flow of Task scheduler application.

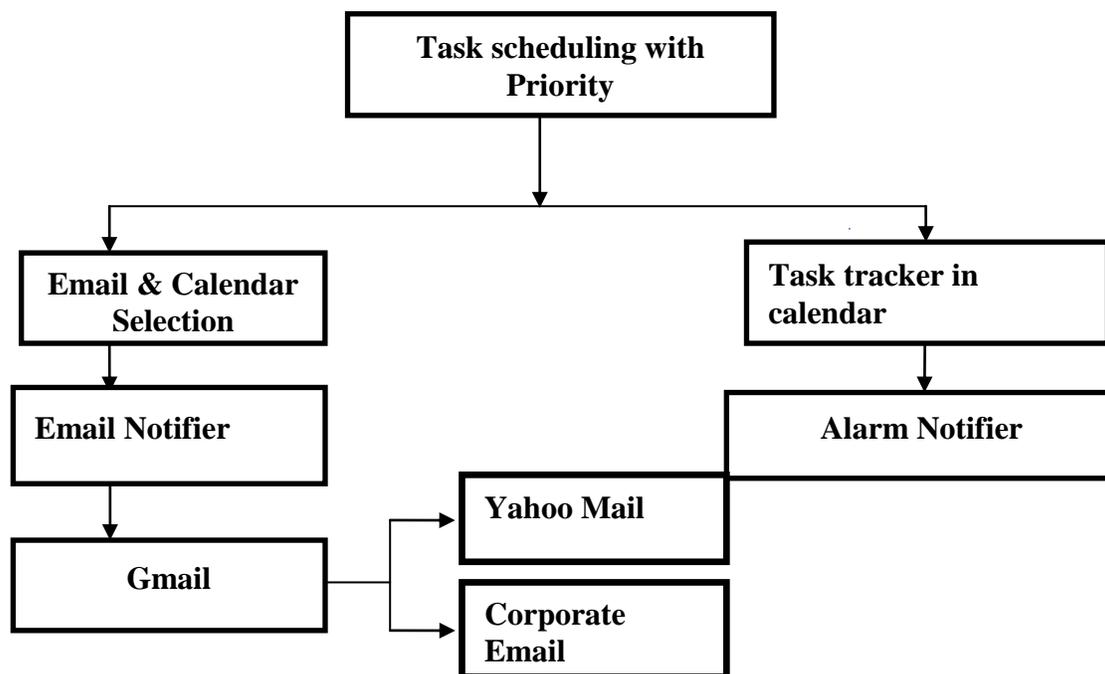


Figure 3.1. Applications overview design.

This application has total 4 layouts. First layout has 4 different textviews. They are (1) Enter a task, (2) Show task summary, (3) Delete task, and (4) Incomplete task. Each tab

will redirect to its specific layout, initial layout has calendar which will store all the tasks with their specific color. High level priority task will be assigned 'Red color', Medium level priority task will be assigned 'Yellow color' and low level priority task will be assigned 'green color' and unprioritized task will be assigned 'blue' color. This app allows you to create, edit, and delete tasks at any time. It allows you to select any Gmail account from spinner (drop down list) as per user's choice and send email notifications to the receiver email address whether it is a Gmail, Yahoo mail and corporate email. It can also be used as an invitation to send to any other user.

Task Scheduler with priority loads List View Activity lets you keep track of tasks with date and time using Android Activity. It uses `android.widget.view` and `android.text` packages to interact with user to retrieve inputs. It will assign the priority to the task based on date and time that is set. Date Picker and Time Picker widgets are used in this Activity.

Email & Calendar selection are designed on same Activity with Android spinner to make selection from multiple options to select the user Gmail account. Since the application is based on android platform user need to set up default Gmail account on the device. Initially User should provide the credentials to login for account setup. User is able to view the saved tasks in Gmail account calendar if they login.

Task tracker in email calendar, in this section tasks will be saved in calendar that is designed on the layout. It will show the dates in calendar that displays color assigned to each priority when user selects the particular date it will display the small description of the task. Phone will sync up with user account to keep track of tasks. All tasks will be saved in Gmail calendar for quick reference. Application uses Calendar class from `java.util` package in the activity to store the user created tasks on phone.

Email Notifier lets you get notified with email notifications in various email accounts or another user if you would like to send as an invitation. It sends task details to recipient address whatever you type in recipient address section. Application uses calendar android core layout to implement this functionality. Users will be able to get email notifications to the particular account when it is sent as an invite from sender email address (Gmail) apart from alarm reminder. From the block diagram, Notifications or Invitations will be sent from Gmail account to Yahoo mail or corporate email based on user selection from the spinner. Not only through Alarm notification, user will be notified through Email reminder.

Task tracker in phone calendar lets you keep track of tasks in phone when you select ‘calendar’ option from the spinner. It will save the tasks on the phone calendar. It uses android widget classes like Date Picker and Time Picker which are implemented by OnItemClickListener Interface. It uses List View Activity to store all the tasks in SQLite Database. It uses Calendar View class to select and adjust the dates for tasks.

Alarm Notifier lets you receive notifications on the phone through Alarm. It uses Date Picker and Time Picker widget classes to set the time and notify when the task is due to complete. It is implemented by android.app.NotificationManager, android.app.AlarmManager classes.

3.3 CORE COMPONENTS

Core components are main components used to build a project and implements the flow of each and every module. The design of each and every core component forms backbone of the project implementation. This section illustrates main and core components that are involved in implementing the project.

3.3.1 Overview

Activity: Android application is built using number of activities. Each and every activity is implemented for specific functionality. Activity is a single focused component which provides access to the user in form of a screen. All activities are composed to form a single application. Each activity communicates with every other activity to send and receive the data between the activities. All activities interact with user and operate the flow in an application. Each time new activity starts, previous activity is stopped and pushed to back stack in the system. System calls callback method called ‘onCreate()’ when creating an activity. You must call ‘setContentView’ to define the layout for activity’s user interface. System calls ‘onPause()’ method when leaving current activity. All previous activities will be resumed when current activity is running. See Figure 3.2 for the application’s overview design with core components.

The three core components of application are activities, intents, services. Intents are used to pass the messages between various activities. An intent object will be passed to context.startActivity() to start new activity in the application and another intent object context.startService() is used to initiate a service to run any application in the background.

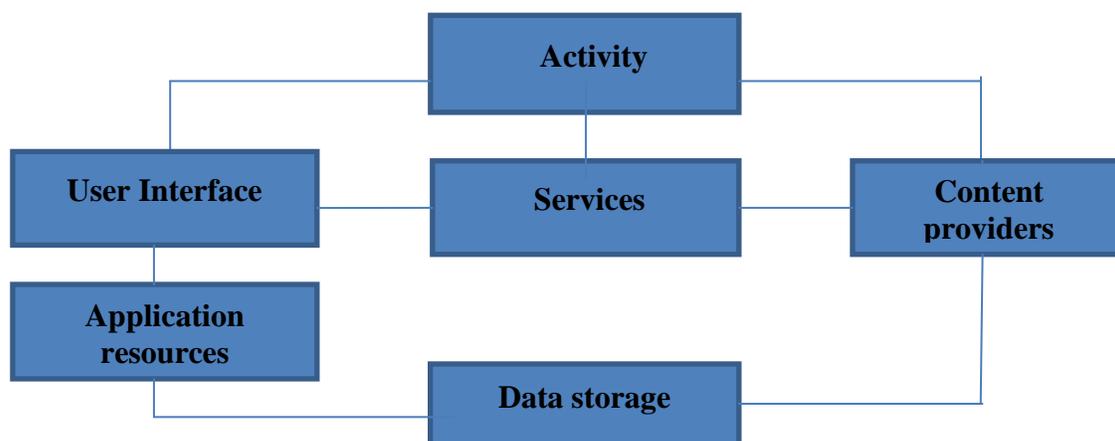


Figure 3.2. Application's overview design with core components.

Objects of `android.content.intent` are used to send/receive data between various activities in application. Every application has `AndroidManifest.xml` file which is in root directory. This file describes components of application such as intents, services, broadcast receivers and content providers. This file provides information about the components and classes used to implement an application. This lets android system know main components of an application.

3.3.2 User Interface

View & ViewGroup: User interface is built using `View` and `ViewGroup` objects. `View` class acts as base for subclasses called “widgets” which offers complete UI objects like text fields and buttons. `ViewGroup` acts as base for subclasses called “layouts” which offers different kinds of layouts like linear, relative, tab layouts etc. A `View` object is data structure contains layout parameters and layout content. `View` object handles own measurement layout, drawing, focus change, scrolling, and key/gesture interaction for rectangular screen. As an object in user interface, view is a point of interaction for the user and receiver of interaction events.

`ViewGroup` offers different layouts for android user interface. They are `LinearLayout`, `RelativeLayout`, `FrameLayout`, `TabLayout` etc. This application uses `LinearLayout` and `LinearLayout` with scroll view. `LinearLayout` uses `ListView` and buttons. `LinearLayout` with scroll view uses `TextView`, `EditText`, `List View`, spinner and buttons. This application uses android core calendar layout to synchronize event with mail accounts and alarm clock functionality used to notify user through alarm ring for all events.

View Hierarchy: Block diagram to show View Hierarchy with View and ViewGroup nodes on android platform (see Figure 3.3 [1]).

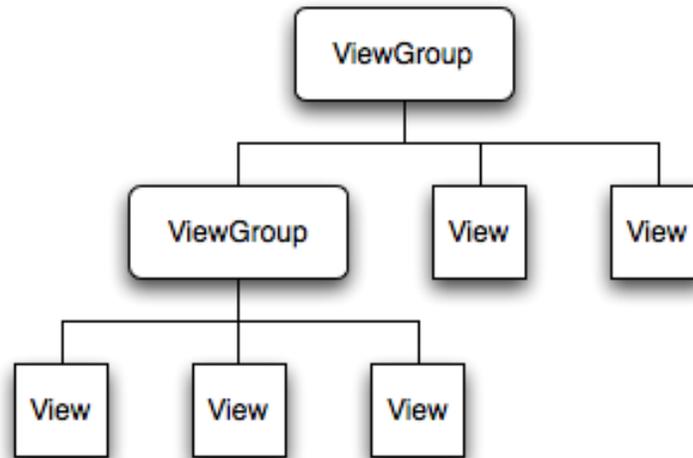


Figure 3.3. View hierarchy with View and ViewGroup classes. Source: ANDROID DEVELOPERS, *User interface. Android Developers*, <http://developer.android.com/guide/topics/ui/index.html>, accessed December 2011, n.d.

Activity in an application calls `setContentView()` call back method and pass a reference to root node of the object. Android system receives the object to measure and invalidate to build a view hierarchy.

ViewGroup node calls its child views and is responsible to allocate size and location for every View in an application. This application uses List views, Buttons, Menu's from Views and Linear Layout with scroll view from ViewGroup for different activities. Each Layout in android application is defined using XML file. In this application Layout XML file is designed using List View, Menu, Buttons, Spinner and Linear Layout (from ViewGroup) objects. The name of XML file is with respect to the java class which it represents. In Linear Layout contains both Button and Text view.

Widgets: Widgets are the view objects acts as an interface to interact with the user. Android platform provides set of fully implemented widgets such as Buttons, Check boxes, Text entry fields, Clock, Date picker, etc. These widgets are useful to quickly build user interface in an application. In this application, UI is built using some of android widgets like buttons, check boxes, clock, calendar, date picker and time picker. Widgets can be created by

extending View class. In this application, Android widgets are implemented using additional files like.

Java implementation file: This file implements behavior of the widget. An object should be instantiated from layout XML, and a constructor that retrieves all attribute values from Layout XML file.

XML definition file: An XML file in res/values/ which defines XML element used to instantiate the widget and attributes it supports.

Layout XML: Layout XML file is optional which is in res/layout/ that describes layout of the widget.

UI Events: Once after adding Views and Widgets we want to know the user interaction with them. There are two important things to be done with respect to UI events. Register event listener with view: The way to listen to events is by defining the listener and register with views on the user interface. This application uses listeners like View.OnClickListener which is for handling Item clicks on a View, View.OnTouchListener for handling touch screen events on a View and View.OnKeyListener for handling key presses on a view. To be notified when a button is clicked implement OnClickListener and define onClick() call back method and register it view as setOnClickListener(). Or other method is to override an existing callback method for the View.

Menus: Application menus play an important of application's user interface. Menus are structured using View hierarchy, In this application onCreateOptionsMenu() call back method is used on the activity to create options menu and include menu items that are needed for specific activity in an application. Menus handle their own events. When an item from menu is clicked, onOptionsItemSelected() method is called by the framework.

3.3.3 Services

Service is a component that runs in background and does not provide user interface. User can run service in the background by another application component while user switches to different applications. Component is bound to service to interact and even perform interprocess communication (IPC). Examples of services are Alarm notification that runs in the background etc. Service has two main components:

- Start service (): An application implements service in background by calling startService() method . It continues to run in background even if the component is destroyed that started the application. Service automatically stops itself when operation is done in background.
- Bind Service (): An application component binds the service by calling bindService(). Components of client server interface interact with service, send requests and get results. Other activities are bound to this service, makes it to run this service as long as they are bound to each other.

Service is base class for all services. This application uses Alarm Service which extends service class. Broadcast receiver class is used to initialize the Alarm Service which runs in the android service back ground of this application. Each activity will start calling service by startService() method and passing an Intent that specifies service and data for service to use. Finally service will be stopped by another activity in an application by calling stopService().

Intent describes about the operation to be performed in the background. It can be used in any activity, broadcast Intent requires to send to interested BroadcastReceiver components to communicate with background service. Primary components of Intent are action and data.

Broadcast Receiver is a class used to initiate Alarm service in this application from the package android.content.BroadcastReceiver. It will receive intents that passes through Service class sent by sendBroadcast().

3.3.4 Content Providers

Content providers are used to pass the data between various applications. It stores and retrieve data and makes accessible to all activities in an application or between various applications. All the content providers are available in android.provider package. Android platform is provided with some content providers like calendar, contacts information etc. In this application, Content providers that were used are calendar and Alarm clock to store and retrieve date and time. Alarm clock class is defined and it contains an intent which starts an activity to set new alarm in application and get the notifications. Alarm clock and calendar content providers are accessible throughout the application. All content providers implement a common interface for adding, altering and deleting data from the activity.

An interface through which applications get access to content model is through ContentResolver () objects. We get ContentResolver object by calling getContentResolver() within in an activity. ContentResolver cr = getContentResolver ().

Multiple ContentResolver objects are communicated through ContentProvider in different processes and applications. The communication between processes is handled by Content Resolver and ContentProvider class.

3.3.5 Application Resources

Application resources in an application are images and strings which we define for application code for building user interface. All the application resources for an application will be organized in res/ directory with sub directories that group resources by type and configurations. Application resources are of two types such as Default and Alternative resources. Default resources are the regardless of device configuration. Alternative resources are user designed for specific device configuration. Default UI layout is saved in res/layout directory. Appropriate resources will be assigned in resource directory by android platform based on device configuration.

Application resources to the application are images, strings. UI layout will be designed with respect to screen size or various strings. Once after initializing application resources we can access through resource IDs from R class of the project.

Resources should be placed like:

```
Project/  
Src/  
    Activity.java  
res/  
    drawable/  
        picture.png  
    layout/  
        main.xml  
    values/  
        strings.xml
```

All the resources will be declared under sub directories like drawable, layout, values resource res/ directory. So the actual resources are image, layout resources, strings which application can make use of. Task scheduler application uses all of the resources mentioned above for each activity

3.3.6 Data Storage

Data storage for android platform is to save application data. Data can be stored according to specific needs such as whether data is private with in application or accessible to all applications.

Data storage has various options such as shared preferences, internal storage, external storage, SQLite Databases and network connection.

Shared preferences provides general framework to save and retrieve primitive data types. Even the application is killed data will be still saved on user sessions. To get multiple preferences for application, `getSharedPreferences ()` should be used with first parameter, and for single preference file `getPreferences()` should be used. When application is compiled aapt generates R class and all the resources of this application will be stored in R class of res/ directory. Resource ID is used to retrieve the resources from the application.

Internal storage stores all the applications that are installed on the android system. This is private to application and other applications cannot access it. When user uninstalls an application from the android system, these files get removed from the device memory.

External storage stores data in device shared external storage. Android-compatible device has external storage where user can store files in form of removable storage media SD card. Users have read/ write permissions on the files that are stored on external storage and can be accessible by any user when they enable USB mass storage to transfer files on computer. This Task scheduler application can be saved in .apk format on external storage using SD card. So user will be able to run and access the application on android system.

SQLite Database is a major data storage component on android system. Android supports SQLite Databases stores the private databases created by user and accessible by name to any class in application. These databases are accessed within in database but not outside the application. SQLite Database stores the user created data in each activity of this application. For example, in this application tasks will be created and saved with date and

time details and other information related to the tasks. This is implemented by creating subclass to SQLiteOpenHelper and overriding onCreate () method to execute SQLite command in order to create tables in databases. All the data will be store in SQLite database and can be accessed only within in application. For reading and writing data to the application, we have to call getWritableDatabase() and getReadableDatabase (). Relational database that is used in application is used to create tables for storing all task details, task title, description, date and time.

CHAPTER 4

MODULES

4.1 OVERVIEW

Task scheduler application is a calendar based application that is designed for saving events based on priority. Here the user can create events that can store the tasks locally in android phone calendar and gets Email and Alarm Notifications. Task scheduler application is a user-friendly application. Each screen with layout description is provided in this chapter. All the layouts are controlled by the main java file that **extends** Activity class that implements onItemClickListener interface. Here the user interface is developed using many activities which are based on many user defined java classes. Figure 4.1 illustrates the flow of the application.

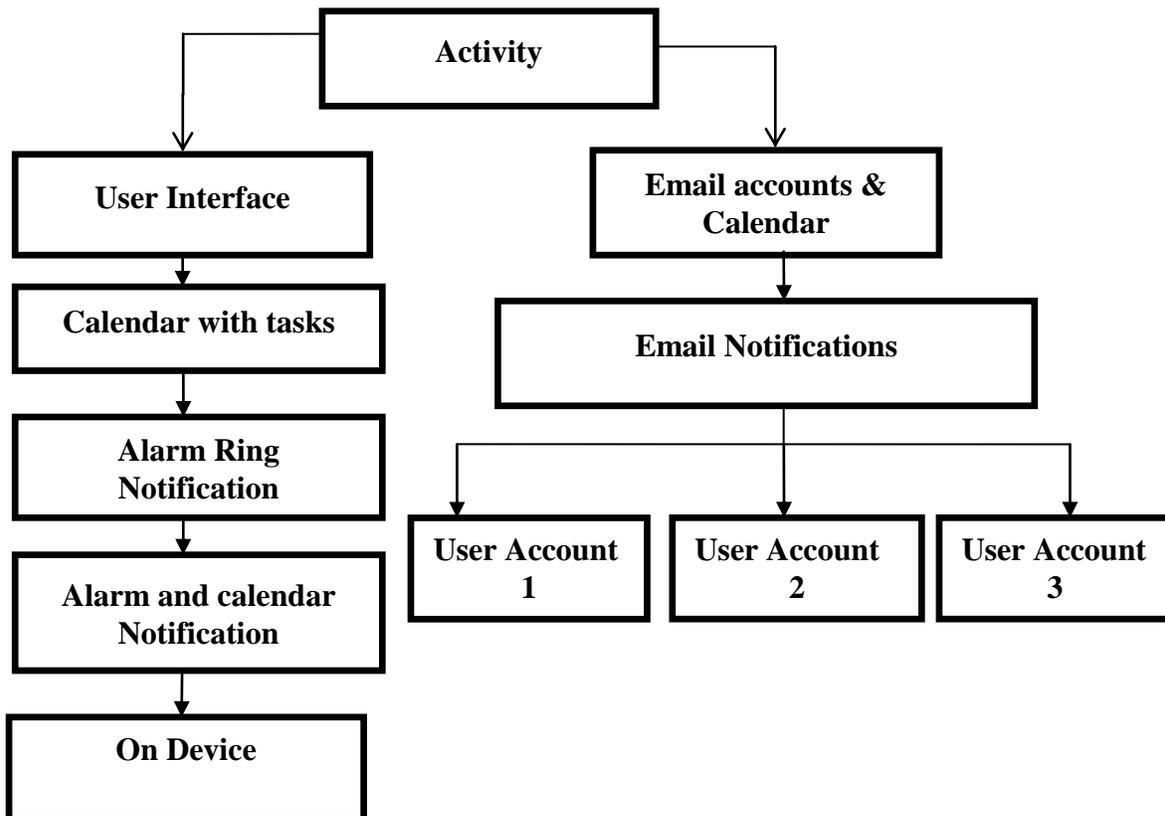


Figure 4.1. Displaying work flow of Taskscheduler application.

New-To-Do implements `OnItemClickListener` Interface: This interface calls the `OnItemClick()` method when an item in the list view is clicked. Implementers call `getItemAtPosition(position)` to access the data associated with selected task from the view.

4.2 APPLICATION HOME SCREEN – LAYOUT 1

Figure 4.2 shows the first layout which is designed with various buttons and calendar widget used in it. Each button upon clicking will redirect to its layout. “Enter a task” button redirects to layout that has the specific fields to provide all the details of the task like title, date, time, description, priority and select calendar fields. This layout will allow user to fill all the details of the task and create a task with user’s priority. “Show task summary” will redirect to the layout which displays all task summary of different priority levels. It will show each priority level and tasks assigned to it. User can have quick view of all priority level tasks. “Delete task” button redirects to the layout which displays the tasks which are to be deleted. The tasks that are marked as done will be listed under Delete task section. “Incomplete tasks” button will redirect to the layout where it displays incomplete tasks. This layout is implemented by `OnClick()` method. The layout has calendar widget which stores all the tasks along with their priority. The dates in the calendar will be highlighted with Red, Yellow, Green colors which represents the priority of the task. Upon clicking date it will display small dialog window with short description of the task. This layout is implemented by using `OnItemClickListener()` for each of buttons that are designed in layout.

4.2.1 Enter a Task – Layout 2

This layout contains 4 `TextViews` to show users the title of the edittexts and buttons. These `TextViews` have following text to ensure about the content to be typed. The `TextViews` are Task, Due Date, Description, Select priority and Select Calendar (see Figure 4.3). These are set as the title for each component. This layout allows the user to enter task details. The task details are title, date, time and description, priority and calendar selection of the task. This layout is designed with buttons `Edittext` and `TextView`. User can assign priority to the task whether it is High priority, medium priority, low priority or put the task under un prioritized section. User can select the type of calendar to send email notifications.

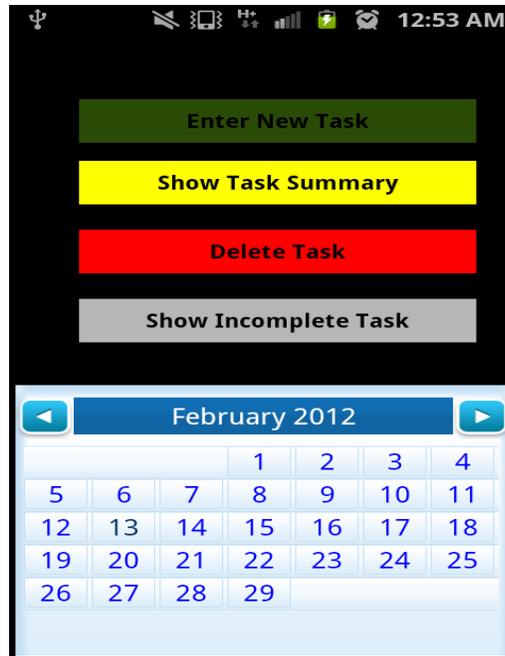


Figure 4.2. Initial home screen layout – Layout 1.

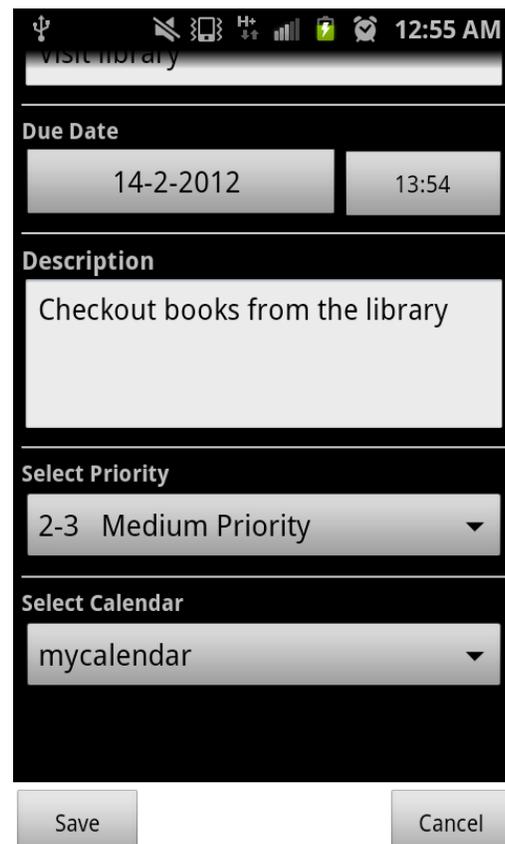


Figure 4.3. Enter a task – Layout 2.

This layout uses DatePicker() and Timepicker() widgets to select date and time and layout is implemented by OnClickListener(). Date button by default contains the text set with current date of the android phone. The date which is set as due date is viewed as setText() of the date button. The email accounts are listed in spinner by using ArrayAdapter. User can save and cancel the task at any time. The layout uses EditText class which configures text to be editable over the TextView. It uses methods Editable getText() to return the text the TextView is displaying and method setText() to set the text that TextView is to display. Based on the priority that is assigned to the task, tasks will be listed under its priority and which will store all the tasks under show task summary layout. Spinner() is used to select the calendar from the layout. The items that spinner holds comes from Adapter object which used getItem() method to get the item from the adapter. The attributes of the layout is declared in XML file under res/layout path and values are declared under res/values path. Spinner widget is captured from layout using findViewById(int). Spinner has implemented OnItemSelectedListener interface to notify when an item is selected.

4.2.2 Editing the Event – Layout 3

After an event is saved, the event is selected and it is taken to the edit the event. It shows the mandatory details of the saved event. Update button is clicked to update the event there the user is free to edit the complete details of the event.

The Update button is clicked to update the edited details of the selected event. The updated event is shown reflected in the Layout 1. Updating the selected event is done by using Intent class object. Intent is an object describes update action on the activity for the selected task.

The event is edited and updated using Cursor, Uri and Intent Class object. The Cursor class is used to make query for the updated calendar event values from android core calendar database that are fetched and saved in local database (using SQLiteDatabase query). Here, Uri is used to locate the calendar event from the android core calendar (similar to locating a folder's path in a disk). Uri uses getData() method to retrieve the data this intent operates on. Intent is used to pass the values into android core layout for updating the event. See Figure 4.4 for the edit event activity screen.

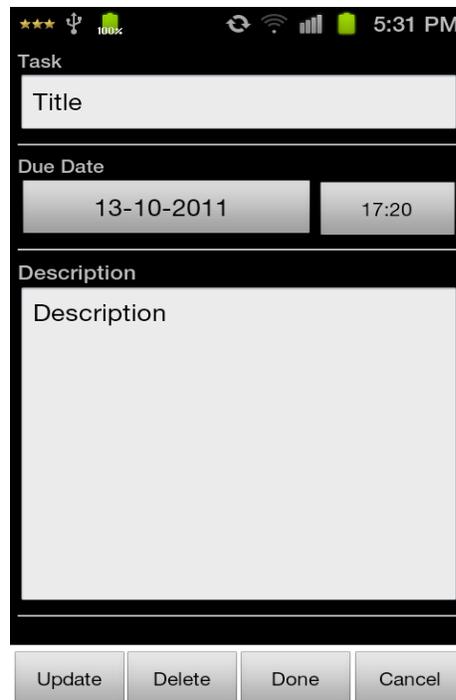


Figure 4.4. Edit event activity.

4.2.3 Deleting the Event

To delete the event, the event from Layout 1 is selected which takes the event to the Layout 2. As the event to be deleted is selected, the Delete button is clicked to delete the event from the list of priority events, as well as from the android core Calendar. Deleting the event is done by querying for deletion of the selected event from the android core calendar using ContentResolver class object. It uses `int delete()` method to delete the tasks from the list.

After the task is deleted from the calendar, the user is notified that the task is deleted successfully by using Toast notification. It invokes `setText()` method to update the text in the Toast that was previously created. See Figure 4.5 for the deleting a task screen.

4.2.4 Show Task Summary – Layout 4

This layout displays list of tasks with different priorities like High priority, Medium priority and low priority (see Figure 4.6). All the tasks that are created will list under these priority levels. Each priority is displayed using TextView and ListView is used to store the tasks. Every priority level is designated using Red, Green, Yellow and Blue colors. High priority uses red color text view, Medium priority uses yellow colored text view,

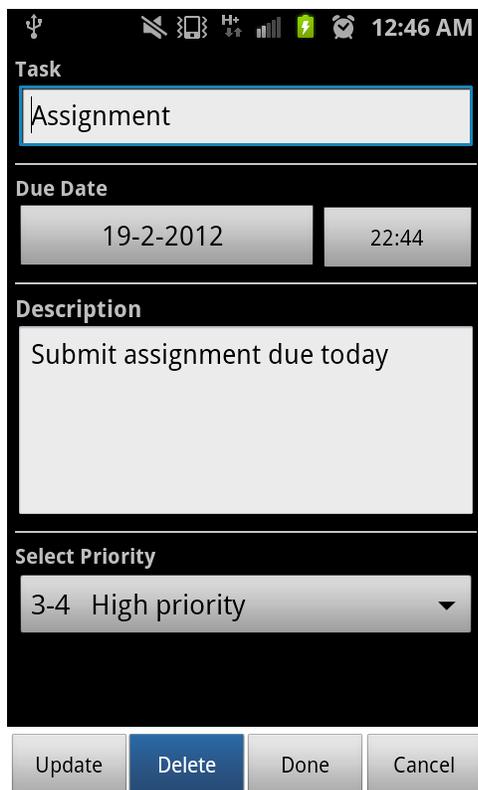


Figure 4.5. Deleting a task.

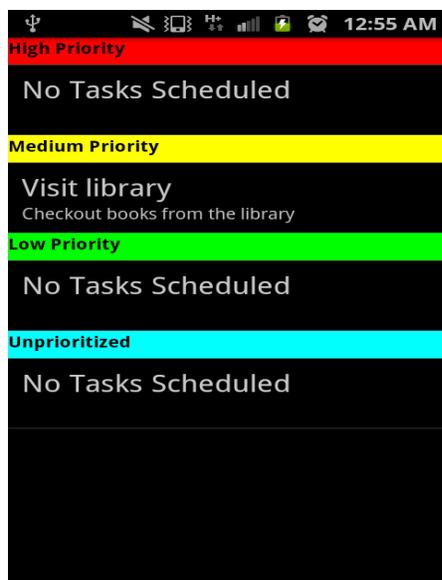


Figure 4.6. Show task summary – Layout 3.

Low priority uses green colored text view and un prioritized section uses blue colored text view. Every task is displayed with title and description with respect to its priority. User can assign the priority to the task with any date and time. Upon assigning the priority, tasks will be listed under its priority. DBHelper() is used to store the list of tasks under its priority. Four TextViews for representing priorities such as High Priority with red color, Medium Priority with Yellow color, Low Priority with Green color and un prioritized tasks with blue color. TextView class from android.widget.TextView package is used. This class displays text to the user such as High priority, Medium Priority, Low priority and Un prioritized text in the Layout. Each Priority TextView contains a ListView below it to represent the corresponding tasks/events that are saved within the respective priorities. Each ListView holds its priority based events in it. Therefore the High priority holds a ListView with the events that saved with high priority, Medium Priority holds a ListView with events saved with medium priority, Low priority holds a ListView with events saved with Low priority and tasks that have not been assigned any priority will come under un prioritized tasks. This activity uses ListView class from android.widget.ListView Package to insert tasks automatically to list using ListAdapter interface. All the Listview and TextView components will be implemented using onCreate() method.

4.2.5 Delete Tasks – Layout 5

This Layout has list of tasks to be deleted (see Figure 4.7). It is designed using text view and ListView() to store list of tasks that are to be deleted. This is a linear layout designed with text view to display the text and database to store the list of tasks to be deleted. Tasks that are marked as done will be listed under ‘Delete task’ layout. This layout is designed using database to store list of tasks to be deleted. The layout is implemented by onOptionsItemSelected() and onItemClick() methods.

It uses menu item called ‘clear completed’. User has to click on this menu item then it will clear all the tasks which are to be deleted from the list.

4.2.6 Incomplete Tasks – Layout 6

This layout contains list of incomplete tasks (see Figure 4.8). Tasks with expired date and time will be categorized under incomplete tasks. The layout contains ListView which stores list of incomplete tasks.

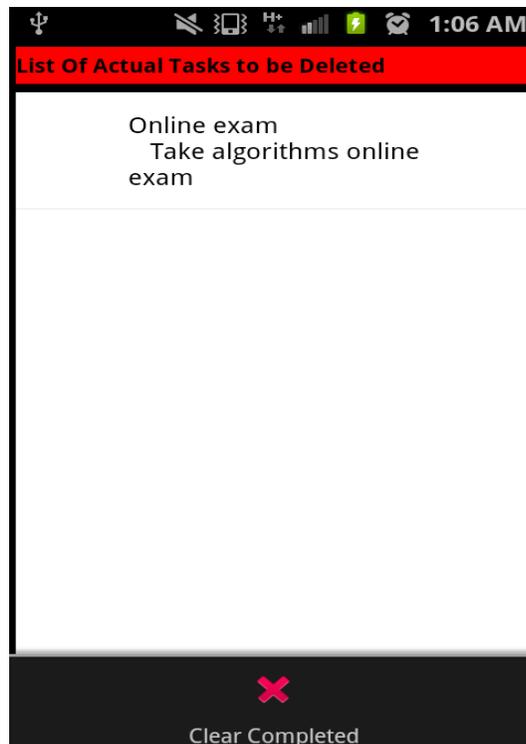


Figure 4.7. Delete task – Layout 5.



Figure 4.8. Incomplete tasks – Layout 6.

This layout is implemented by onItemClickListener() method, TextView and ListView. List of tasks will be stored in database called DBHelper. The layout contains ListView to store list of Incomplete tasks on the layout.

4.2.7 Notifications Shown

When the saved event time is reached, the calendar notification of the event is shown at the top of the screen. The calendar notification is a short term notification that notifies at the notification area (top) of the screen. The alarm is made to ring only 3 times using CountdownTimer. The Alarm notification using AlarmManager is shown along the **Calendar notification**. It uses method setTime() to schedule the time for task. For such event notification on the top of the layout, NotificationManager class object is used which invokes notify() method to post a notification to be shown in the status bar. This notification is shown even when the application is not running

Ringtone for alarm is set using RingtoneManager class object that used to set the default ringtone for the alarm ringing. Here MediaPlayerclass object is used to control the ringtone which must ring only three times when the event's time is reached. AudioManager is used to control over the streaming volume of the alarm and the audio service.

BroadcastReceiver class and IntentService class are extended to initiate the alarm along with the calendar event notification. For each event due time, the alarm is set using PendingIntent class object with varying unique id for the individual event of the Calendar.

For those events saved with e-mail notification, the event is saved in both "mycalendar" as well as the selected email (from the spinner) id's calendar. The email from the sender e-mail id (selected id from spinner) is sent to the recipient mail id (entered recipient e-mail id in EditText) as email notification.

CHAPTER 5

USER'S MANUAL

5.1 OVERVIEW

This chapter provides information about setting the development environment for an application such as information on getting started and installation of the application.

5.2 DEVELOPMENT ENVIRONMENT FOR ANDRIOD

There is a specific development environment set up for every android application to install on the device. The steps to set up development environment are given in the android developer website for developers to develop the application. This application can be developed on any platform with java and android software. The steps for setting development environment are:

- Installing the SDK.
- Install Eclipse IDE.
- Install Java Development Kit (JDK 5 or 6).
- Install ADT plugin for Eclipse for Android Development.
- Setting up SDK using Android SDK and AVD manager.

5.3 INSTALLING APPLICATION ON THE ANDRIOD DEVICE

Android applications are in .apk format. We can download .apk files from android market or download applications from unknown sources. Downloading applications from Android market is a direct approach it's just one click and run the application on the device. But for downloading applications from unknown sources that are not from android market, for these applications settings on the device should be changed. We can directly copy .apk file on SD card and enable the Application settings to run the application. Figure 5.1 illustrates the settings to change for running android application from unknown sources. For manual instructions following are the instructions to be followed.

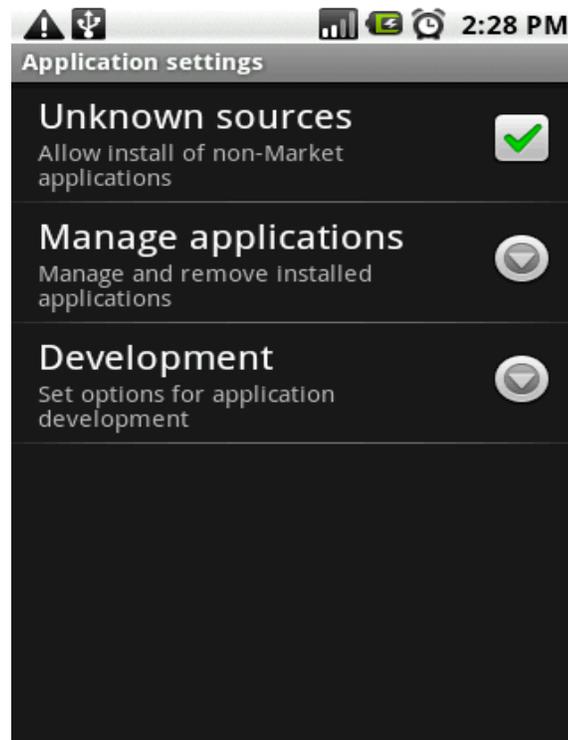


Figure 5.1. Installation of the android application on device from “unknown source.”

For manual instructions, the following steps should be followed:

- Check if you are in Home screen on android device.
- Step1: Click on Menu.
- Step2: Select settings from the Menu.
- Step3: Click on application.
- Step4: Check the option “Unknown Resources.”

After changing these settings, Application will be installed on the android device.

5.4 TASK SCHEDULER ON ANDROID DEVICE

Once you install the task scheduler on the android device. Initially when you run the application data base will be empty for the first time usage. The activities in the application can be loaded with data by user when scheduling the tasks. See Figure 5.2 for the initial layout which displays all buttons and calendar and creating a task with high priority on layout.

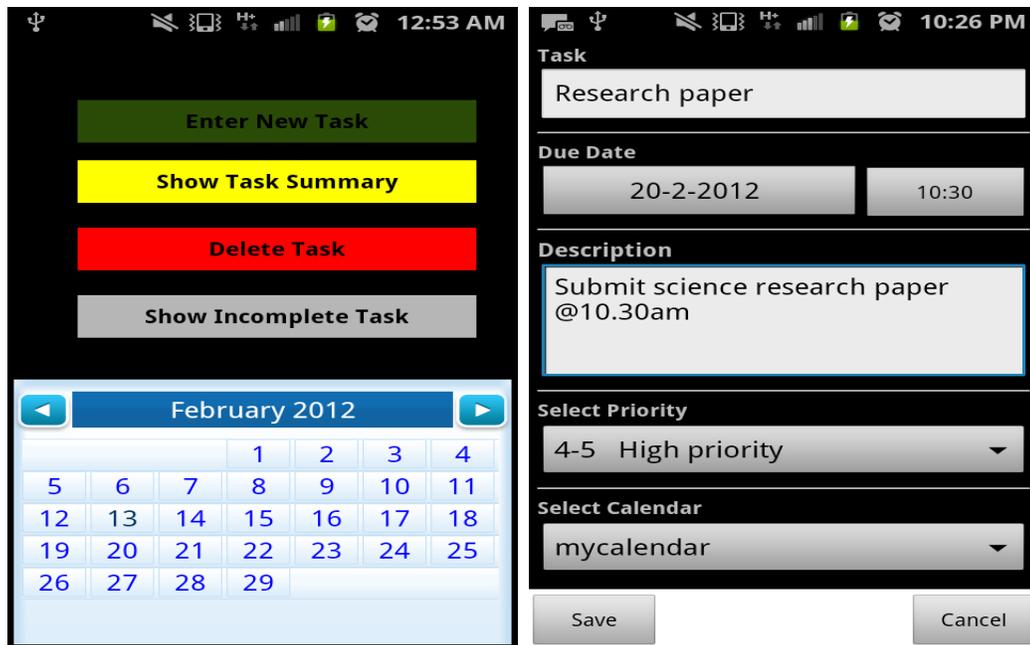


Figure 5.2. Initial layout displays all buttons and calendar (left) and creating a task with high priority on layout (right).

Steps to create a task:

1. Tap on “Enter New Task” button to create a Newtask and it will transfer the control to the next layout where you can provide task details.
2. Second layout is built with buttons such as Title, Due Date, Description, Select priority and select calendar options. You can enter all task details and click on ‘save’ button and it will be saved in respective priority according to date and time.

Once the task is created with title, date, time, description, priority and calendar option it will be listed in listview called show task summary layout. The tasks will be assigned priority upon user’s choice and preference to the task. With the priority listing user can keep of tasks that are to be completed first with priority. If you would like to edit the task with date, time and its description you can edit the existing tasks any time with its information. Layout consists of “selectcalendar” option and “select priority”. Select calendar option will display list of email accounts that are saved in device previously and email address to be given in which the notifications to be sent. If user wants to save the tasks in default calendar. User can select the my calendar from the spinner. “Save” button has to be clicked to save all the information for the tasks. If you want to cancel the events user can simply cancel it by using “cancel” button. Before selecting any email account calendar from the in mycalendar option. You have to set up and add the email accounts on the android

device. So this application fetch those email ids and show up in spinner. Then you can select these email ids and store the tasks.

Steps to setup and add Email accounts on android device - the following are instructions to be followed for setting mutple email accounts on the android device:

1. Select settings in android device.
2. Select accounts and sync.
3. Click “Add an account option.”
4. Enter your email id and password for gmail account etc.and configure your Gmail account on the android device.
5. Configured Gmail account will be fetched in to drop down menu in the mycalendar option.

5.4.1 Creating a Task

Figures 5.3 and 5.4 show how to create the task with different priority levels. Each task should be given title, date, time, description, priority and calendar. User can select priority for the task from the spinner where it shows High priority, medium priority, low priority or un prioritized.

The layouts show how the task should be created.

Figures 5.3 and 5.4 show the activities where the tasks created with high priority, medium and low priorities. Each activity shows how the user has to input all details for creating a task. Based on priority tasks will be listed under its respective priority. The tasks with expiry date and time will be reflected under “Incomplete tasks” list. Tasks that are marked as done will be reflected under “Delete tasks” layout.

5.4.2 Task Summary Layout

Once the tasks are created with all details, they will be listed under show task summary layout using Listview. There are 4 Listviews which holds different priority level tasks. User will have quick view of tasks whether it is high priority, medium priority, low priority or even un prioritized tasks. Each priority is assigned a specific color Red, Yellow, Green and Blue.

Figure 5.5 shows the list of tasks assigned different priority levels along with their colors. Upon the user’s preference to the specific task, task will be assigned a priority and will be listed under its category.

The figure consists of two side-by-side screenshots of a mobile application's task creation interface. Both screens show a top status bar with various icons and the time. The left screen is titled 'Task' and shows a task named 'Research paper'. Below the title, there are two input fields for the due date: '20-2-2012' and '10:30'. The description field contains the text 'Submit science research paper @10.30am'. Below the description, there are two dropdown menus: 'Select Priority' is set to '4-5 High priority' and 'Select Calendar' is set to 'mycalendar'. At the bottom, there are 'Save' and 'Cancel' buttons. The right screen is titled 'Library' and shows a task named 'Library'. Below the title, there are two input fields for the due date: '21-2-2012' and '11:30'. The description field contains the text 'Checkout books from library'. Below the description, there are two dropdown menus: 'Select Priority' is set to '2-3 Medium Priority' and 'Select Calendar' is set to 'thesiswrk1@gmail.com'. Below the calendar dropdown, there is an email input field containing 'bokka@rohan.sdsu.edu'. At the bottom, there are 'Save' and 'Cancel' buttons.

Figure 5.3. An activity showing task created (left) and an activity showing task created with created with high priority medium priority (right).

The figure shows a single screenshot of a mobile application's task creation interface. The screen is titled 'Professor appointment'. Below the title, there are two input fields for the due date: '23-2-2012' and '13:30'. The description field contains the text 'Meet professor to discuss thesis'. Below the description, there are two dropdown menus: 'Select Priority' is set to '1-2 Low priority' and 'Select Calendar' is set to 'thesiswrk1@gmail.com'. Below the calendar dropdown, there is an email input field containing 'thesiswork@yahoo.com'. At the bottom, there are 'Save' and 'Cancel' buttons.

Figure 5.4. An activity showing task created with low priority.

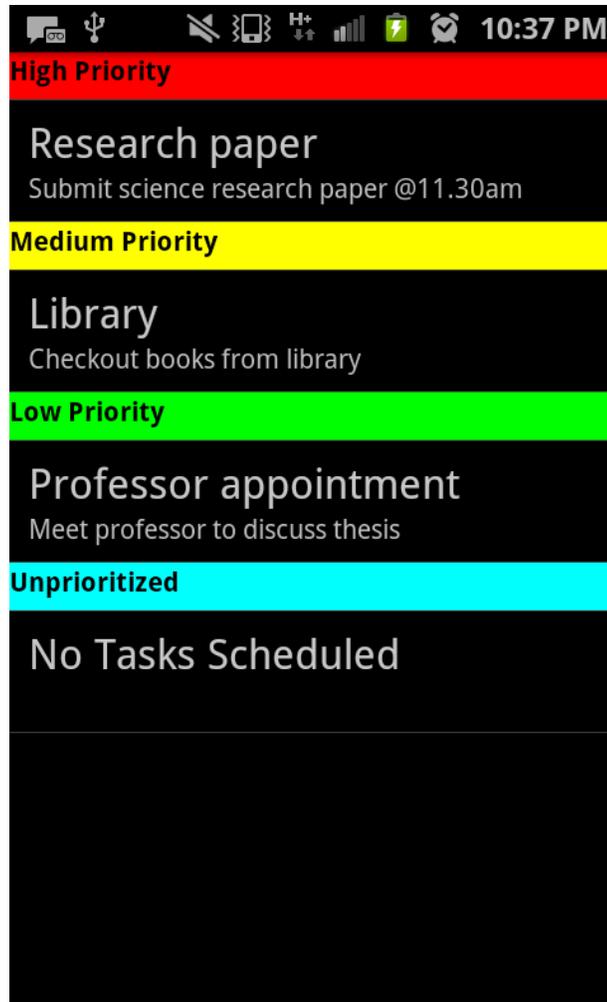


Figure 5.5. Show task summary layout.

5.4.3 Editing the Task

User can edit the task with respect to all task details like title, date, time, description or change the priority. All of these details will be updated with respect to change. By simply clicking on update button all the details that are modified will be updated in the layout.

The task that was created before was modified with time and description and there fore it will be updated by using update button. The event that is updated will be stored under show task summary layout. So user will have flexibility to update task any time with respect to its details. Once the task is updated with specific details it will show in task summary layout. (See Figure 5.6).

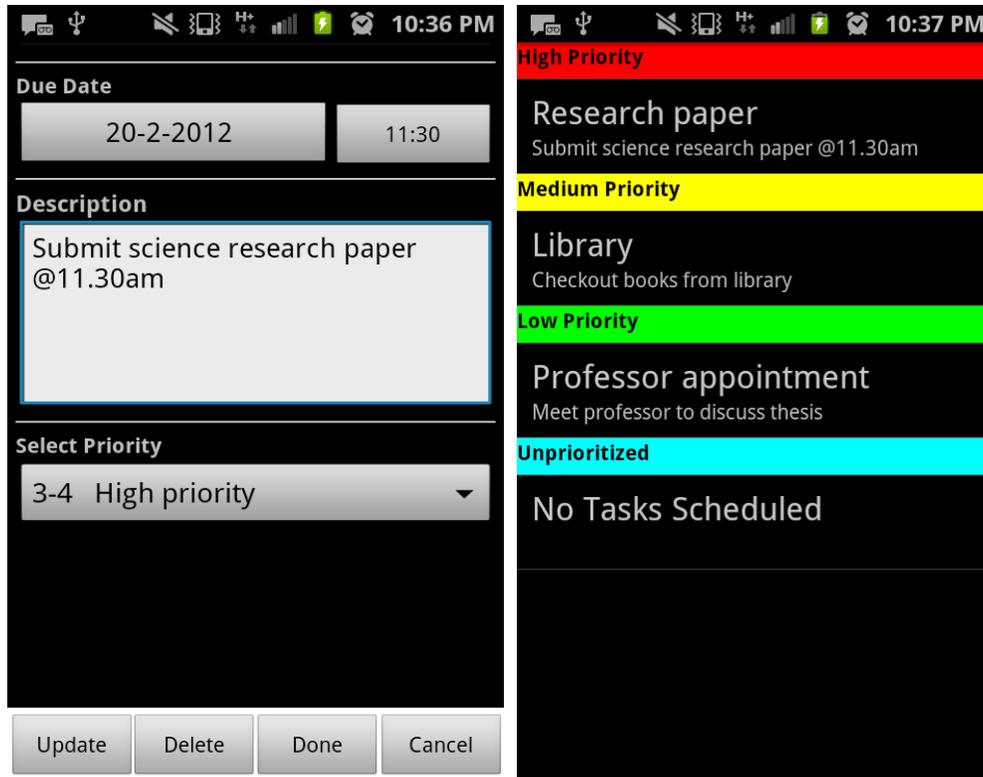


Figure 5.6. Edit a task layout (left), and updated task on task summary layout (right).

5.4.4 High, Medium and Low Priorities on Calendar

Once the task are created with specific details like date , time , priority and calendar, those tasks will be store in task summary layout to give quick view of tasks with various priority levels. User has to select the priority from the spinner whether it could be high,medium or low priorities. Once the user assigns priority to the tasks, those particular tasks will be listed under its priority in form of Listview on a layout. Sometimes user can leave it unassigned to the priority and put in unprioritized section. There is calendar in the layout where tasks that are created will be displayed in calendar using colors that each priority level has. Dates in which tasks are created will be highlighted with its color. If it is high priority it will be highlighted using Red color, If it is Medium priority particular date will be highlighted using Yellow color and if it is low priority it will be highlighted using Green color.

Figure 5.7 shows dates on calendar are highlighted with its specific colors of priority in which tasks are set to those dates as due date.

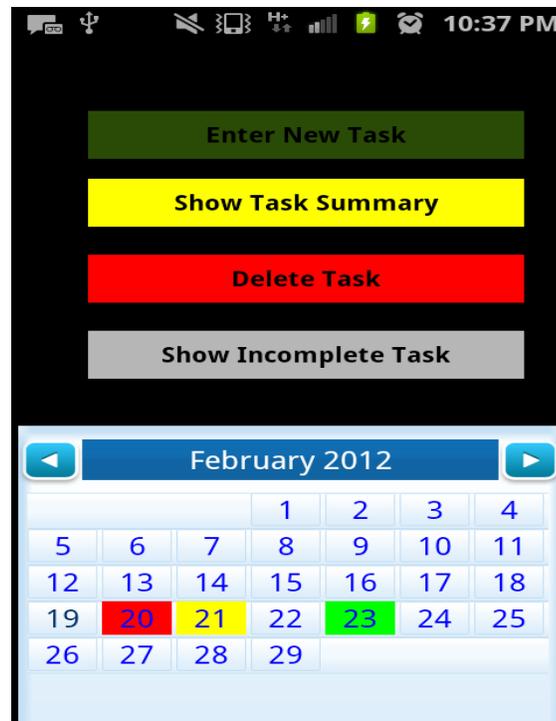


Figure 5.7. Layout showing high, medium, low priorities on calendar.

5.4.5 Delete Task

Tasks which are marked as done will be listed in Delete tasks layout. Those tasks are to be deleted to clear from the task summary layout. There is a Menu item Clear Completed Layout holding list of tasks to be deleted and designed with Menu Item called “Clear completed.” Clear completed menu item has to be clicked to clear those tasks from the list. This will give clear idea of what tasks are completed and cleared from the list. (See Figure 5.8).

5.4.6 Incomplete Tasks

The tasks with expired date and time will come under the category “Incomplete tasks” and will be listed in Incomplete tasks layout. This layout keeps hold of actual incomplete tasks in form of List. In that way, User can have clear idea of what tasks are still incomplete with expired date and time. This layout will keep track of expired tasks. And these tasks are supposed to be deleted using delete button of the particular task.

Figure 5.9 shows list of incomplete tasks in ListView. Tasks which are incomplete will be listed under this section.

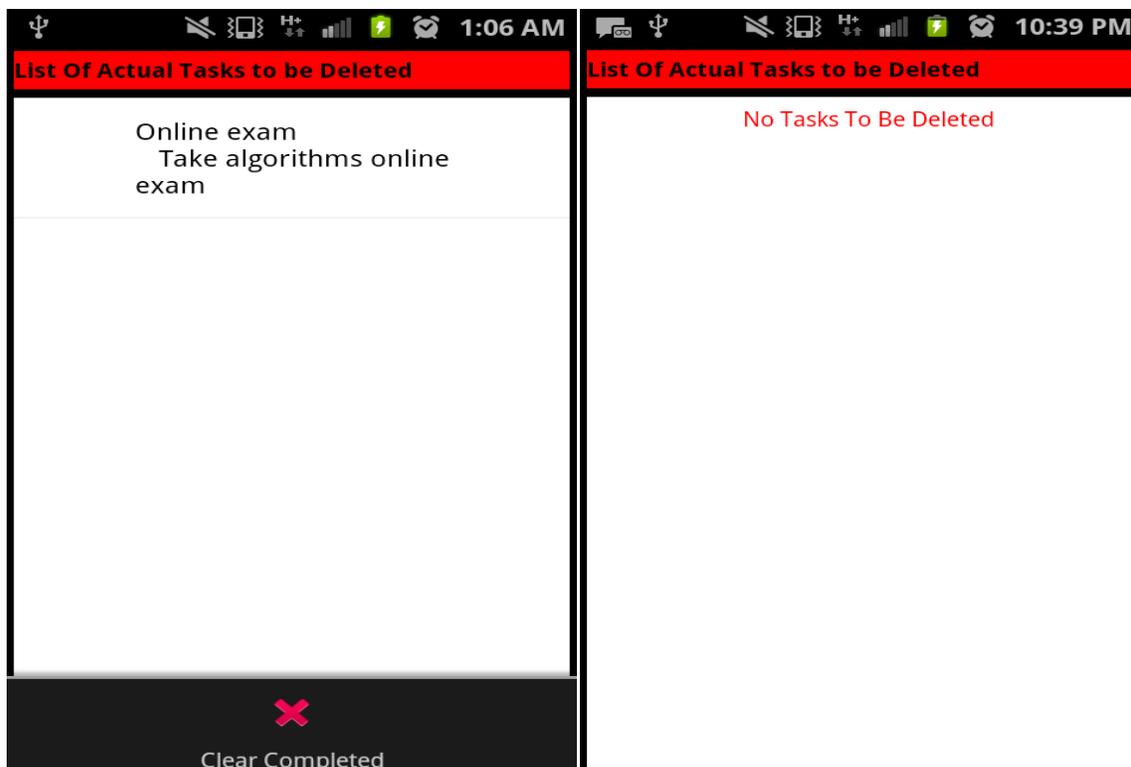


Figure 5.8. Delete tasks layout (left), and a layout showing no tasks to be deleted (right).

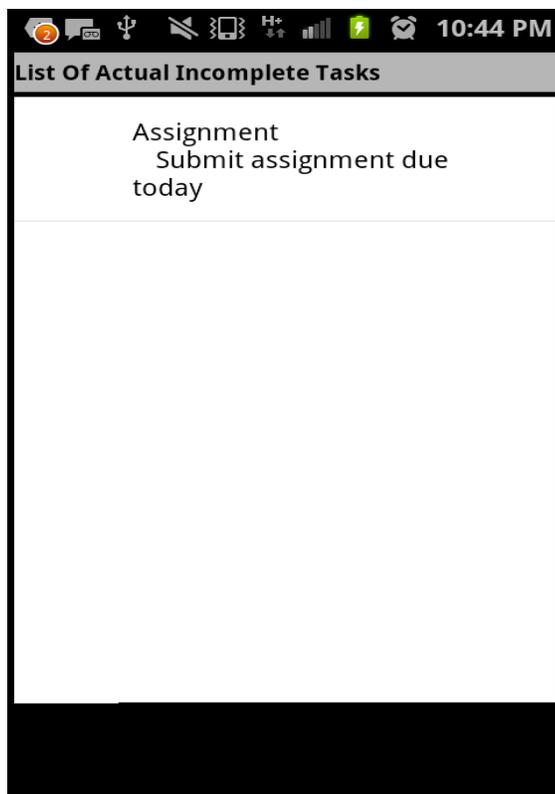


Figure 5.9. Incomplete tasks layout.

5.4.7 Notifications

In this application tasks are notified using Alarm and Email Notifications at their particular due date and time. There are two ways to receive notifications through this Android task scheduler application (1) Alarm notification, and (2) Email notification.

1. **Alarm notification:** User will be notified using Alarm notification at particular date and time for each and every task. Alarm will notify user for 3 times and shows notification on Notifications area of the android device. Along with notification, task title will be displayed during the Alarm notification.

Figure 5.10 shows the Alarm notification on the device and which also displays title of the task. User will be notified through alarm notifications for 3 times at particular due date and time.

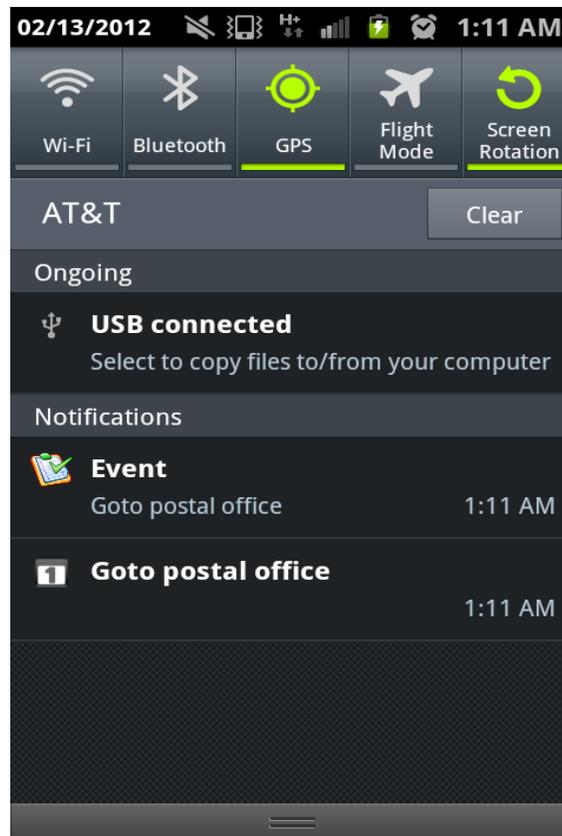


Figure 5.10. Alarm notification on Android device.

2. **Email notifications:** User has to select the calendar from the spinner. The email accounts that are synchronized on the phone will be listed in the spinner. In the calendar section, Mycalendar and email calendars will be displayed. Mycalendar stores the tasks on android calendar where as the email calendars will send an email notification for every task to notify user to complete the task on due date and time. When the email calendar is selected from the spinner, a recipient email id will be

enabled to enter the recipient email address. The recipient email address will receive notifications from the sender email account. This will work as invite or notification to receiver to notify on tasks

The email notification received on yahoo email account from the android device to notify for particular task. The sender email account should be gmail account and should be synchronized on the device. The recipient could be yahoo email account or email account of any organization. In this way, users receive email notifications for the tasks that have been provided calendar of email account. In this Android application, users will be notified by Alarm and Email notifications for the tasks. (See Figure 5.11).

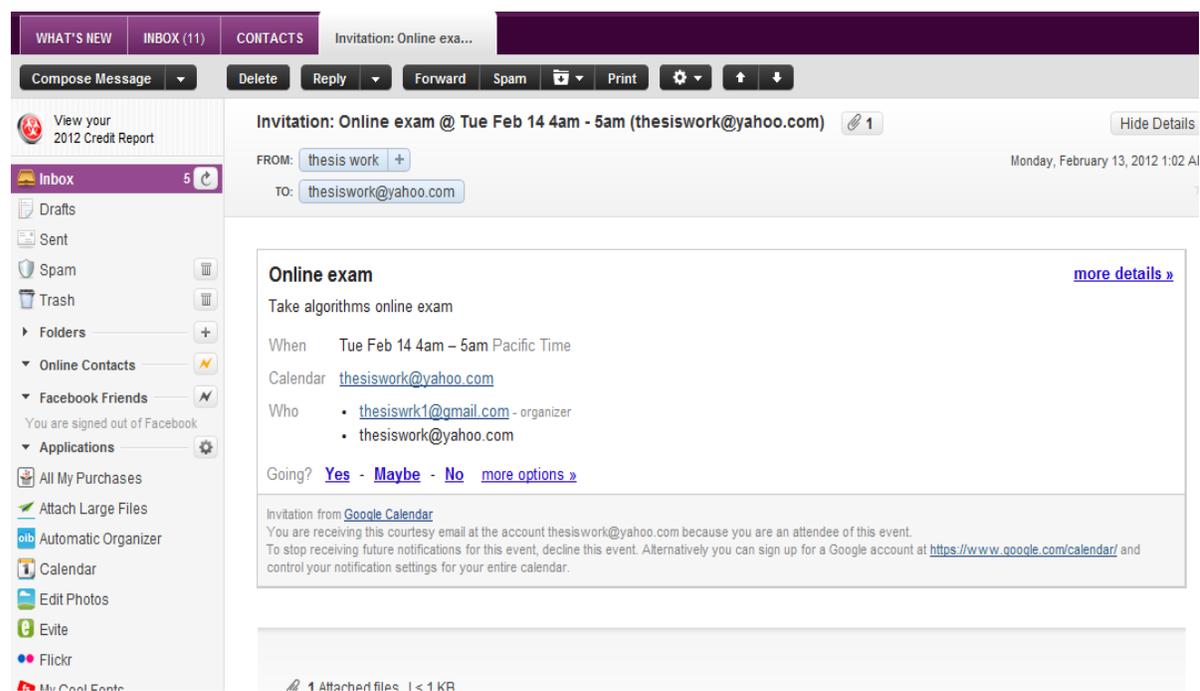


Figure 5.11. Email notification on yahoo account.

CHAPTER 6

CONCLUSION

6.1 FUTURE ENHANCEMENTS

This project is developed as a beginner application for creating tasks. It can be further developed to provide rich user interface. As this application can be developed to enable location aware services using Google maps for any task that is required to reach any location. This would help to find the particular location if you are required to visit.

When selecting the calendar from the application, it can be further developed to add one more section where user can add multiple email id's to invite or notify more people depending upon the task. So people can be notified along with the owner who sets up the meeting to invite more people. And it can be further developed to provide more features like settings themes and fonts for the application.

The current application supports 'Gmail' as 'sender id' which can send notifications to any user account, it can be further developed to give yahoo or corporate email accounts using their core API's for sending the notification to email. So the users are free to access any email account in the calendar section in senders email.

6.2 CONCLUSION

Main motto of the thesis is to build a simple and user friendly To-do list application for students and business people to schedule tasks and finish them on time with help of Quick way of multi-notifications. So the users will be benefited to get notified either through Alarm and email notifications. Application is built using Android SDK and Eclipse IDE as the development environment based on its design standards.

Task scheduler application makes uses of Calendar, Alarm services of the Android device to provide rich user interface and provides flexibility to the users to setup tasks on priority basis as per date and time, edit the tasks with details any time and notifies users through Alarm service and email notification. This application is very simple to use that displays all tasks priority wise and clears them after we finish.

Task Scheduler application has been tested on Android Samsung Galaxy S II, Motorola Bravo devices having Android 2.3.4.

6.3 FUTURE RESEARCH

This application can be developed with some additional features like to support multiple email ids under recipient section so that multiple users can receive notifications/reminder. This feature can be developed as an additional feature for this application.

The current application supports Gmail account as sender to send any notifications to any other user account. The limitation of the application is it doesn't support yahoo mail and hotmail as sender. I tried to work for Yahoo mail and hot mail as sender that can send notifications to various accounts and I had to find other way to work this. Because Yahoo and hotmail API's are Android OS dependent some of the features won't be available on certain Android SDK versions. I would find a solution for this in future, If Yahoo and hotmail API's are compatible with an advanced Android SDK such that a single Android SDK should support all these features that Yahoo and hotmail API's require.

BIBLIOGRAPHY

- [1] ANDROID DEVELOPERS, *User interface*. Android Developers, <http://developer.android.com/guide/topics/ui/index.html>, accessed December 2011, n.d.
- [2] ANDROID DEVELOPERS, *What is Android 2011?* Android Developers, <http://developer.android.com/guide/basics/what-is-android.html>, accessed October 2011, n.d.
- [3] ANDROID TAPP, *QuickList*. Android Tapp, <http://www.androidtapp.com/quicklist/>, accessed July 2011, n.d.
- [4] C. BOULTON, *Android market share to top 40% in 2011: IDC*. Eweek.com, <http://www.ciainsight.com/c/a/Latest-News/Android-to-Top-40-Market-Share-in-2011-IDC-664673/>, accessed September 2011, Jun. 2011.
- [5] M. BUTLER, *Android changing the mobile landscape*, IEEE Perv. Comp., 10 (2011), pp. 4-7.
- [6] A. CONSTANTINOU, *Mobile operating systems: The new generation*. VisionMobile, http://www.visionmobile.com/rsc/researchreports/Mobile_Operating_Systems_The_New_Generation.pdf, accessed June 2011, Sept. 2006.
- [7] A. GASIMOV, C. H. TAN, C. W. PHANG, AND J. SUTANTO, *Visiting mobile application development: What, how and where*, in Proceedings of the Ninth International Conference on Mobile Business, Athens, Greece, 2010.
- [8] K. PAUL AND T. K. KUNDU, *Android on mobile devices: An energy perspective*, in Proceedings of the Tenth IEEE International Conference on Computer and Information Technology, Bradford, England, 2010.
- [9] SLEETHERZ, *10 reasons why Android OS is able to displace Apple iOS*. Sleetherz, <http://www.sleetherz.com/android-news/10-reasons-why-android-os-is-able-to-displace-apple-ios/3819/>, accessed October 2011, Jun. 2011.
- [10] J. SNELL, *iPhone 3GS 8GB*. Macworld, <http://www.macworld.com/article/1151354/iphone3gs8gb.html>, accessed July 2011, Jun. 2010.
- [11] J. STICHBURY, *Introduction to Symbian OS*, John Wiley & Sons, Chichester, England, 2005.
- [12] WEBOPEDIA, *Introduction to mobile devices*. Webopedia, http://www.webopedia.com/quick_ref/mobile_OS, accessed June 2011, Sept. 2006.
- [13] WIKIPEDIA, *Android (operating system)*. Wikipedia, http://en.wikipedia.org/wiki/Android_%28operating_system%29, accessed August 2011, n.d.
- [14] WIKIPEDIA, *iOS*. Wikipedia, [http://en.wikipedia.org/wiki/IOS_\(Apple\)](http://en.wikipedia.org/wiki/IOS_(Apple)), accessed July 2011, n.d.

- [15] WIKIPEDIA, *Symbian*. Wikipedia, <http://en.wikipedia.org/wiki/Symbian>, accessed August 2011, n.d.
- [16] WIKIPEDIA, *Windows Mobile*. Wikipedia, http://en.wikipedia.org/wiki/Windows_Mobile, accessed August 2011, n.d.