ISSN:0975-9646

Change Profile with SMS on Android Platform

B. Lakshmi[1], B. Vijay Dattu[2] and K. Anji Reddy[3]

[1]*Asst. Professor, Department of Computer Applications,V.R.S.E.C, Vijayawada -7, Andhra Pradesh, India*
[2]*Pursuing MCA in V.R.Siddhartha Engineering College,Kanuru, Vijayawada 520010, Andhra Pradesh, India*
[3.]*Sr. Asst. Professor, Department of computer Applications,V.R.S.E.C, Vijayawada-7, Andhra Pradesh, India*

Abstract **- The Change Profile with sms is nothing but it is android platform based application runs in the android mobile or in the android emulator at the development time. The purpose of the application is to change the mobile phone profile by sending sms. The Change Profile with sms is based on the sms to change profile as silent to ringing & find by playing a audio file. This application enables or disables this feature on your choice. This application is being built on Android 2.2. This is built by the assistance of Eclipse IDE with ADT plug-in configured with android SDK for windows The android is the platform (mobile os) is now a day is mostly used in the 60% of mobile devices. This is built by the assistance of Eclipse IDE with ADT plug-in configured with android SDK for windows The android is the platform (mobile os) is now a day is mostly used in the 60% of mobile devices. It is more flexible to develop the application and test it on mobile or on emulator itself before deploying in to the mobile. This is the first version of its kind and is developed based on Android platform. The goal of this paper is used to change the profile automatically after some specific time.**

Index Terms – Change Profile with SMS, Analysis, Architecture and Work Flow

1. INTRODUCTION

The purpose of this paper is to list down requirements for the **Profile Change** application. The goal is to find mobile when it is silent and forgot where the user put it. The current system only has no way to find out the silent profile mobile with an SMS. With the help of SMS user can send an SMS to his/her mobile which gone went in silent and forgot.

1.1 Organization of the Paper

The Profile Change feature is a vital layer of the ANDROID framework for combining many of the lower level ADROID component's tasks to find out the mobile. The application needs to receive the SMS send by the user from other mobile. The scope of the system is to develop a Profile Change application that runs on Android platform. This is the first version of its kind and is developed based on Android platform. The goal is to find mobile when it is silent and forgot where it is placed.

2. OVERVIEW OF ANDROID

Profile Change is an Android based application to find the mobile when it is misplaced in silent mode and need to find out with some info like playing some sound and changing the profile from silent to normal with an SMS. To provide flexibility to the users, the interfaces have been developed that are accessible through a browser. The GUI'S at the top level have been categorized as

1. Administrative user interface
2. The operational or generic user interface

The 'administrative user interface' concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. These interfaces help the administrators with all the transactional states like Data insertion, Data deletion and Date updating along with the extensive data search capabilities.

The 'operational or generic user interface' helps the end users of the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information in a customized manner as per the included flexibilities

2.1 Input & Output Representation

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

Input Stages

The main input stages can be listed as below:

- Data recording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

Input Types

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.
- Operational, which are computer department's communications to the system?
- Interactive, which are inputs entered during a dialogue.

Input Media

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

- Type of input
- Flexibility of format
- Speed
- Accuracy

- Verification methods
- Rejection rates
- Ease of correction
- Storage and handling requirements
- Security
- Easy to use
- Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

Output Design

External Outputs whose destination is outside the organization. Internal Outputs whose destination is within organization and they are the User's main interface with the computer. Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs

- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly with the system.

Output Definition

The outputs should be defined in terms of the following points:

- Type of the output
- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

Output Media

In the next stage it is to be decided that which medium is the most appropriate for the output. The main considerations when deciding about the output media are:

- The suitability for the device to the particular application.
- The need for a hard copy.
- The response time required.
- The location of the users
- The software and hardware available.

Keeping in view the above description the application is to have outputs mainly coming under the category of internal outputs.

The outputs were needed to be generated as a hard copy and as well as queries to be viewed on the screen. Keeping in view these outputs, the format for the output is taken from the outputs, which are currently being obtained after manual processing. The standard printer is to be used as output media for hard copies.

2.2 Architecture

The architecture diagram represents mainly flow of requests from users to database through servers. In this scenario overall system is designed in three tires separately using three layers called presentation layer, business logic layer and data link layer. This application was developed using 3-tier architecture.

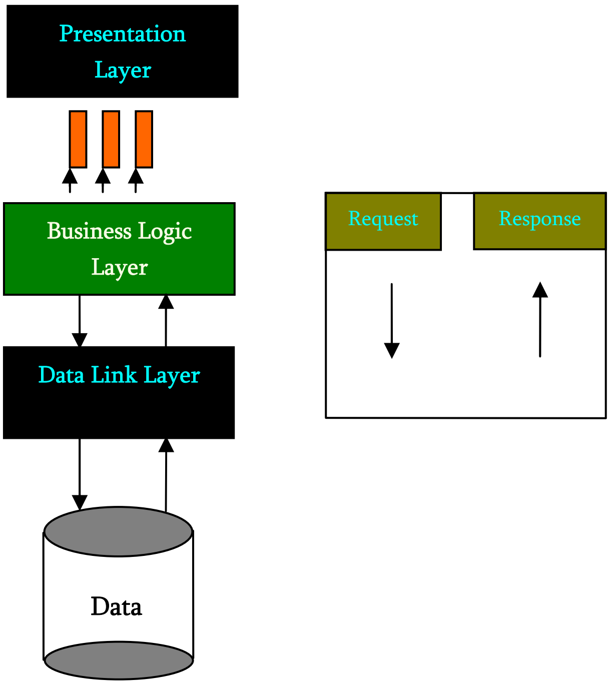


Fig: 1. Architecture Diagram

3. Feasibility Study

All applications are feasible if they have unlimited resources and infinite time. But the development of software is plagued by the scarcity of resources and difficult delivery rates. It is necessary and prudent to evaluate the feasibility of a system at the earliest possible times.

3.1 Economic Feasibility:

This procedure is to determine the benefits and savings that are expected from the candidate system and compare with cost. The implementation of this application can be done with the help of open source environments.

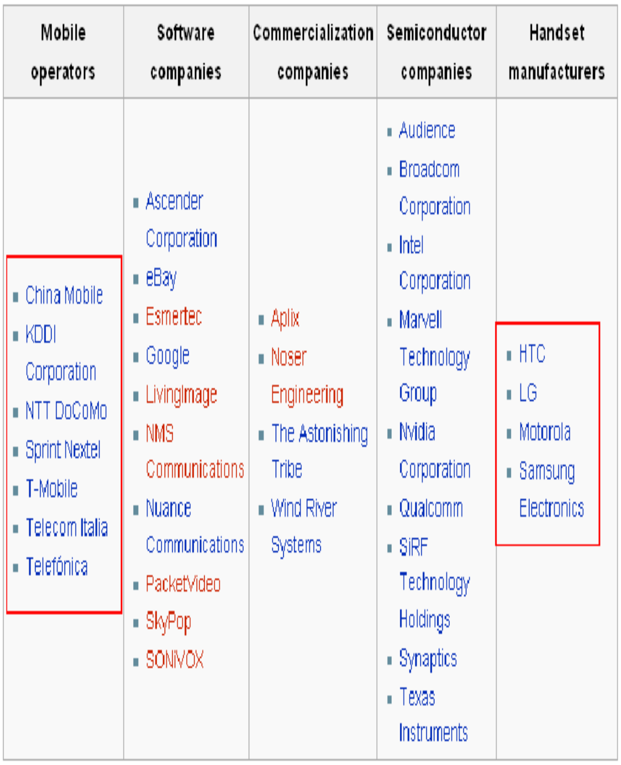
3.2 Technical Feasibility

Technical feasibility centers on the existing mobile system and to what extent it support the proposed addition if the budget is serious constraint, then the application is judged not feasible.

3.3 Operational Feasibility

People are inherently resistant to change and mobiles have been known to facilitate change. In this paper it was explained that a technical person requires configuring the software and technical background is necessary to work on the sensors.

3.4 Intended Developers



Mobile operators	Software companies	Commercialization companies	Semiconductor companies	Handset manufacturers
China Mobile KDDI Corporation NTT DoCoMo Sprint Nextel T-Mobile Telecom Italia Telefónica	Ascender Corporation eBay Esmertec Google LivingImage NMS Communications Nuance Communications PacketVideo SkyPop SONiVOX	Aplix Noser Engineering The Astonishing Tribe Wind River Systems	Audience Broadcom Corporation Intel Corporation Marvell Technology Group Nvidia Corporation Qualcomm SiRF Technology Holdings Synaptics Texas Instruments	HTC LG Motorola Samsung Electronics

Fig: 2. Intended Developers

This research is on latest open source mobile development platform called Android and to implement a game based technologies for different requirements.

3.6 Assumptions and Dependencies

Having only hardware is not sufficient, to access an application Software is must. Assumption is made in such a way that the mobile is charged with enough battery. The battery should be in working mode. The one who is using the mobile must have a minimum knowledge how to create an account in the buddy locator and send request to another buddy and add buddy in the account.

3.7 Android Development Environments

It includes a devices emulator, tools for debugging, memory and performance profiling, a plug-in for the Eclipse IDE. There are a number of the Hardware dependent features, for instance, a huge media and connection supports, GPS, improved supports for Camera and simply GSM telephony. A great work was done for the developers to start work with Android using device emulator, tools for debugging for Eclipse IDE.

4. Architecture

The following diagram shows the major components of the android operating system. Each section is described in more details below

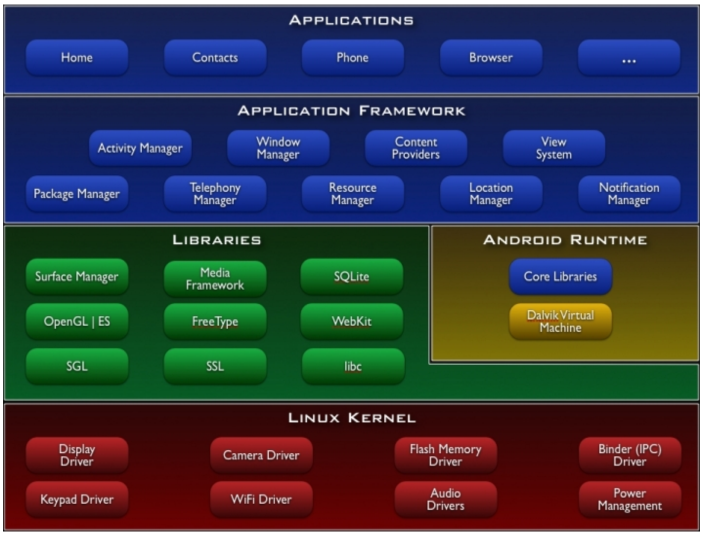


Fig: 3.Architecture of Android

4.1 Linux Kernel

Android Architecture is based on Linux 2.6 kernel. It helps to manage security, process management, network stack and other important issues. Therefore, the user should bring Linux in his mobile devices as the main operating system and install all the drivers required in order to run it.

Android provides system and install the drivers required in order to run it. Android provides the supports for the Qualcomm MSM7K chipset, but in the second half of the 20085 we should see mobile devices with stable versions Qualcomm MSM 7200, which includes major features.

- WCDMA/HUSPA and EGPRS network supports
- Bluetooth 1.2 and Wi-Fi supports
- Digital audio supports for mp3 and other formats
- Supports for Linux and other third party operating systems
- Java Hardware acceleration and supports for Java applications
- Qcamera up to 6.0 mpx
- GPS-One-solution for GPS

4.2 Libraries

In the next level there are set native libraries written in c/ c++, which are responsible for stable performance of various components. For example, surface managers are responsible for composing different drawing surface on the mobile screen. It manages the access for different process to compose 2D & 3D graphics layers. OpenGl ES and S GL make a core of graphics libraries and are used accordingly for 2D & 3D hardware acceleration.

Moreover, it is possible to use 2D and 3D graphics in the same Android application. The media framework was provided by Packet Video, one of the members of OHA. It gives libraries for a playback and recording support for all the major media and static image files. Free Type libraries are used to render all the bitmap and Vector Fonts.

For data storage, Android uses SQLite. As mentioned before, it is extra light relational management system, which locates a single file for all operation related to database. WebKit, the same browser by Apple's Safari, was modified by Android in order to fit better in small size screen.

4.3 Android Runtime

At the same level there is Android Runtime, where the main component Dalvik Virtual Machine is located. It was designed specifically for Android running in limited environment, where the limited battery, CPU, memory and data storage are the main issues. Android gives an integrated tool "dx", which converts generated byte code from .jar to .dex file, after this byte code becomes much more efficient to run on the small processors.

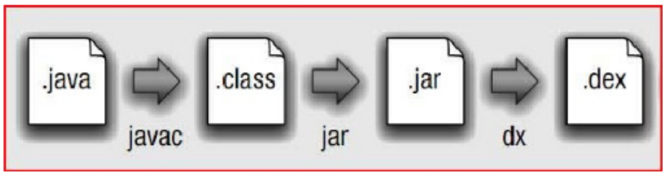


Fig: 4. Conversion from .java to .dex file

5. Developing Applications

5.1 Application Building Blocks

Android application is a collection of components; of various kinds. These components are for the most part quite loosely coupled, to the degree where you can accurately describe them as a federation of components rather than a single cohesive application.

Generally, these components all run in the same system process. It's possible (and quite common) to create multiple threads within that process, and it's also possible to create completely separate child processes if you need to. Such cases are uncommon, because Android tries very hard to make processes transparent to your code.

Google provides three versions of SDK for Windows, Mac OSX and one for Linux. The developer can use Android plug-in for Eclipse IDE or other IDEs such as intelliJ. First step for Android developer is to decompose the application into the components, supported by the platform. The major building blocks are these:

- Activity
- Intent Receiver
- Service
- Content Provider

Activity

User interface component corresponds to one screen at time. It means that for the simple application like Address Book, the developer should have one activity for displaying contacts, another activity component for displaying more detailed information of chosen name and etc.

Intent Receiver

It wakes up a predefined action through the external event. For example, for the application like Email Inbox, the developer should have intent receiver and register his code through XML to wake up an alarm notification, when the user receives email.

Service

It is a task, which is done in the background. It means that the user can start an application from the activity window and keep the service work, while browsing other applications. For instance, he can browse Google Maps application while holding a call or listening while browsing other application.

Content Provider

A component, while allows a sharing of the data with other process and applications. It is the way to communicate the between each other. Android will ship with a set of core application including an email client. SMS program, calendar maps, browser, contacts and others. All applications are using the Java Programming language.

5.2 AndroidManifest.xml

An AndroidManifest.xml file is tells the system what to do with all the top-level components (specifically activities, services, intent receivers and content providers' described below) you've created. For instance, this is the "glue" that actually specifies which Intents your Activities receive.

A developer should predefine and list all components, which he want to use in specifies AndroidManifest.xml file. It is requires file for all the application and is located in the root folder. It is possible to specify all global values for the package, all the components and its classes used, intent filters, which describes where and when the certain activity should be start, permissions and instruction like security control and testing.

Here is an example of AndroidManifest.xml file

```
        <?xml version="1.0" encoding="utf-8"?>
  <manifest
xmlns:android=http://scheme.android.com/apk/res/android
package="dk.mdev.android.hello">
<application android:icon="@drawable/icon">
<activity class=".HelloAndroid"
android:label="@string/app_name">
<intent-filter>
<action android:values="android.intent.action.MAIN"/>
<category
amdroid:value="android.intent.category.LAUNCHER">
</intent-filter>
</activity>
</application>
</manifest>
```

The line 2 is a namespace declaration, which makes a standard Android attribute available for that application. There <application> elements, where the developer specifies all application level components and its properties used by the package. Activity class in the line represents the initial screen the user see and it any have one or more <intent-filter> elements to describe the action that activity supports.

5.3 Application Lifecycle

In Android, every application runs in its own process, which gives better performance in security, protected and other benefits. Therefore, Android is responsible to run and shut down correctly these process when it is needed.

It is important that application developers understand how different application components (in the particular Activity, services and BroadCastRecіver) impact the lifetime of the application's process. Not using these components correctly can result in the system killing the application's process while it is doing important work.

To determine which process should be killed when low on memory? Android places each process into an "importance hierarchy" based on the components running in them and state of these components. These process types are (in order of importance).

1. A **foreground process** is one that is required for what the user is currently doing. Various application components can cause its containing process to be considered foreground in different ways. A process is considered to be in the foreground if any of the following condition holds.
   - (i) It is running an Activity at the top of the screen that the user is interacting with (its onResume() method has been called).
   - (ii) It has a BroadCastRecіver that is currently running (its BroadCastRecіver.onRecive() method is executing).
   - (iii) It has a service that is currently executing code of one of its call callbacks (Services.onCreate(), Service.onStart(), or Service.onDestroy()).

2. A **visible process** is one holding an Activity that is visible to the user on-screen but not in the foreground (its onPause() method has been called).This may occur, for example, if the foreground Activity is displayed as dialog that allows the previous Activity to be seen behind it. Such a process is considered extremely important and will not be killed unless doing so is required to keep al foreground processes running.

3. A **service process** is one holding a Services that has a process started with the startService() method. Through these process are not directly visible to the user, they are generally doing things that user cares about(such as background mp3 playback network data upload or download). So the system will always keep such process running there is not enough memory to retain foreground and visible process.

4. A **background process** is one holding is one holding an Activity that is not currently visible to the user(its onStop() has been called). These process have no direct impact on the user experience. Provided they implements their Activity life-cycle correctly(see Activity for more details), the system can kill such process at any time to reclaim memory for one of the three previous processes types. Usually there are many of these processes are running, so they are kept in an LRU list to ensure the process that was most recently seen by the user is last to be killed when running low on memory.

5. An **Empty process** is one that doesn't hold any active application components. The only reason to keep such a process around is a cache to improve startup time the next time a components of its application needs to run. As such, the system will often kill process in order to balance overall system resources between these empty cached and the underlying kernel caches.

6. DEPLOYMENT

A set of nodes and their relationships are clearly shown in the diagram. These nodes are physical entities where the components are deployed. This diagram is used for visualizing deployment view of a system.

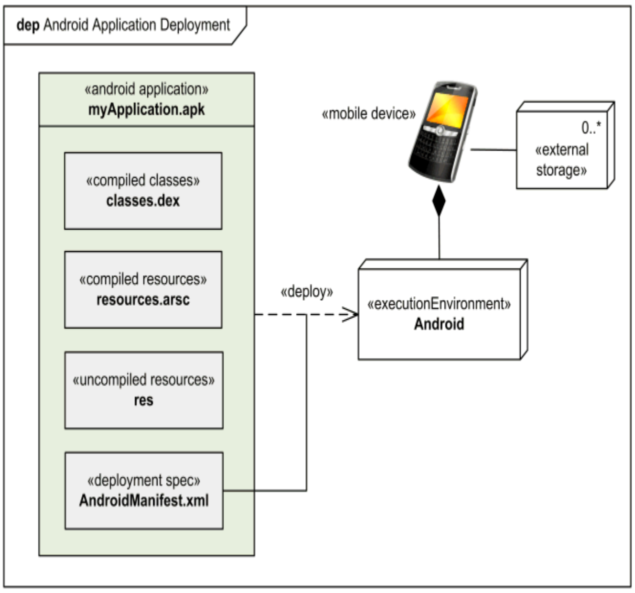


Fig: 5. Android Deployment

7. CONCLUSION

This is an easy way of finding the mobile when it is silent and it can change the mode easily by sending SMS. This application will works only on Android OS mobiles. With the vigorous development through Android, mobile Applications have been widely used on the various mobile Devices. Android mobile applications are evolving at a meteor Pace to give a rich and fast user experience. The maturity of The hardware and software platforms of mobile devices and The promotion of the Mobile Internet have brought a great Opportunity to the migration of the web applications to mobile
platforms. In case of the security, Static analysis scans the Software for malicious patterns without installing it.

Dynamic Analysis executes the application in a fully isolated Environment, i.e. sandbox, which intervenes and logs low level interactions with the system for further analysis. Both The sandbox and the detection algorithms can be deployed in The cloud, providing a fast and distributed detection of Suspicious software in a mobile software store akin to Google's Android Market.

The ultimate goal is to protect the Mobile applications from the malicious attributes and Safeguard the interests of Android mobile users. With the Mobile capabilities, the Internet connection capabilities and complete software platforms available, the future of mobile Web applications appear limitless.

REFERENCES

[1] http://developer.android.com/guide/basics/what-is- android.html

[2] 3G Mobile Terminal Development Trend of the operating system [M/OL] http://pda.c114.net/32/c4948.html, 2007

[3] Android Architecture 2010[R/OL]. http://www.cnmsdn.com/html/201003/1268713218ID2058_2 .html

[4] Static detection of malicious code in executable programs by J. Bergeron, M. Debbabi, J. Desharnais, M. M. Erhioui, Y. Lavoie, and N. Tawbi.

[5] Android Official Website (2008)—"Android | Official Website", <http://www.android.com/>

[6] An Android Application Sandbox System for Suspicious Software Detection, by Thomas Bl¨asing, Leonid Batyuk, Aubrey-Derrick Schmidt, Seyit Ahmet Camtepe, and Sahin Albayrak

[7] www.blackhat.com

AUTHORS' BIOGRAPHY

Mrs. B.Lakshmi is currently working as an Asst. Professor, Department of Computer Applications, VRSEC (Autonomous), Vijayawada, Andhra Pradesh. She has 8 years of teaching experience. Her areas of interest include Database Management Systems, Operating Systems. She had received M.C.A from Acharya Nagarjuna University. She has ratified under Nagarjuna and JNTUK, Kakinada. She had completed the OCA certification.

Mr.Vijay Dattu.B is currently working as an Asst. ANDROID developer, Geyosis Pvt. Limited, Hyderabad and is going to complete his master's degree from the department of Computer Applications. His main research interests include Android, Java, and Mobile Networking.

Mr. K.Anji Reddy received the M.C.A degree from Osmania University, in September 1988. He is currently working as Head of the Department and Sr. Asst. Professor, Department of computer applications, VRSEC (Autonomous), Vijayawada, Andhra Pradesh. He has 12 years of teaching experience and pursuing Ph.D (computer Science) in Rayalaseema University, Kurnool. His research areas are Database management systems, Data mining and Data warehousing.