Abstract—This paper discusses Android-Bots which is a software developed for the Android™ mobile devices. Since Android™ operating system lack of proper sound profile, alarm and battery management features this 3rd party software adds these features for the device. This application mainly changes device settings according to the current location of the mobile device. Although there are some similar software developed, their location finding is inefficient due to GPS usage.

Index Terms—android, GPS, sound, profile, battery, alarm, mobile.

I. INTRODUCTION

This document consists methodologies, techniques and technologies used for developing Android-Bots. It describes the architecture of Android™ and the architecture of the Android-Bots as well as implementation of the mobile software.

Android-Bots is developed to work with any Android™ mobile device which has version 1.5 or higher. This software can be divided into five major parts depending on the functionality. In the user interface these five sections can be clearly identified. They are named as Profile-Bot, Alarm-Bot, Battery Bot, Locator and Bot Service.

The concept of Bot describes an automatic robot that is designed to perform a specific task. Once they are setup and start they will perform until user stops them. This concept is used in the overall software so that it will reduce the complexity of the software architecture and the user interface.

Profile-Bot mainly targeted on setting up sound settings of the device depending on the location. Android™ doesn’t have feature called profiles as other mobile operating systems does. Therefore this software gives that feature to the Android device.

Alarm-Bot is simply an alarm manager which activates after specified location is reached. It finds the current location by using cell tower id of the mobile device.

Battery-Bot is the battery manager. User can set various system settings such as screen timeout duration, brightness, Bluetooth on/off, WiFi on/off etc. It has three pre-defined battery levels.

Chapter 2 of this document describes the main sections of the software and their features. It includes all five sections, user interfaces and how these sections reduces it complexity.

Chapter 3 of this document describes the architecture of the Android™ operating system and its’ layers such as kernel, runtime, libraries and application framework.

Chapter 4 includes design and implementation of the software and the core techniques which added for communicating between different Bots.

Chapter 5 is the conclusion and the future developments of the application. It summarizes the content of this document.

II. MAIN FEATURES

A. Profile-Bot

Profile-Bot give Android™ user a new experience by giving sound profile management which almost all the mobile operating systems have but not in Android™. Profile-Bot has different sound profiles which already defined for the user. User only have to do is setting up the locations to the each profile. For an example user can set “silent” profile within the office. The Profile-Bot automatically switches to “silent” profile if the user reaches to office. Multiple locations can be set to a profile.

For simplifying user interface and user settings all the sound settings of each profile is pre-defined. This will improve user-friendliness of the application. There is software which has many options for the user. But it will annoy the user because of the complex settings.

Basically these profiles have sound settings such as ringer type and ringer volume. These are automatically changed by the software so that user should not worry about changing ringing volume.

B. Alarm-Bot

Most of the time alarms will activate with the time. Since Android-Bots is capable of identifying location Alarm-Bot can activate its’ alarm when specified location is reached. This is mainly targeted users who use public transportation. Because user can setup an alarm to activate when the destination is reached. Then user doesn’t want to bother on the current place. Alarm-Bot will alarm the user when destination comes.

Alarm-Bot notifies the user by adding a notification to the toolbar of the Android and also plays music as an alarm tone. User can switch off by clicking the notify toolbar.
C. Battery-Bot

Battery-Bot is a battery management tool. It’s mainly target on controlling hardware features depending on the battery level. It has three pre-defined battery levels so that user will not get confuse with complex settings. Hardware settings such as screen brightness, screen timeout, WiFi, Bluetooth, Data synchronization on/off etc. can be changed using Battery-Bot.

Battery management software is not new to the Android world. But most of them have problems due to Android version compatibility. Android versions keep changing and accessing to the hardware is restricted/allowed in new versions. Therefore as a solution I use class reflection methods. It will select suitable class for different Android version.

D. Locator

Most essential section of the software is the Locator. It gives the user to add, edit and delete locations from the location database. User can set a name for the location and can be scan for getting cell-id from the network. It will save the locations after user stops the scan.

Cell-ID is given to the area by the network provider. Depending on the signal strength mobile device will select a suitable network sub stations. Therefore it will have more than one cell-id for a single place. This reason adds scanning option to the Locator.

Scanning cell-id is implemented by using separate java Thread. It will help to improve responsiveness of the software while scanning is done. It will search for new cell-ids for the location. Once a new cell-id is found it will be added to the software database.

E. Bot Service

Bot Service is the real time activating service which runs on the background of the Android™ devices. User can start or stop this service. If the service is stopped it can be considered as the overall software is disabled. Bot Service captures the cell-id and it will act depending on the settings given by Bots.

Using Android services is very resource consuming method. It will consume battery and discharges quickly if the services are not handled properly because services are always running until user stops them or turning off the device.

For achieving proper handling it is used separate threads, low memory and CPU consuming techniques when implementing Android-Bots.

If we use same software instead of implementing separate service the overall software will consume more memory to save state of the user interface. When using separate service user interface applications can be cleared from the memory. Therefore it will improve the performance of the software application by saving more RAM for the other software.

III. Architecture

A. Android Architecture

Android™ is one of the most popular mobile operating system built by Google. It gives open software platform for the developers and since Android™ is an open source operating system developers are encouraged to develop software or to develop core of the operating system.

Android™ is an Open Handset Alliance (OHA) project done by Google. It uses Linux kernel as the kernel of the operating system. Therefore Android architecture and the performance are more similar to other Linux based operating systems. It is registered under Apache 2 license as an open source project.

Mainly Android™ architecture can be categorized as OS, middleware and applications. Figure 3.1 shows a complete architecture of Android operating system.

OS category includes the Linux Kernal which has drivers of the Android device. Most of the drivers like camera, display, flash memory, keypad, binder (IPC), WiFi, audio and power management are included in the Kernal. Linux Kernal connects the hardware of the device with the upper layers. This is names as Hardware Abstraction Layer (HAL). Other than the drivers Kernal is responsible of few more important tasks such as memory management, process management and networking. Since Android devices have less amount of memory it has to be managed efficiently. Otherwise it will overloaded by the applications.

Middleware category includes Libraries and Android Runtime. Libraries are the techniques which add features to the operating system. Most of these libraries are developed using C/C++ programming languages. But it has interface for Java. Therefore developers are capable of using Java when they developing applications or operating system. Some of the common libraries are Surface Manager, SQLite, Media Framework, OpenGL, SSL, Free Type, SGL, libc and Webkit.
Surface Manager handles the user interface windows. There are many View controllers in Android to give the user a user-friendly environment to work. All of these controllers are managed by the surface manager. Android is capable of showing 2D graphics as well as 3D graphics. OpenGL is used for these kinds of graphics. For database handling SQLite is used. This is more similar to SQL but it reduces some of the functionalities. Webkit is used for browser engine. It is considered that Android browser is one of the fastest mobile browser engines in the mobile world. For playing media files Media codecs are included in the libraries.

Android Runtime can be included in middleware of the operating system. It has virtual machine which is similar to Java Virtual Machine (JVM) names as Dalvik Virtual Machine. This is where the developed applications are running. Because of the limited memory and battery power Android cannot directly use JVM. Therefore more reliable and mobile device targeted Dalvik Virtual Machine is used. It has Java 5SE libraries which will reduce restrictions of the developers when programming.

Application layer includes built-in Android applications and the application frameworks. Every mobile/portable device must have applications which perform basic features. If the device is a mobile then it must have Phone, Contacts, Home screen, browser applications. If it is a media device it should have all these basic applications other than the phone application. These applications should also give an Application Programming Interface (API) for the 3rd party applications. Therefore these Android applications are included in Application layer. Application framework gives the user interface controllers for developing software. They manage the behavior of the running application. For this purpose Activity, Window, View, Content, Notification, Location, Resource, Telephony and Package managers are implemented in AndroidTM operating system. Each manager is assigned to perform some sort of management. For an example Activity Manager manages application life cycle running in the device. Since these managers are implemented developers do not have to design these managements manually.

B. Android-Bots Architecture

Android-Bots has two tier architecture. Upper tier runs in the Dalvik VM. Upper tier consists user interface and the background android service. Since these two components are logically separate it connects to common file storage and settings, cell-id locations are passed through by using this file storage. This file storage is the lower tier. A manually created database is used to save data and application dedicated storage is used to store the necessary files. Figure 3.2 shows the basic architecture of the Android-Bots.

IV. DESIGN AND IMPLEMENTATION

A. User Interface design

Since Android-Bots is mainly targeted on user-friendly environment, settings of the various Bots can be setup in a wizard manner. Figure 4.1 shows the basic steps of the various Bots.

B. Internal Design and Implementation

Each Bot (Profile Bot, Alarm Bot and Battery Bot) has a separate class to implement them in an objective manner in Java. All the functionality of each Bot are included in the class methods. This technique is also used for Locator as well as in “Bot Service”. As described in architecture user interface classes and “Bot Service” class is logically separate. Although they are included in the same app, in runtime they execute as two applications.

File storages are used to connect the classes. Although it is inefficient, this is the one of the best way to reduce memory consumption. Other technique is binding the service with the user interfaces which costs a large amount of memory.

Application dedicated file storage is used. This enhanced the security of the data stored. Data is serialized and stored in files. Using SQLite for small data storing is not an efficient method. Since this software settings are small in size application dedicated storage is a good solution.
V. CONCLUSION

A. Future Developments

The concept of using Bots can be improved by adding artificial intelligence. User might be fed up with setting locations and other system settings. Therefore it will be developed so that Bots will identify the user decisions according to the time and location. Then Bots can perform automatically without manually setting up.

B. Conclusion

Android-Bots is mainly a profile, battery and alarm management software. There are similar utility software in Android Market. But Android-Bots has user-friendly interface and has three different functionalities in single software. This will be helped for the Android users in their day-to-day life. Since Android-Bots use almost best techniques for protecting its’ reliability and responsiveness this application will perform well in any Android™ mobile device which has version 1.5 or higher.

ACKNOWLEDGMENT

I would like to thank Dr. Shahani Markus Weerawarana, Dr. Shehan Perera and Dr. Praveen for their guidance and support to complete this project.

REFERENCES