D. C. S. Weerasinghe

**Android Bots
Software Architecture Document**

Version 1.0

Revision History

Date	Version	Description	Author
29/Oct/2010	1.0	Software Architecture Document	Chamika Weerasinghe

Table of Contents

Software Architecture Document

1.   Introduction

1.1   Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

1.2   Scope

This document covers complete design description of the system Android Bots. System Engineers, project managers, developers and end-users must use this document as the reference.

1.3   Definitions, Acronyms, and Abbreviations

Android Bots   – the product, hereafter referred-to as just 'product' and 'program'.
Android   – the Android™ mobile operating system developed by Google Inc.
CellID   – the ID code associated with the particular 'cell' tower of the cellular service provider.
Wi-Fi   – Wireless Fidelity - wireless network connectivity service
Bluetooth   – simple microwave-based mobile connectivity service for data transmission
SDK   – Software Development Kit
ADT   – Android Development Toolkit
IDE   – Integrated Development Environment
OS   – Operating System refers to the Android operating system in this document.
UI   – User Interface

1.4   Overview

This document contains different architectural views including Architectural representation, Use-Case view, Logical View, Process view, deploying and implementation views to depict different aspects of the system.

2.   Architectural Representation

All the diagrams and descriptions are used standard methods of representing design architectures. Use-Case view is designed using UML 2.0 standards.

3.   Architectural Goals and Constraints

Android Bots provides extended software for the user to manage some functionalities of an Android mobile device. Profile, battery and alarm are some of the functionalities are used in this system. Therefore this system always ensures the safety and security of the user. Since this system does not record any user activities and privacy data, it does not affect the privacy issues of the user. This system records CellID of the current location of the mobile device. Therefore system designed to encrypt these details to avoid unauthorized access.

The system current version 1.0 is currently developed for free and open source community which allows any developer to distribute or reuse the system for better improvements. It is illegal to use this system for corruptions and other software based illegal activities.

This system will be developed according to the project schedule. Mainly there are two iterations in the overall project and will be finalized them before the scheduled date. System will be used Eclipse™ IDE for the overall development process

4. Use-Case View

4.1 Use-Case Realizations
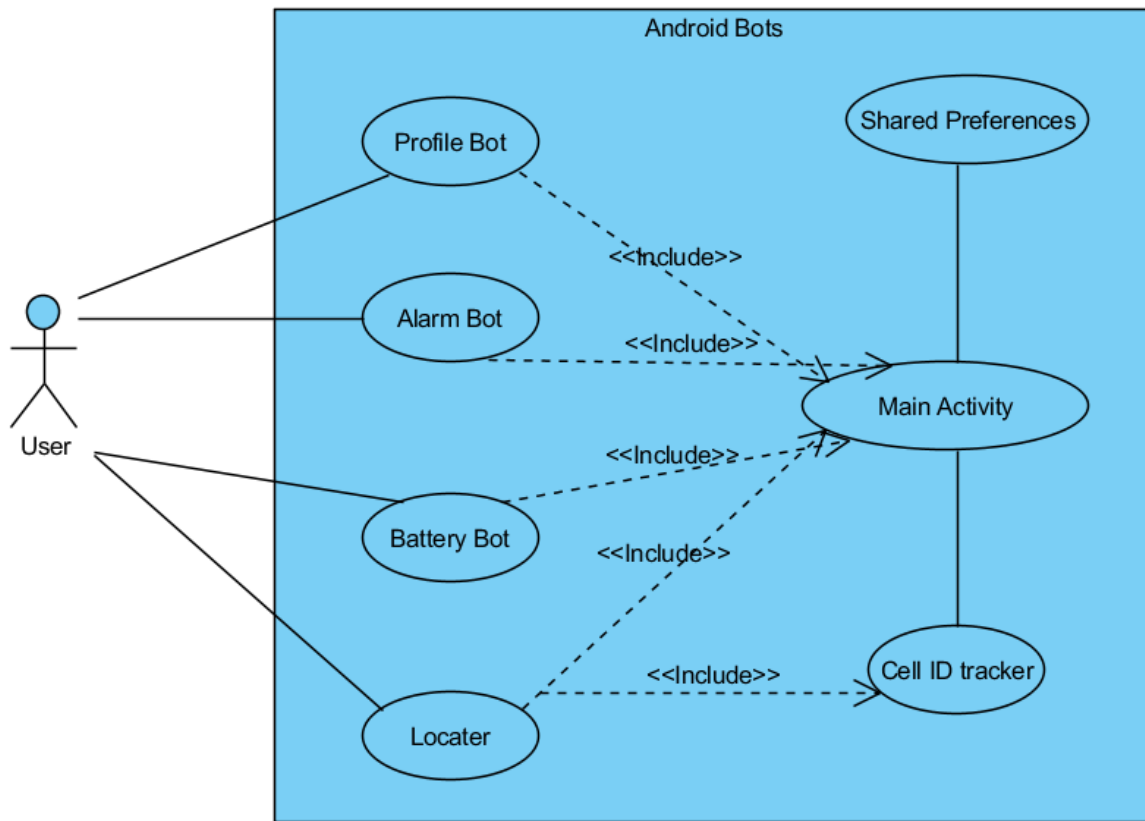
This software has main 4 user interfaces which provide user to set current location and setup new setting of profiles, battery and alarm of the device. Locater UI uses for setting up current location settings. Other 3 UIs are used for setting up specific feature. All the UI has Android activity which extends program "Main Activity". Android Shared Preferences is used for storing user preferences throughout the system.

5. Logical View

5.1 Overview

In Android application life cycle each graphical user interface is an Android activity. Therefore each interface must have separate class. It must be implemented other classes for the functionality of the system. A main class should be designed for communicating between user interface classes. All these facts must be considered when designing software for Android OS.

**5.2   Architecturally Significant Design Packages**
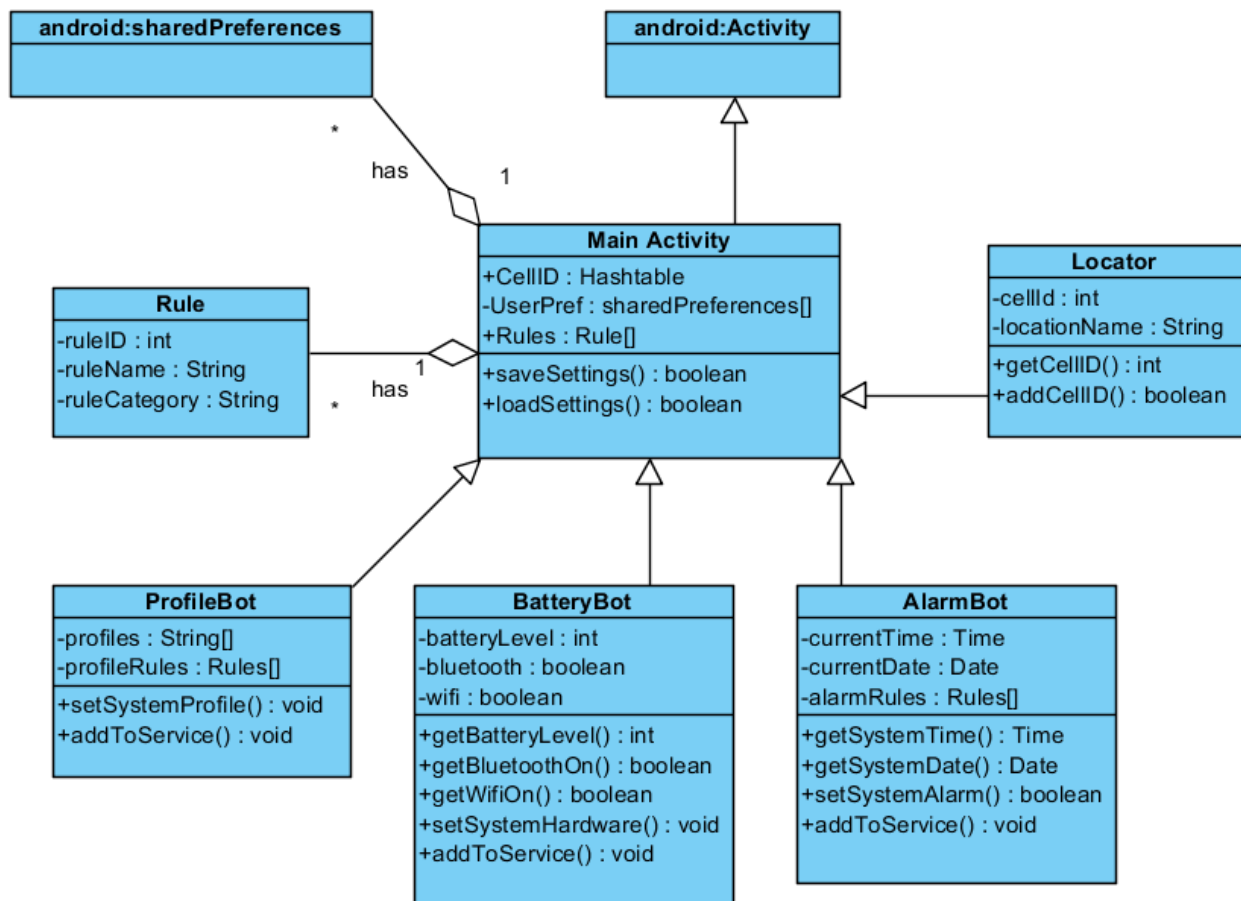
System designed as shown in above UML class diagram. Main Activity class handles all other classes and each class has its own attributes to do their activity. Rule class is used for making rules for specific task. For an example if user wants to change profile in specific location user will create a rule for that using Profile Bot. "android" classes are used for implementing android application life cycle functions.

6.   Process View

Following activity diagrams show the system processes. There are 4 processes for overall system. Locator is used for catching locations and adding them to the application. Profile Bot, Battery Bot and Alarm Bot gives interface for managing profiles, battery and alarm respectively.
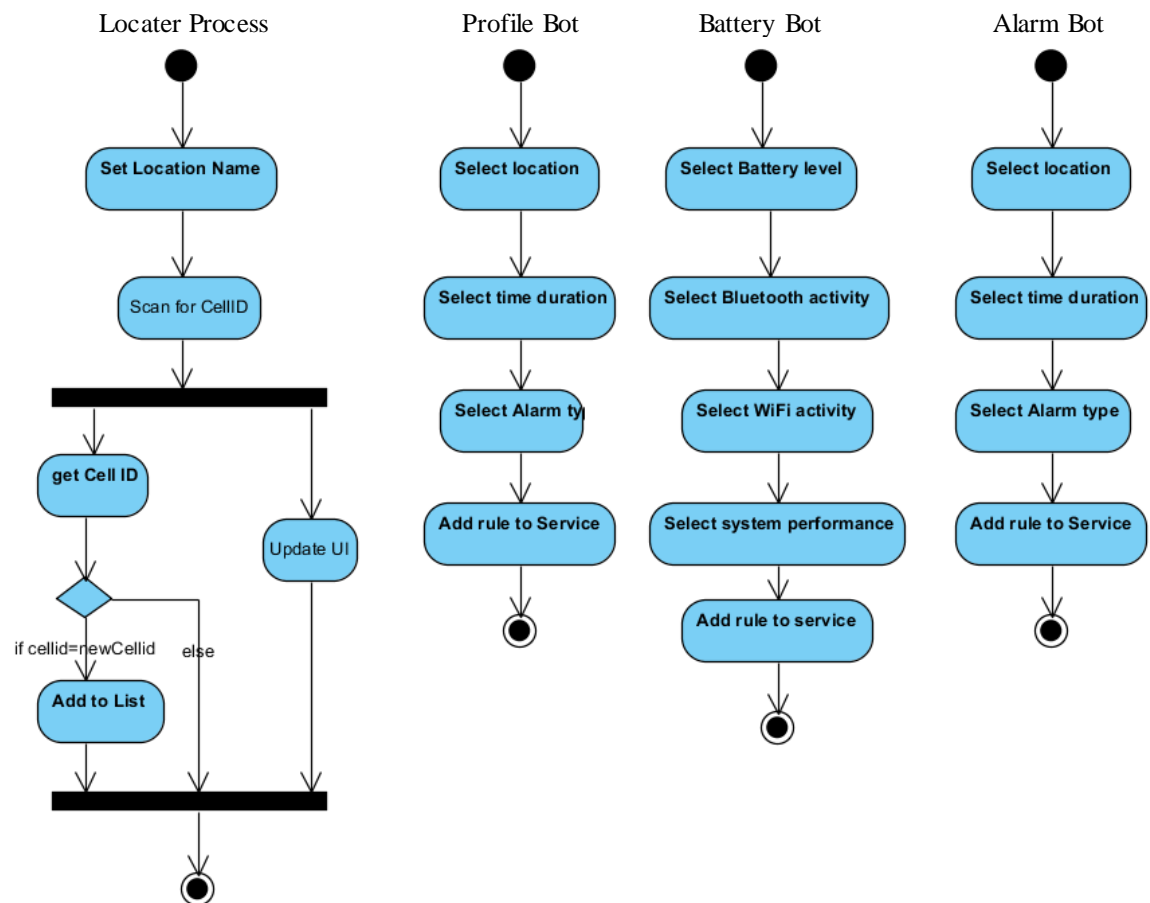
Locater Process
Locater process gives user to get the current cell tower id and save it as a preferred location. It uses separate processes for getting cell id and updating UI to ensure system reliability. Otherwise system may not respond to the user and user has to force close the application.

Profile Bot
This process displays a UI to add new profile rule to the system. Then it adds that rule to the service for performing at the right location and time.

Battery Bot
This process also gives a UI for user adding battery management rule. User can select Bluetooth, Wi-Fi and other system setting when specified battery level reached.

Alarm Bot
This process displays a UI to add new alarm rule to the system. Then it adds that rule to the service for performing at the right location and time.
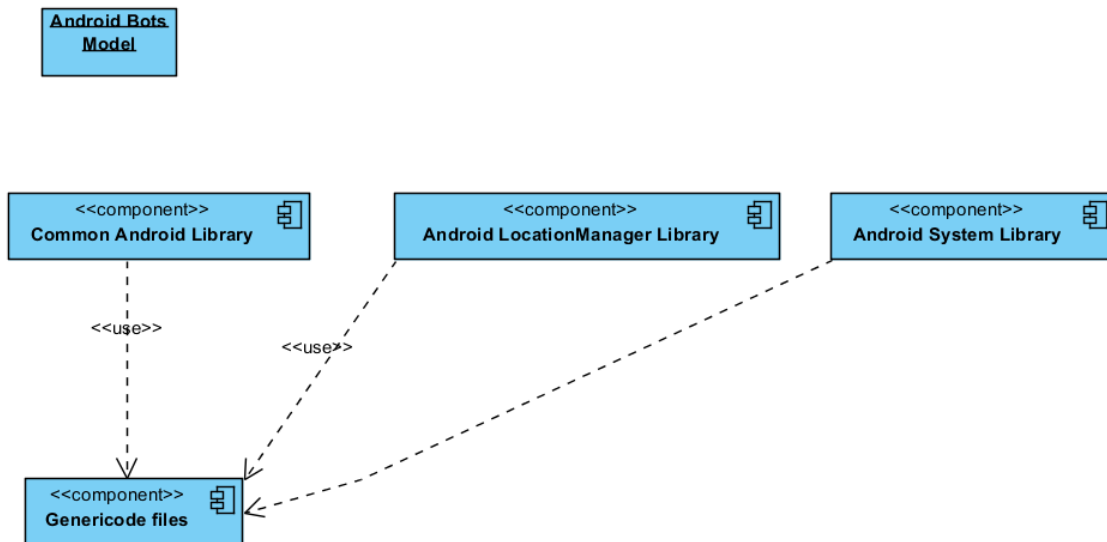
7. Implementation View

7.1 Overview

In designing process it must use common libraries of android to build application framework. Apart from that some other libraries must use to get cell ID and system states of the device. All these libraries and

coding parts are described below.

7.2 Layers

7.2.1 Common Android Library

When implementing system common android library must use in order to build basic android application. Every android application must have some compulsory methods which provide by these common libraries.

7.2.2 Android Location Manager Library

This library must use to identify the cell tower id of the current location from the device. Cell ID is majorly use in this system. Therefore it is a "must have" library for this system.

7.2.3 Android System Library

For the battery management process it is essential to get system state such as Bluetooth on/off, Wi-Fi on/off and other system performances. To get all these details Android Development Kit has many system libraries. These libraries should be imported.

7.2.4 Generic codes

All the application functionalities are to be designed in the generic codes. This layer includes XML files such as layout files, string files, color files, dimension files etc. Other than that resources this layer includes all the codes which are included in the java files.

8. Size and Performance

Android devices have small and slow hardware specifications comparing to PC. But Android OS manages this limited hardware for the application. Architecture design always try to reduce system consumption so that it can achieve maximum perform with limited hardware in the device. Separate processes and android services are used in design to reduce processor consumption.

9. Quality

Application is designed for releasing international Android Market. Therefore all the international standards will be used. Apart from that this software specially consider about the reliability. User will face unexpected events and sounds if the software does not operate accurately. Therefore overall quality can be good comparing to other open source software.