Software Design Specification

For

Secure Android

Version 1.0 approved

Prepared by:

NC Afifa Maqbool

PC Maryam Khan

NC Shahid Nazir

Supervisor
Lt Col Muhammad Tayyab Ali, Ph.D.

MCS, NUST
Monday, December 21, 2015

TABLE OF CONTENTS

FIGURES

Revision History

Name	Date	Reason For Changes	Version

1. INTRODUCTION

The aim of our project is to develop a secure version of Android which will address the increasing use of mobile devices particularly in government bodies, with the need for improved security. This design document captures all our functional requirements and shows how the modules and components interact with each other conceptually. The design also shows as to how we plan to implement all these requirements. Augmented with various diagrams, the design document shows, the design ponders upon different facets of user interactions with the system followed by their responses. The document also caters for the brief tradeoffs of the few aspects of the design, intended to be included.

1.1. Purpose of this document

The purpose of this document is to present a detailed description of the design of our Secure Android. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, its entire process, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system. It will explain how the system will particularly be designed to help the users of Secure Android to know when the security of their device is being breached and when the device wants you allow or reject certain actions and permissions.

1.2. Scope of the Development Project

The aim is to develop a secure version of Android which will address the increasing use of mobile devices particularly in government bodies, with the need for improved security. Android was chosen, for its widespread use and the openness or adaptability of the platform. The goal is to prevent external mobile apps from granting themselves extra privileges, prevent apps from sharing too much data and to prevent the bypass of security features.

1.3. Definitions, Acronyms, and Abbreviations

Attached as Appendix ‘A’

1.4. References

1.4.1. UML Diagrams

[1] The Unified Modeling Language Reference Manual. James Rumbaugh, Ivar Jacobson, AND Grady Booch. 1998. P.81. **ISBN:** 0-201-30998-X.

[2] The Unified Modeling Language Reference Manual. James Rumbaugh, Ivar Jacobson, AND Grady Booch. 1998. P.87. **ISBN**: 0-201-30998-X.

[3] The Elements of UML™ 2.0 Style. Scott W. Ambler. May 2005. **ISBN:** 9780521616782

1.4.2. Android Architecture

[4] https://source.android.com/devices/

[5] http://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture

[6] https://en.wikipedia.org/wiki/Application_software

1.4.3. Android Security

[7] http://dl.acm.org/citation.cfm?id=2046779

[8] http://developer.android.com/reference/java/security/Certificate.html

[9] http://www.androidhive.info/

1.5. Overview of Document

This document shows the design and working of Secure Android. It starts from higher level details for a non-technical reader to understand just by seeing the diagrams to the lower level details that aid the developer to code and understand other technical details. The document is divided into sections and is already listed in the table of contents and figures list. However, here is a brief description of all the sections.

1.5.1. System Architecture Description

In this section, the overall architecture of the system is discussed, including the introduction of various components and subsystems. It is mainly supported by system Architecture diagram which shows an insider's perspective of the system by describing the high level software components that perform the major functions to make the system operational.

1.5.2. Structure and relationships

This section ponders upon the interrelationships and dependencies among various components. It is mainly described by a diagram which is further augmented by explanatory text.

1.5.2.1. UML Class diagram

UML Class diagram further manifests the description of low level components of the software that include data storage and state details, thus making the system adequately comprehensible.

1.5.3. User Interface Issues

This section presents the main principles of the product's user interface. Not touching about the technical details, the section is described by an overall diagram which is also augmented by explanatory text. Moreover, UML Activity diagrams, UML Sequence diagrams, and UI Design diagrams also elaborate the User Interface issues in a more intelligible manner.

1.5.3.1. UML Activity diagrams

UML Activity Diagrams follow a workflow-based approach to describe the overall functioning of the system. They are a very good means to see how various steps are involved in major tasks inside a system using a flow chart pattern without getting into the technical details. [1]

1.5.3.2. UML Sequence diagrams

UML Sequence diagrams show how different objects are involved in the completion of a functionality of the system. They have a unique format that allows the reader to see how many objects are used vis-à-vis their duration; for the completion of a system requirement. [2]

1.5.4. UI Design

Some snapshots of graphical user interfaces are shown in this section that prototype the way a user shall be interacting with the system.

1.5.5. Detailed description of components

This section contains detailed description of all the major components of the system in a structured pattern (table), comprising of 10 x rows. The pattern (table) maintains symmetry in the document structure; and therefore it is followed for each of the components. Each part/row of the table is identified by a *label*, explaining the purpose of each point. The description of each point vis-à-vis the component being discussed, ponders upon the detailed account of it in the system.

1.5.6. Reusability and relationships to other products

This section focuses upon the Reusability aspects of the various components of the system. Since the project in hand is all new and doesn’t carry out any enhancement work in the already existing system, so Reusability is just a recommended strategy to be employed while organizing various system components.

1.5.7. Design decisions and tradeoffs

This section highlights various design decisions and the ideas behind those. It enables the reader to understand the important crux of the design that is being used while excavating a bit more about the motivations behind those decisions.

1.6. Work Breakdown Structure

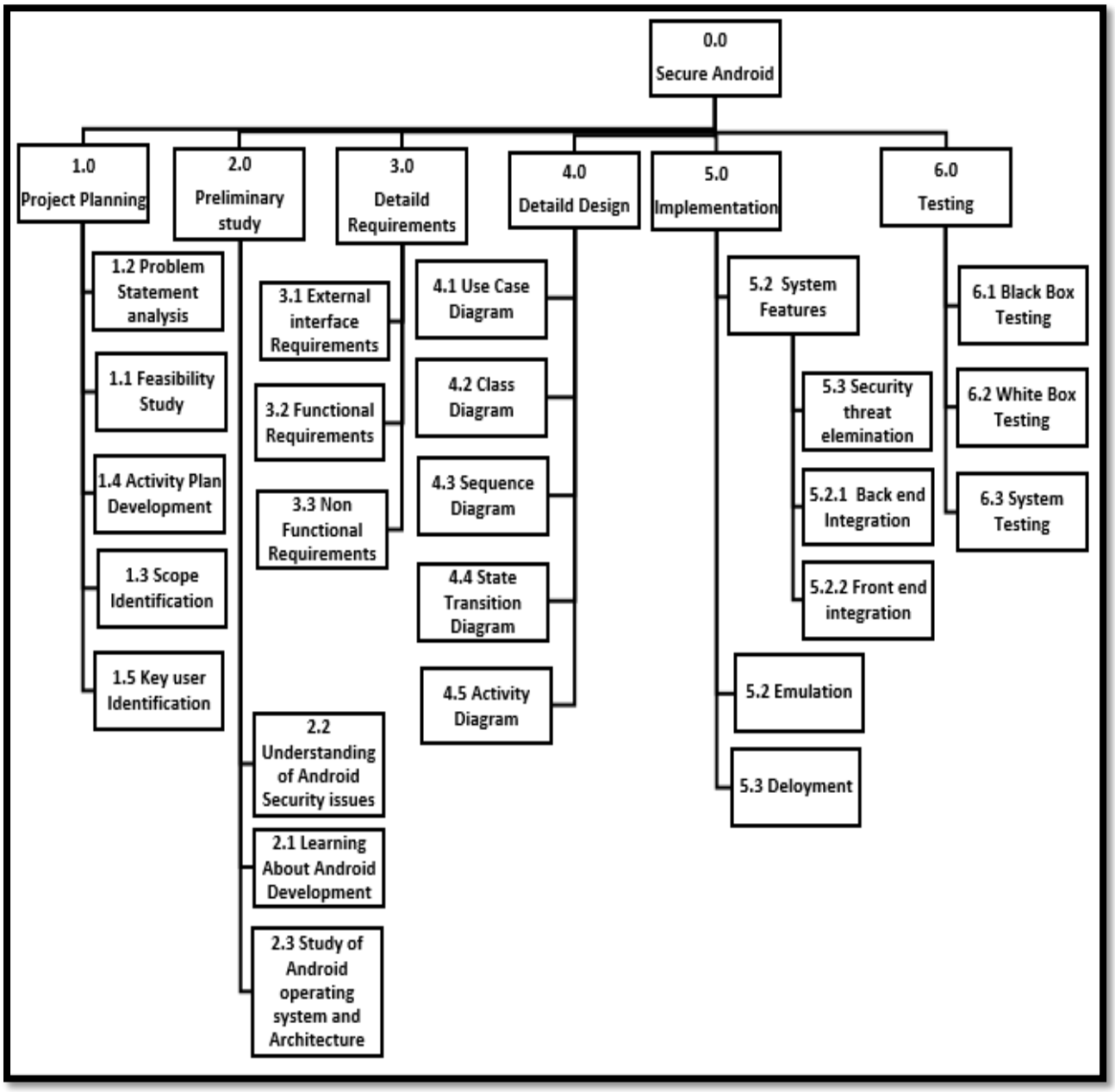

Figure 1 – Work breakdown Structure

2. SYSTEM ARCHITECTURE DESCRIPTION

2.1. Overview of the modules

The system will be architected mainly in 5 fundamental modules “managing app permissions”, ”runtime security check and warnings”, ”Scheduled Phone scan” and “certificate and keychain manager” having other sub modules too as shown in the following abstract diagram:

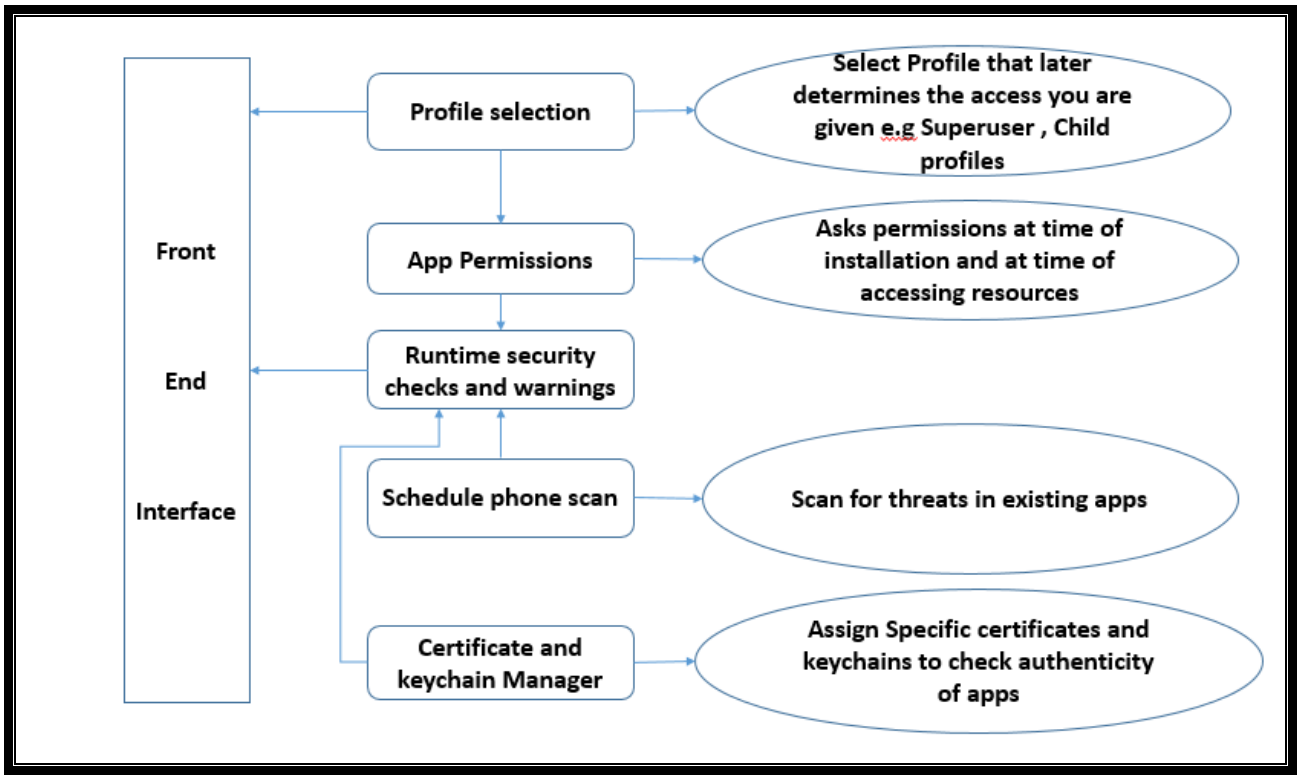

Figure 2 - Overview of the Modules

2.1.1. Description of the modules

The “Front End Interface” will be used to display all the warnings and to ask about all the permissions that are required. “App permissions” will ask the user about the permissions she/he wants to grant an app at the time of installation and at the time of accessing a resource that is previously not allowed to the user. “Schedule phone scan” will allow the user to schedule full phone scans which will allow Secure Android user to scan for threats and security lapse inside existing apps on the device. ”Certificate and keychain manager” will allow the user to issue certificates and keychains to the apps that she/he trusts and that app will be given the access to the phone according to what is allowed by that certificate.

2.1.2. System Architecture

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.

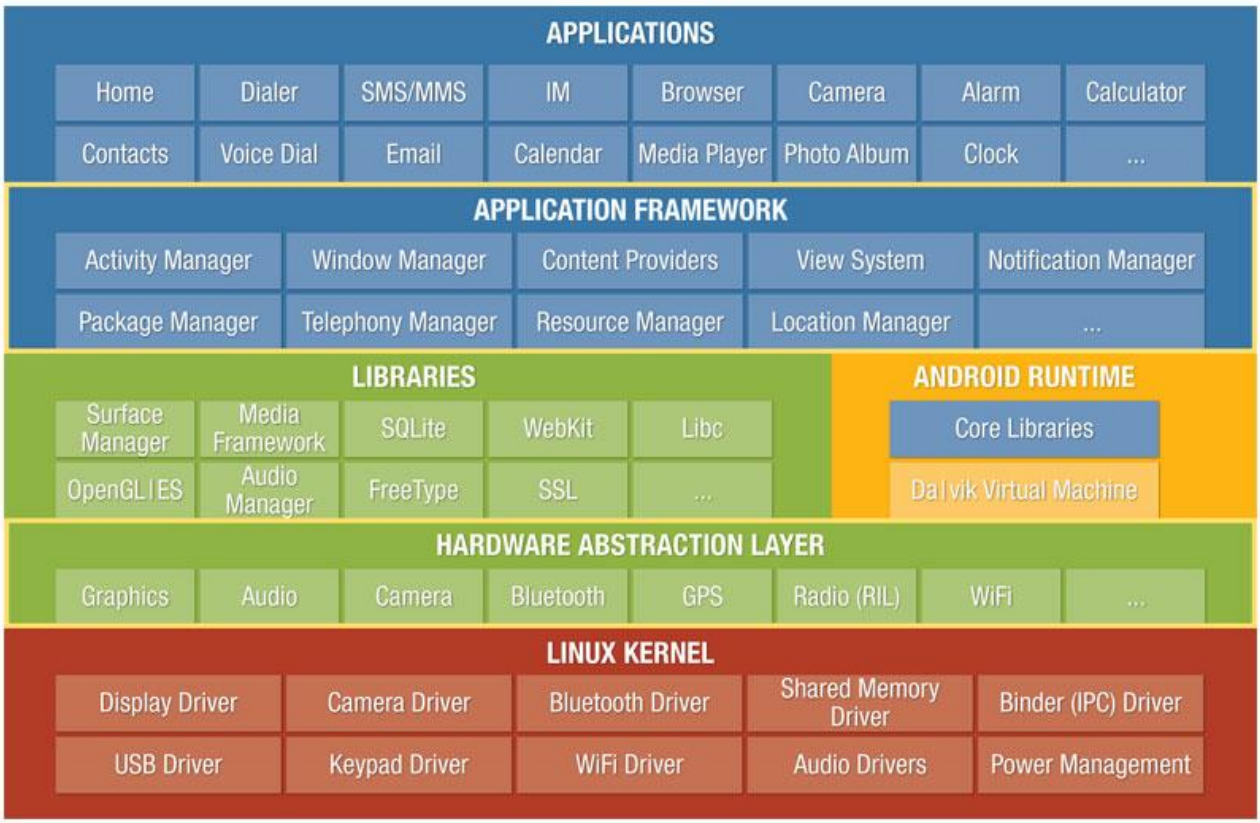

Figure 3 - System Architecture

2.1.3. Layers Details

The details of the layers have been discussed below.

2.1.3.1. Linux kernel

At the bottom of the layers is Linux. This provides a level of abstraction between the device hardware and it contains all the essential hardware drivers like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

2.1.3.2. Libraries

On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.

2.1.3.3. Android Libraries

This category encompasses those Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access.

2.1.3.4. Android Runtime

This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called Dalvik Virtual Machine which is a kind of Java Virtual Machine specially designed and optimized for Android.

The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine.

The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

2.1.3.5. Application Framework

The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

The Android framework includes the following key services −

1. Activity Manager − Controls all aspects of the application lifecycle and activity stack.
2. Content Providers − Allows applications to publish and share data with other applications.
3. Resource Manager − Provides access to non-code embedded resources such as strings, color settings and user interface layouts.
4. Notifications Manager − Allows applications to display alerts and notifications to the user.
5. View System − an extensible set of views used to create application user interfaces.

2.1.3.6. Applications

You will find all the Android application at the top layer. You will write your application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, and Games

2.1.3.7. Hardware Abstraction layer

The hardware abstraction layer (HAL) defines a standard interface for hardware vendors to implement and allows Android to be agnostic about lower-level driver implementations. The HAL allows you to implement functionality without affecting or modifying the higher level system. HAL implementations are packaged into modules (.so) file and loaded by the Android system at the appropriate time.

2.2. Structure and relationships

Focusing upon the internal structure of the system, this section ponders upon the interrelationships and dependencies among various components.

2.2.1. Overall Structure of the system

The diagram shows the main components of the system along with their interactions with each other. It mainly describes the system structure which is further augmented by the explanatory text as follows:

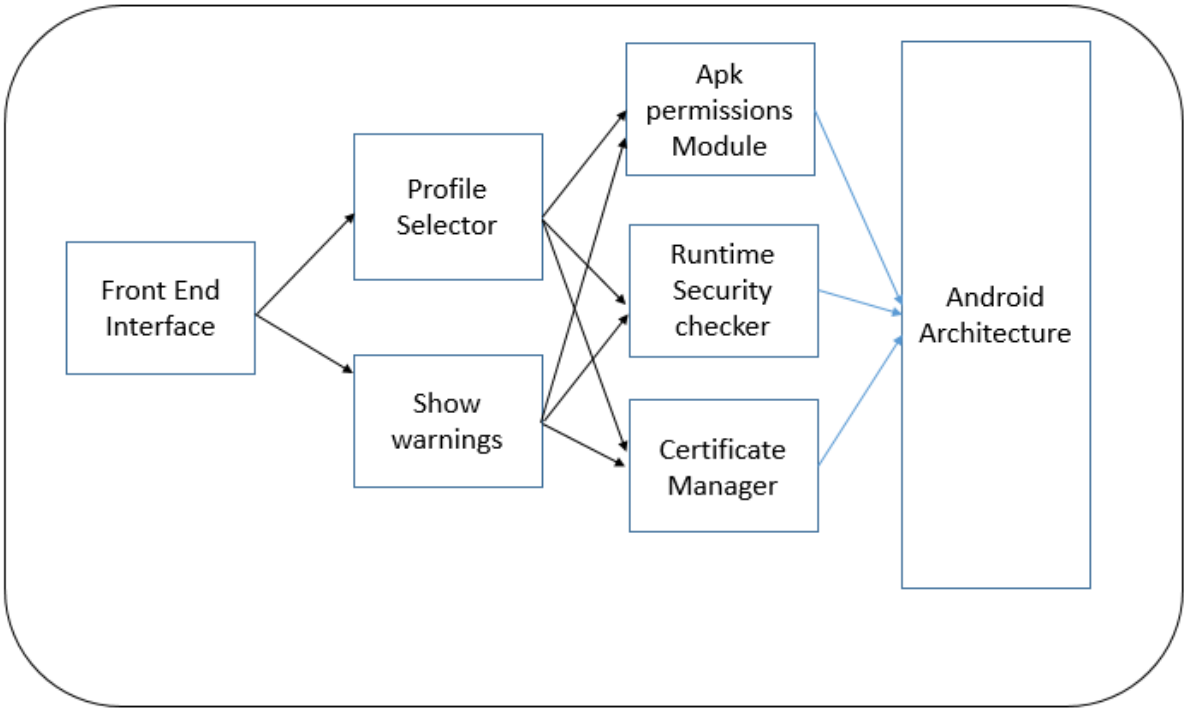

Figure 4 - Overall Structure of the System

2.2.1.1. Front End Interface

Front End Interface caters for the visual needs of the System, wherein the Human-Computer-Interaction aspects are considered to enable the user to communicate with the system profoundly. It is connected with App permissions and shows warnings when a security breach occurs or a new permission is needed for an app. The basic operating system that is Android works normally in terms of the interface as it works on any other device.

2.2.1.2. Profile Selector

This module controls the permission and access that a user is granted. There is the superuser profile that has total access of the device and then there are child profiles that have relatively less control on the device and are all controlled by the superuser profile and the only the superuser has the access to create new profiles and set the access that they are granted.

2.2.1.3. Show warnings

This module is the used to show:

1. Pop-ups will appear as the system asks for the permissions that are required for different applications.
2. Warnings will appear whenever the system suspects a security breach.

2.2.1.4. Apk Permissions

This modules controls the permission of an app at all times when the app is being installed.

2.2.1.5. Run Time Security Check

This module is used to perform runtime security checks on all the apps and general working of the device. This module can be used to schedule security checks and scans.

2.2.1.6. Certificate Manager

Certificate will be generated whenever an app is installed according to the permissions it is given the certificate is then used as an authentication for the app to access different features of the system. The Certificate Manager will handle, delete, edit and save any certificates that are available to the system.

2.2.1.7. Android System Architecture

This module is the basic functionality of Android, since we are working on the already built version of Android so the Android is as it is it all the rest of its functionality except for the added security features.

2.2.2. Component Diagram

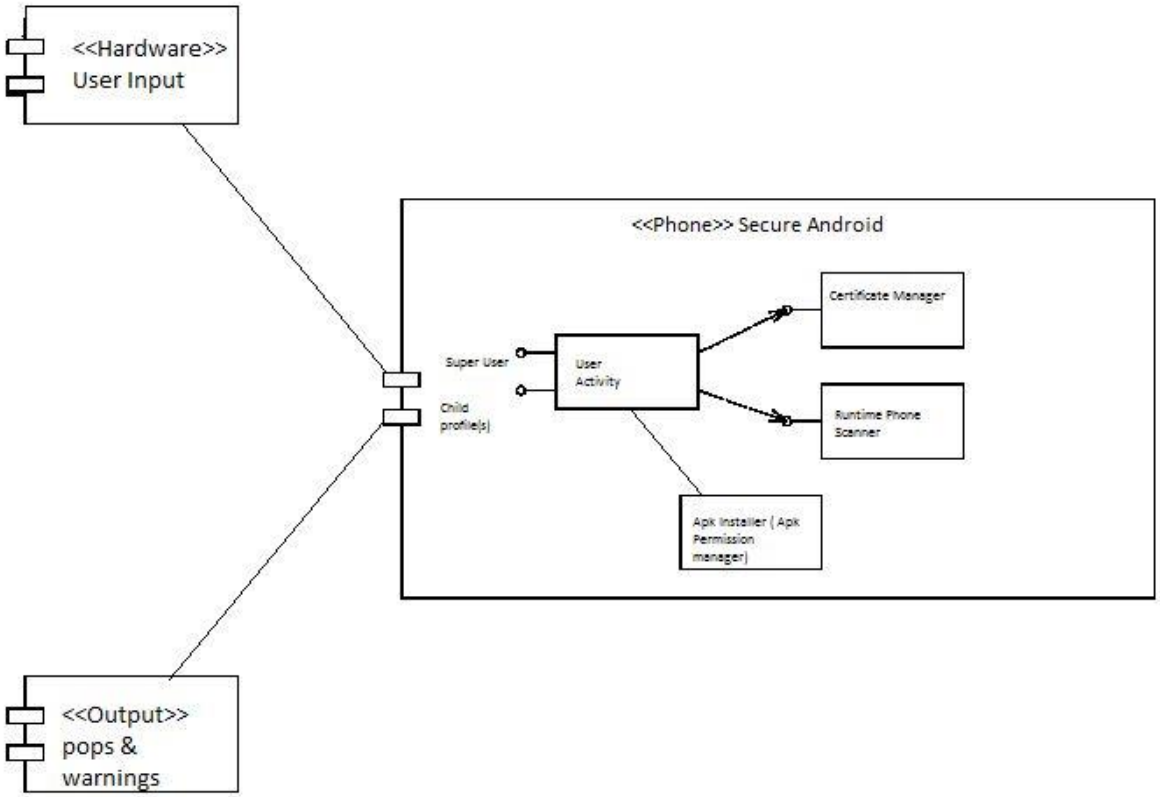

Figure 5 – Component Diagram

2.2.3. Use Case Diagram

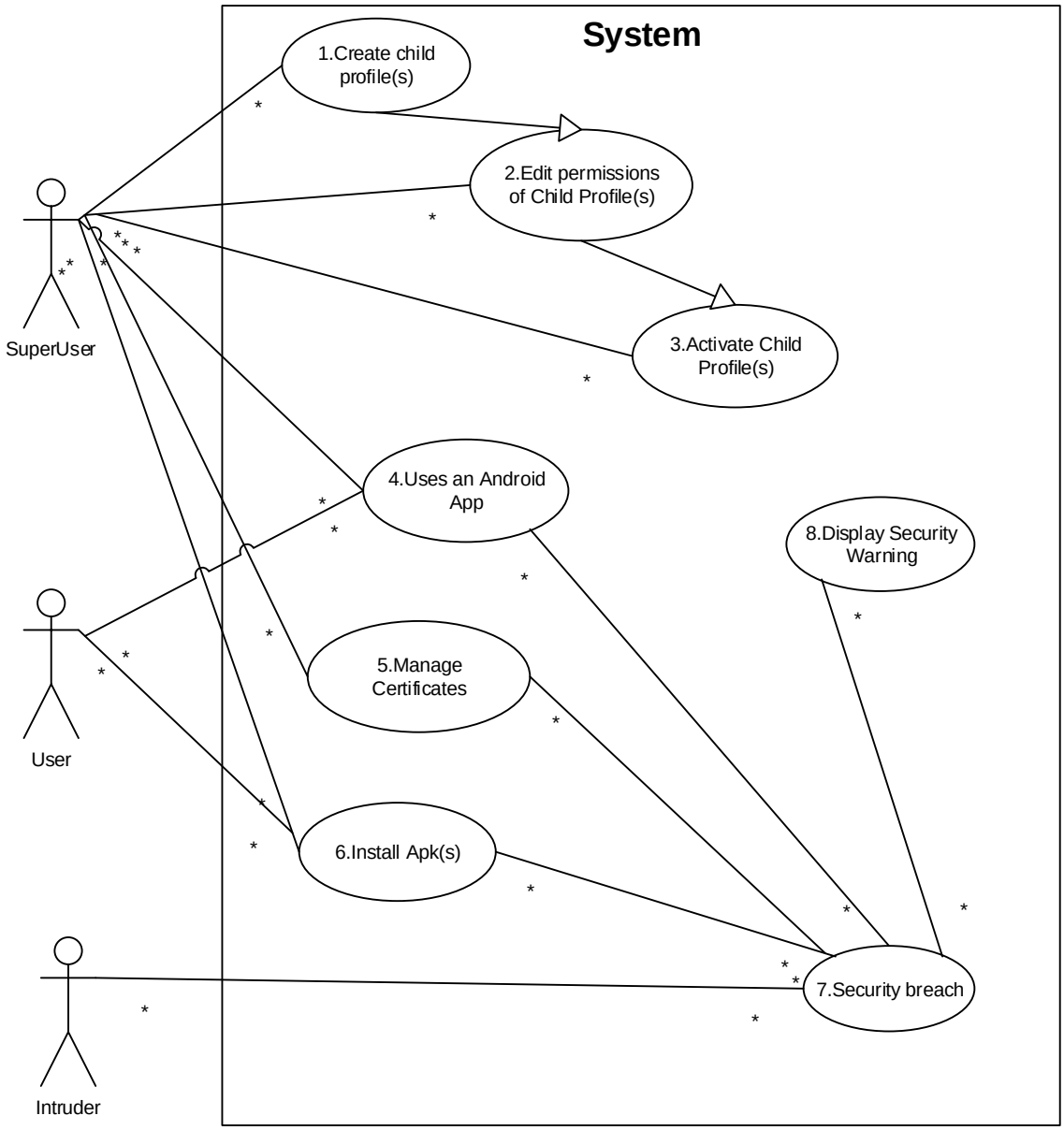

Figure 6 - Use Case Diagram

2.2.3.1. Description of Use Cases

Attached as Appendix 'B'

2.2.4. Class Diagram with description

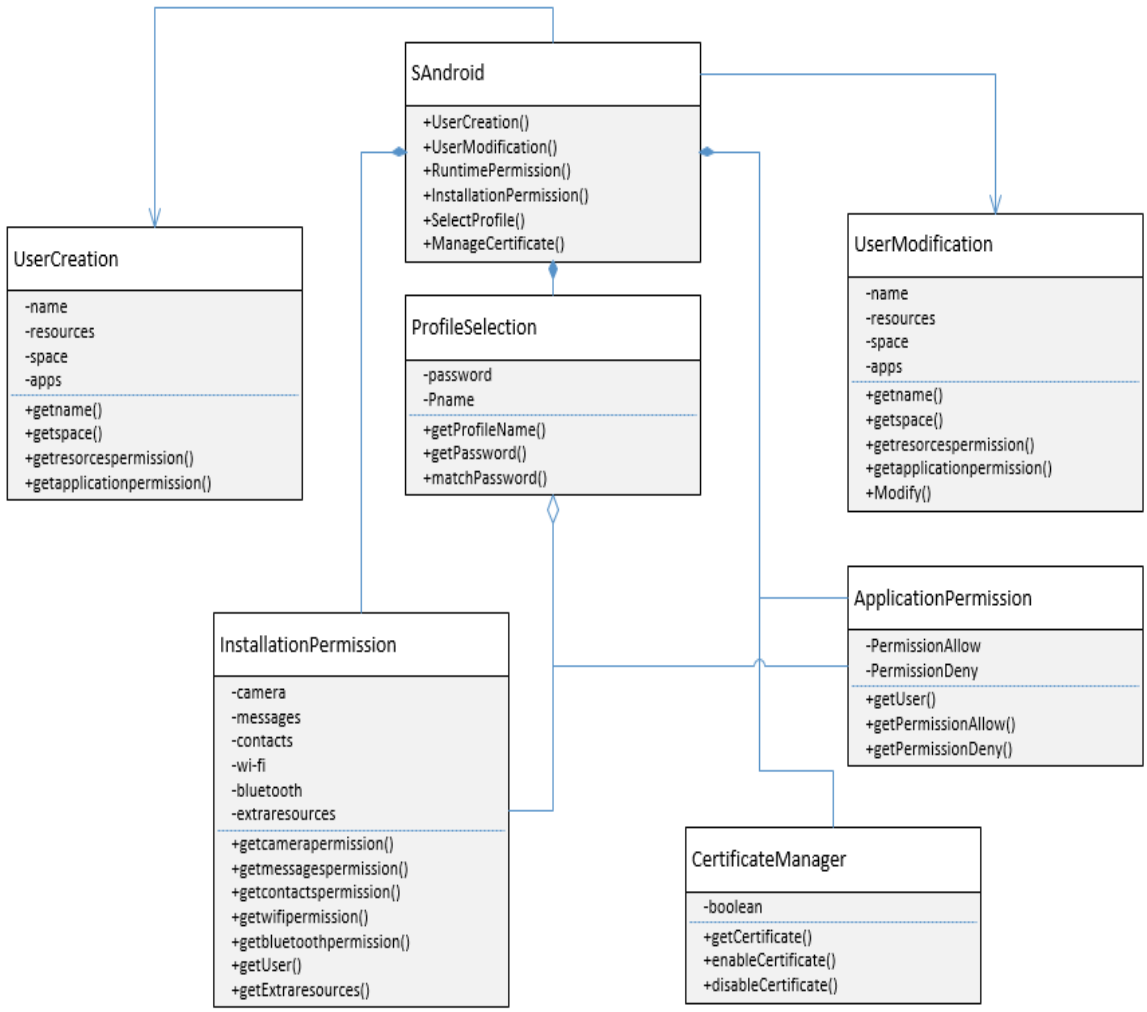

Figure 7 - Class Diagram

2.2.5. Description of the diagram

The diagram reveals class-structure of the application. All the classes shown in the class diagram are further described in the sub sections.

2.2.5.1. SAndroid

SAndroid is short for Secure Android, Contains all the functions which our system has to offer. It calls all the functions from respective classes as and when invoked by the user.

2.2.5.2. User Creation

This class contains the information about user creation and how we will create a new user.

2.2.5.3. Profile selection

This class contains the information about profile selection we will have the passwords usernames and all the relevant information here.

2.2.5.4. User Modification

It contains all the functions required for user management features. Such as allocating space editing name and other information.

2.2.5.5. Application Permission

This class has information on which apps to allow access to certain resources and which apps we should deny access to those resources.

2.2.5.6. Certificate Manager

This class contains data and functions of the certificate manager and its ability to enable disable and delete app certificates.

2.2.5.7. Installation permission

This class has information on allowing access to resources such as camera, messages, contacts, Bluetooth, Wi-Fi and extra resources or denying that access to applications at the time of installation.

2.3. User Interface Issues

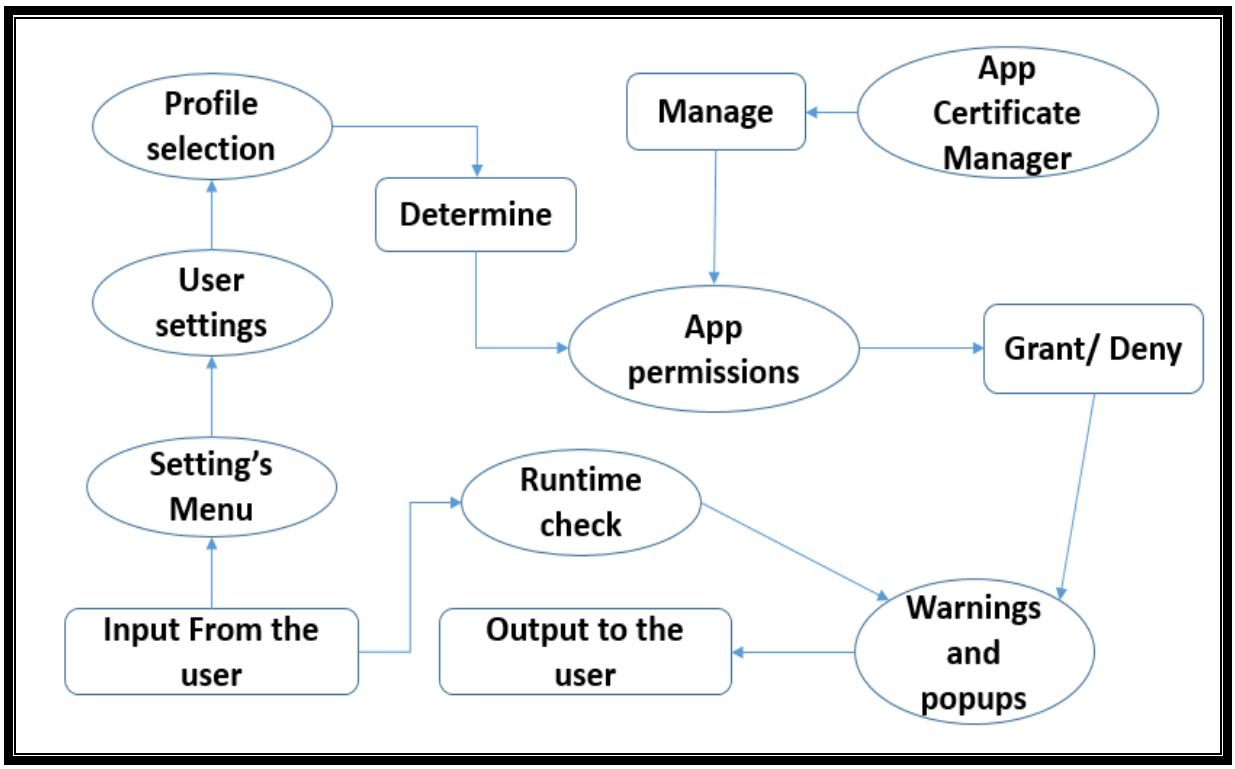

Figure 8 - User Interface Issues

2.3.1. Description of the diagram

The diagram reveals the functionality of the System, keeping in view the user interface issues of the system.

2.3.1.1. Settings Menu

The settings menu is the way to access profile selection in the settings menu we find a user settings option inside which we have the profile selection settings.

2.3.1.2. Certificate Manager

Certificate Manager manages the available certificates and allows full access of those certificates to the superuser. We can enable/disable and delete all the available certificates.

2.3.1.3. App Permissions

App Permissions are managed by the Superuser for all the other child users the app permissions are checked and then warnings and popups appear accordingly.

2.3.1.4. Runtime checks

Runtime checks include security scans and scheduled auto scans that are done by the system to check and find all the security breaches in the system.

2.3.1.5. User Settings

User settings include all the settings to manage the users of the device the user setting also includes the profile selection settings.

2.3.1.6. Profile Selection

Profile selection allows the user to set child profile(s) if he/she is the superuser. He/she then sets the allowed access of those profiles i.e. determines the permissions that are granted to the user of that profile.

2.3.1.7. Determine

The current profile of the system determines the app permissions that it is given. The superuser has access to all the permissions of the system and the apps.

2.3.1.8. Manage

The App certificate Manager manages the permissions that are given to an Application.

2.3.1.9. Grant/Deny

When an app asks for a resource the resource is given to it if the permissions that are given include the permit to use that particular resource else the resource is denied access and blocked from asking for the resource again or trying to access that resource.

2.3.1.10. Warnings and Popups

Warnings or pops appear to show if an app is granted a permission or denied a permission. The popups appear when there is a security breach and a resource is being accessed to which the permission is not initially provided. Popups also appear when an Auto system security check is completed and a report is generated. Popups also appear to tell when a security check reveals a security breach.

2.3.2. UML Activity diagrams

This section shows the activities that a user need to preform to accomplish a task.

2.3.2.1. Install Apk

Description: This scenario describes the flow of activities necessary for the user to install a new app in Android phone.

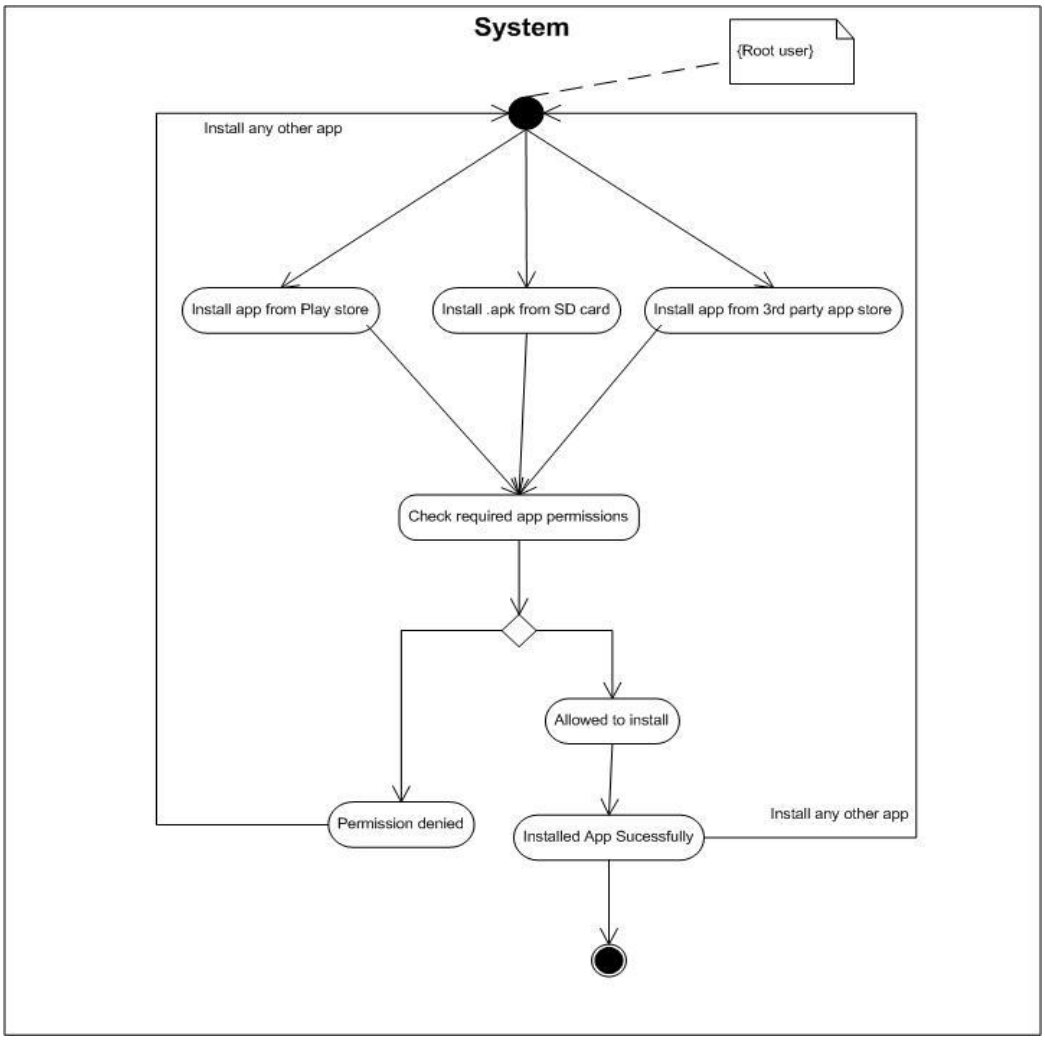

Figure 9 – Install App Activity

2.3.2.2. Create Profile

Description: This scenario describes the flow of activities necessary for the Superuser to create a new child profile.

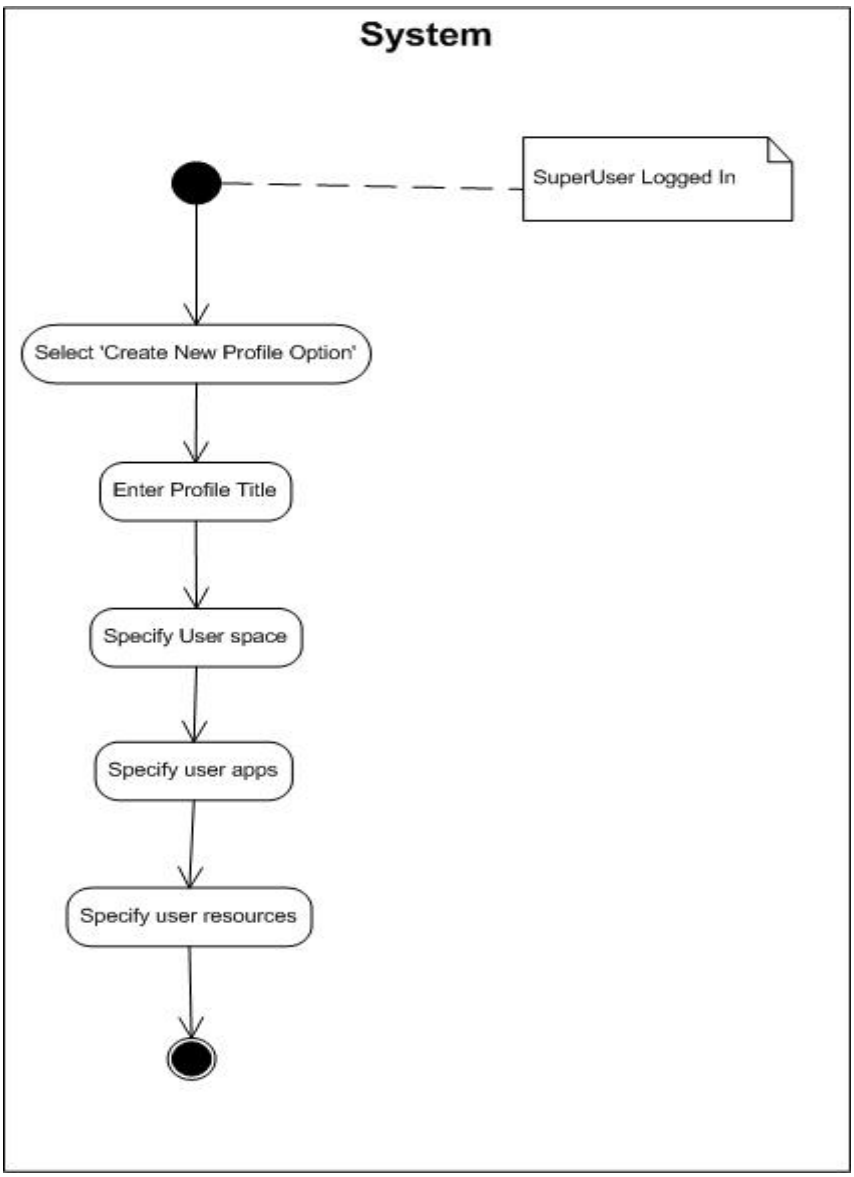

Figure 10 – Create child Profile Activity

2.3.2.3. Manage User Profiles

Description: This scenario describes the flow of activities required to interact with the system in the event of either creating new profile or modifying settings of existing child profiles.

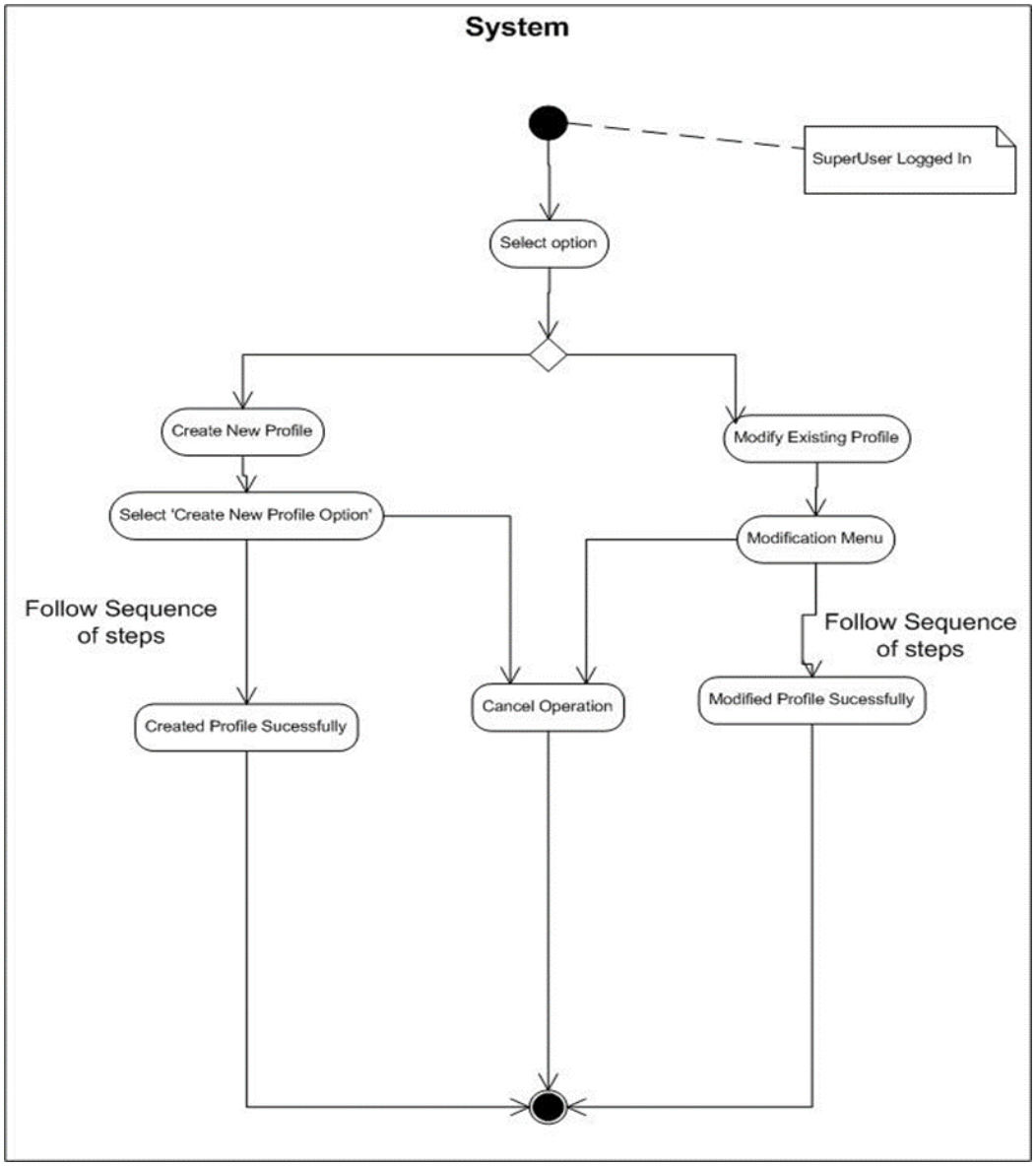

Figure 11 - Manage User Profiles Activity

2.3.2.4. Modify Profile

Description: This scenario describes the flow of activities the system performs in order to modify the settings of existing child profile.

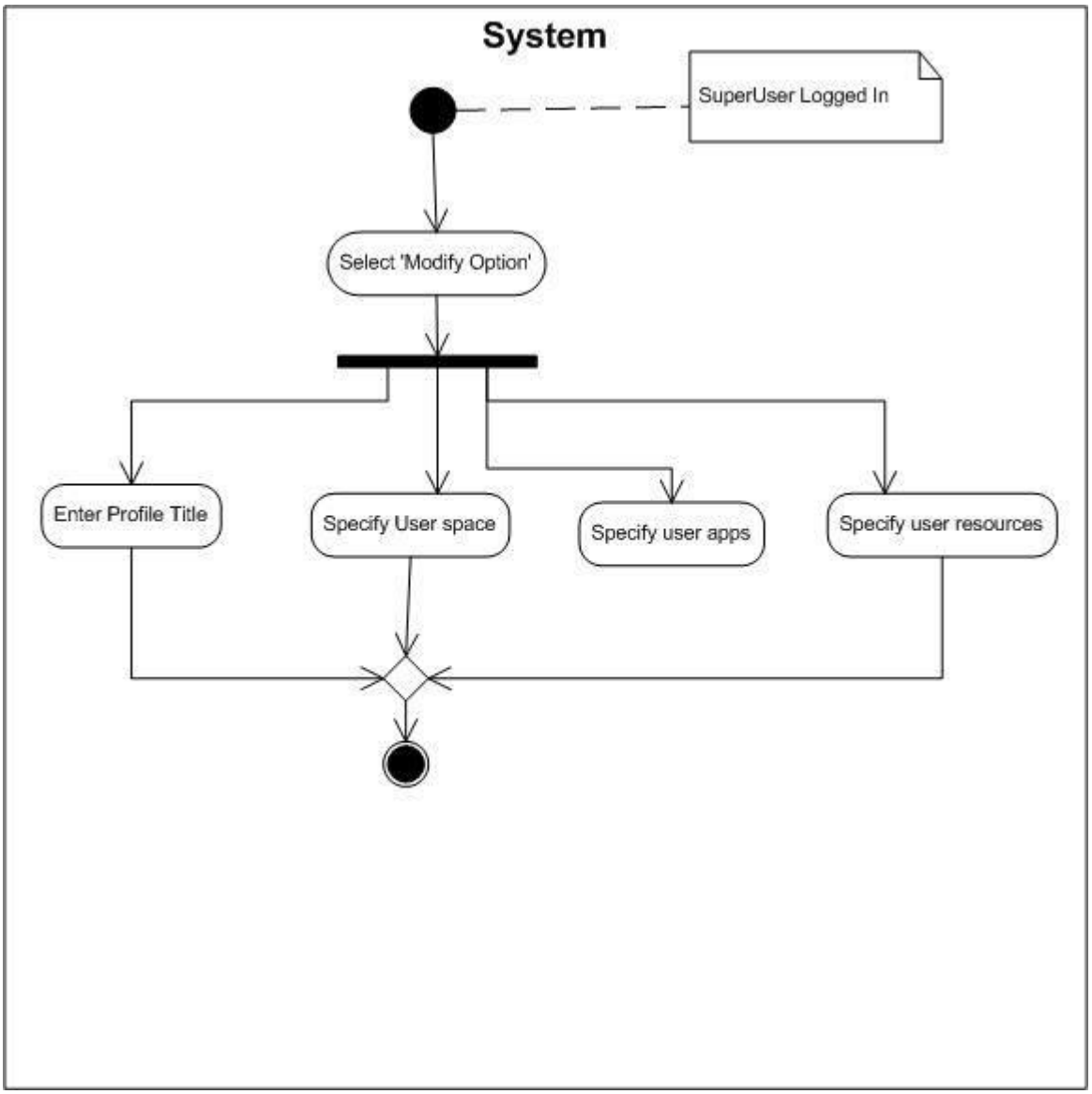

Figure 12 - Modifying Profile Activity

2.3.2.5. Manage Certificates

Description: This scenario describes the flow of activities required to enable/disable/add/ delete certificates.

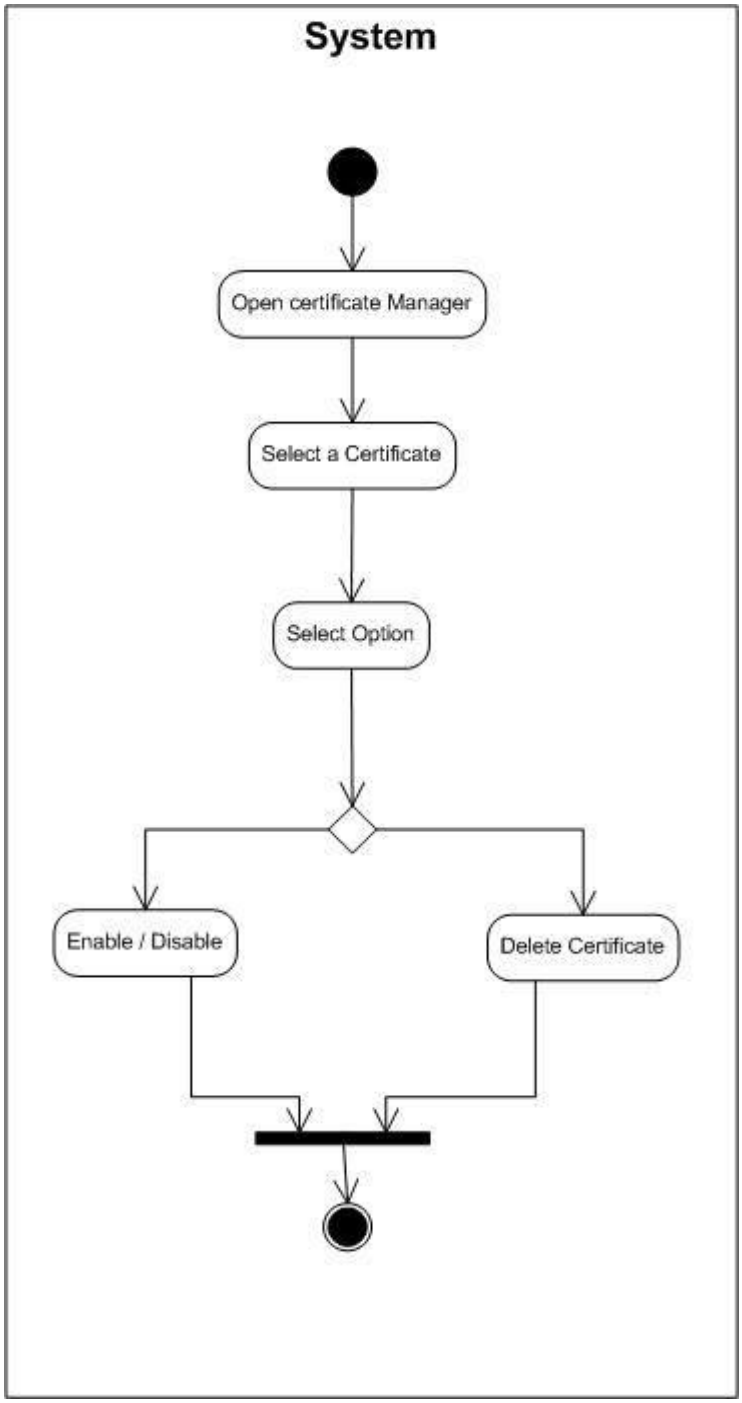

Figure 13 – Manage Certificates Activity

2.3.3. UML State-transition Diagrams

2.3.3.1. Create Profile

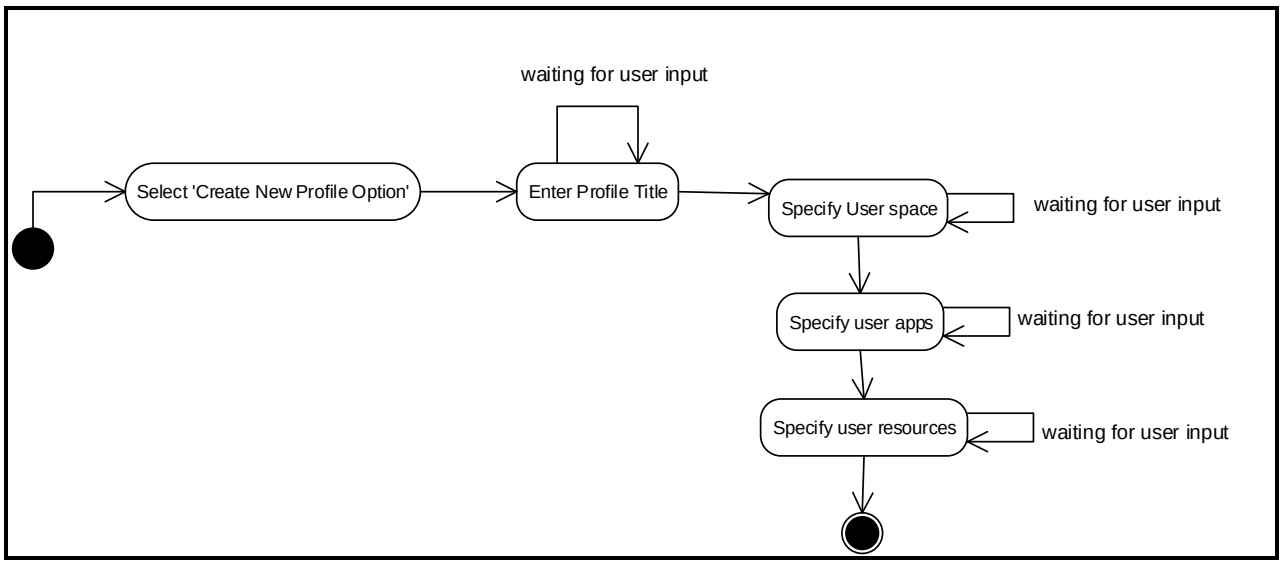

Figure 14 – Create Profile state transition

2.3.3.2. Install Apk

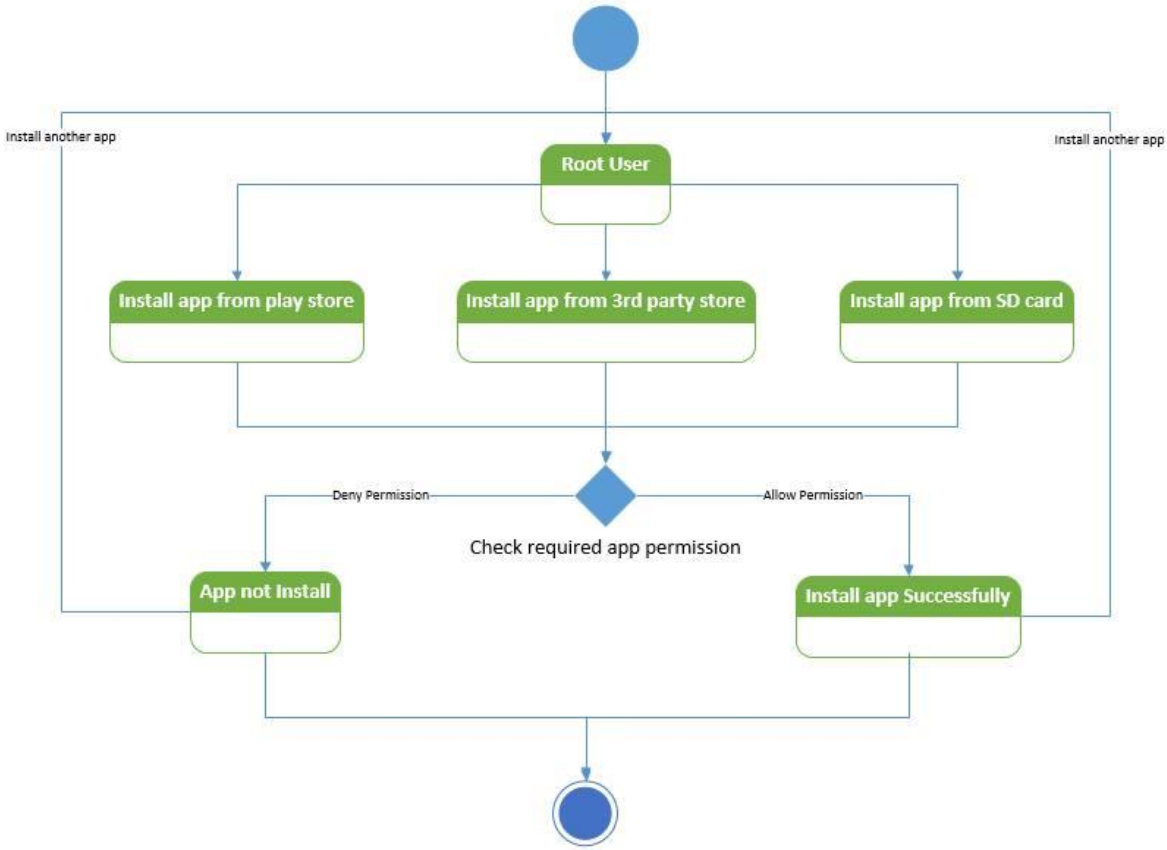

Figure 15 – Install Apk State Transition

2.3.3.3. Modify Profile

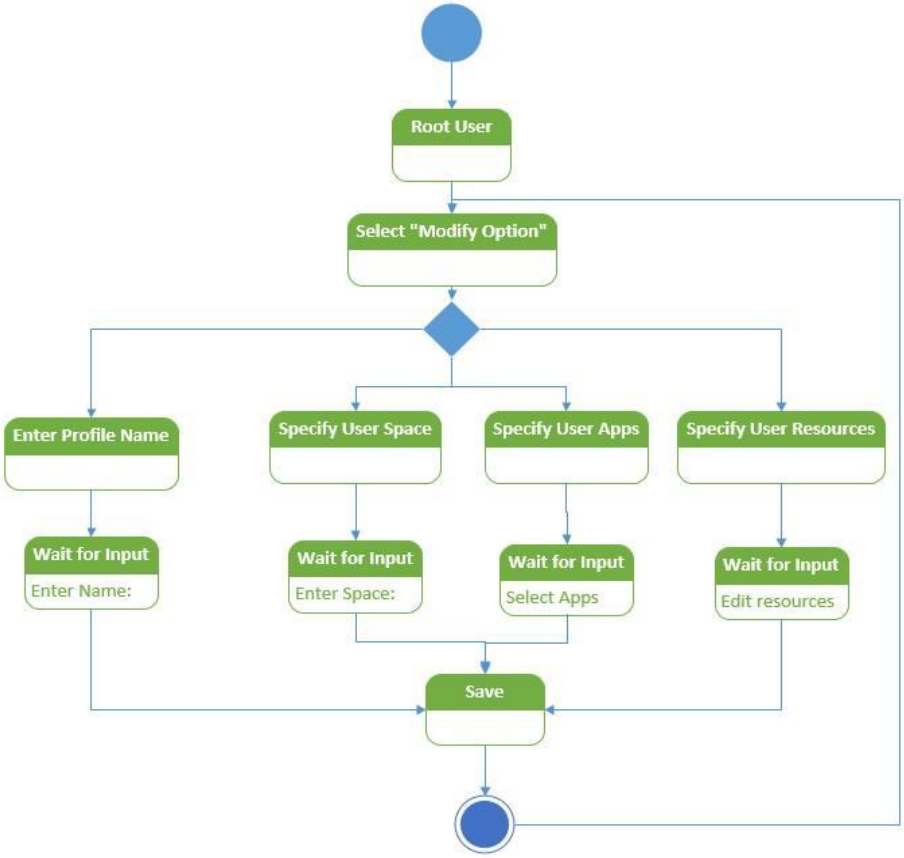

Figure 16 –Modify Profile State Transition

2.3.3.4. Manage Certificates

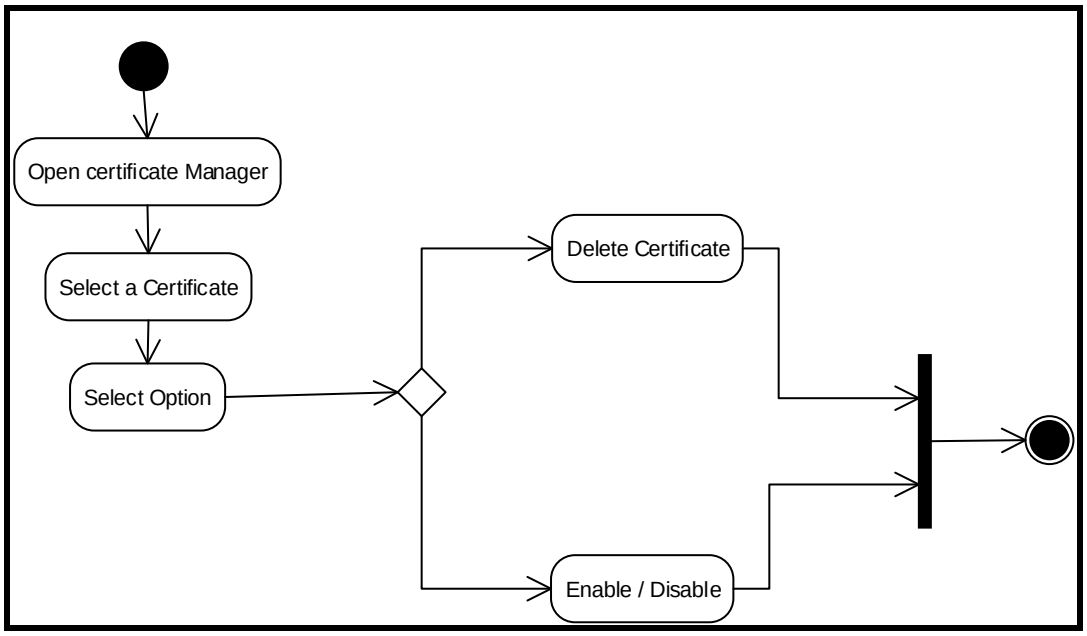

Figure 17 – Manage Profile State transition

2.3.4. UML Sequence diagrams

Different Scenarios and their corresponding events are discussed in this section with the help of sequence diagrams.

2.3.3.1. Create child Profile

Description: This scenario describes the sequence of events that take place when Super user intends to create a new child profile.

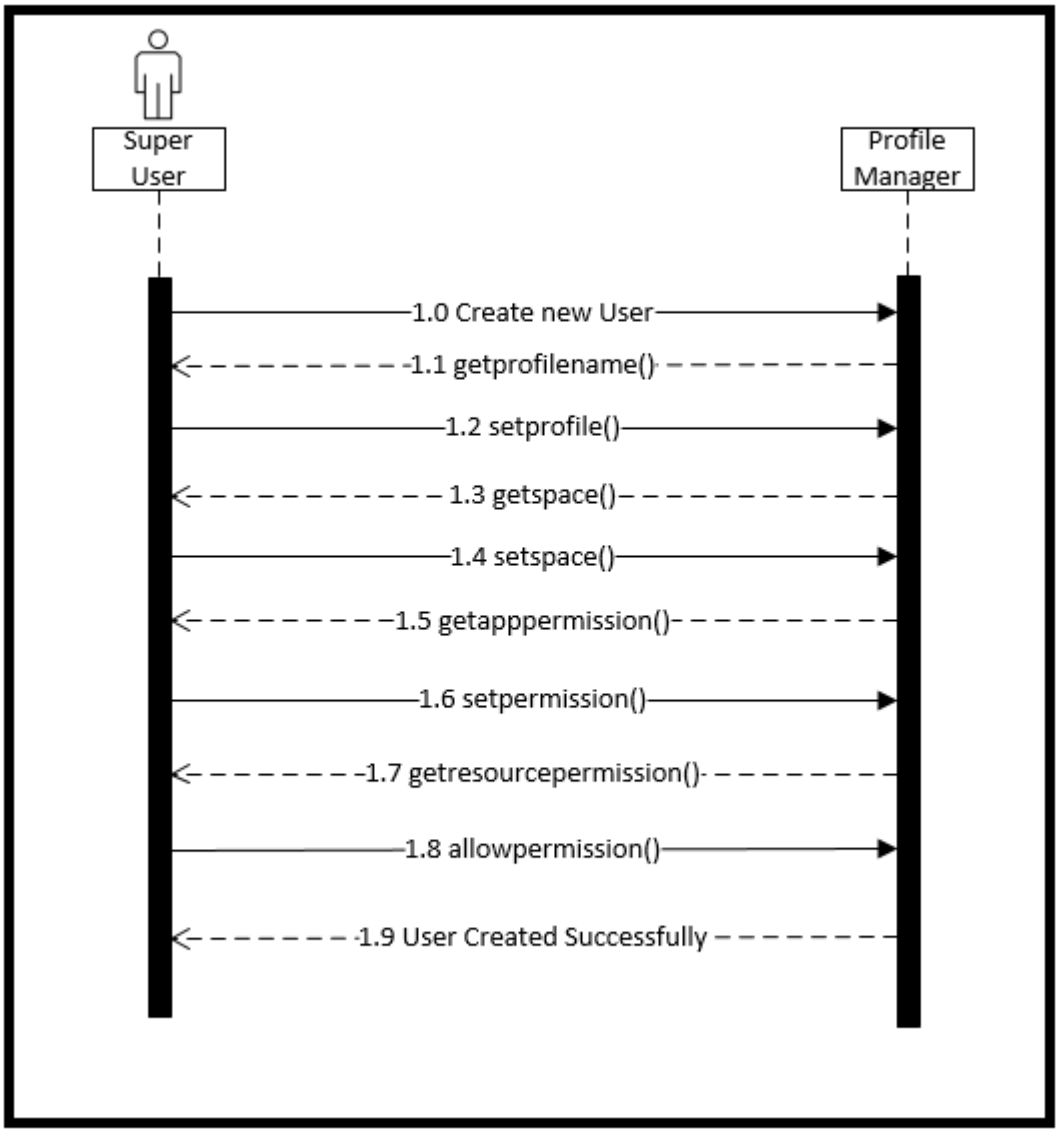

Figure 18 – Create Child Profile Sequence

2.3.4.1. Edit Child profiles

Description: This scenario describes the sequence of events that take place when Superuser intends to edit child profile(s), including name, resource and space; etc.

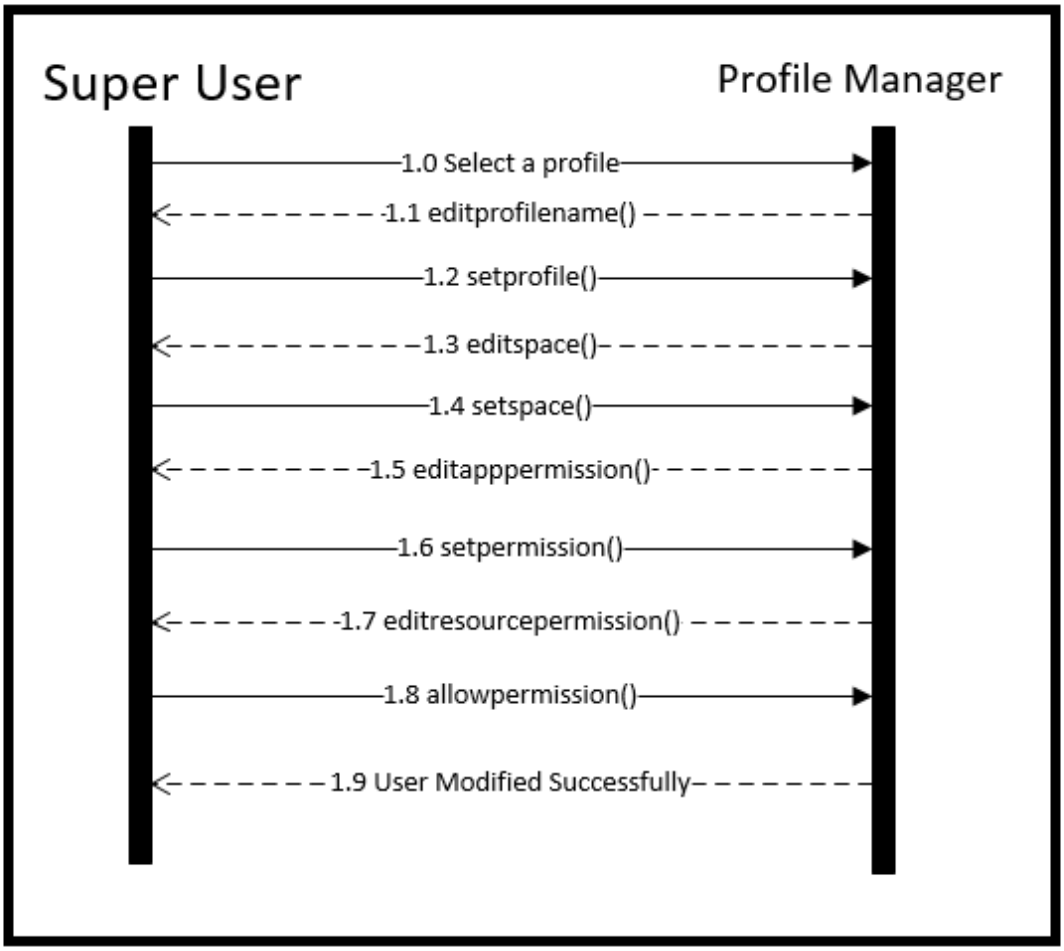

Figure 19 - Edit Profile Sequence

2.3.4.2. Certificate Manager

Description: This scenario describes the sequence of events that take place when the user intends to either enable, disable certificates using certificate manager.

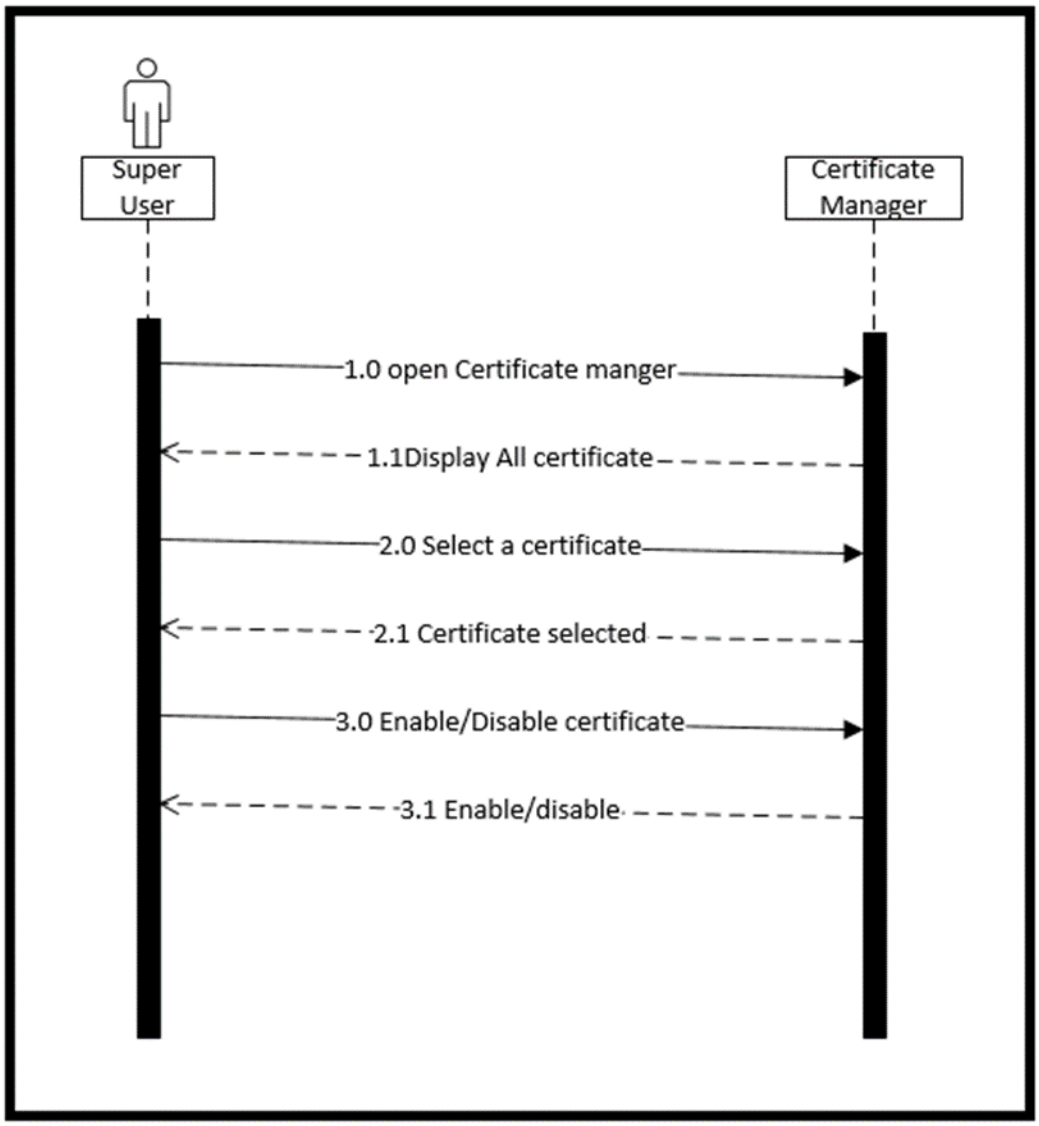

Figure 20 - Service Management Sequence

2.3.4.3. Runtime Security check

Description: This scenario describes the sequence of events that take place when the system performs runtime security checks on the Secure Android

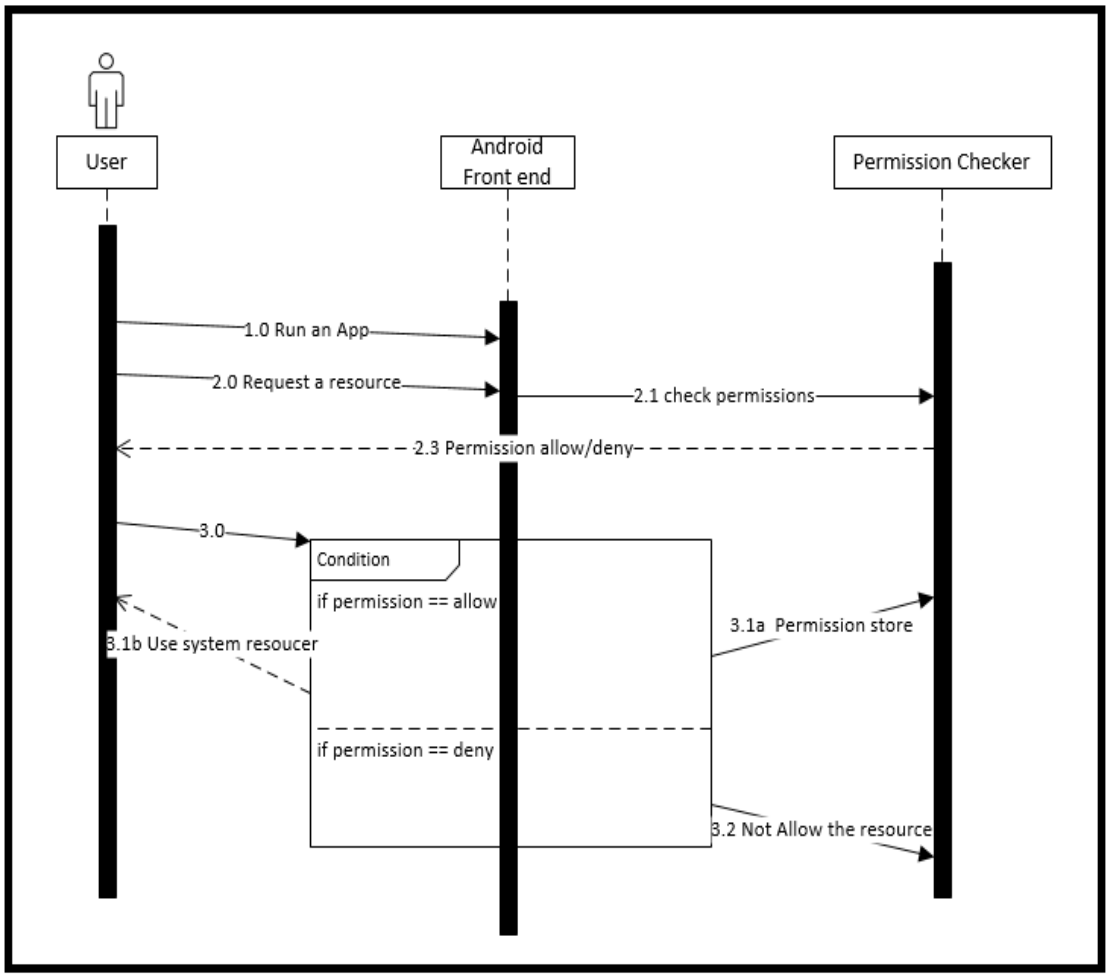

Figure 21 – Runtime Security checks Sequence

2.3.5. UI Design

Secure Android is an Android architecture intended to be used by the users from diverse background knowledge. This requires that the interface of Secure Android should have an easy learning curve for the user. Most of the important features should be visible to the user.

2.3.5.1. Settings Menu

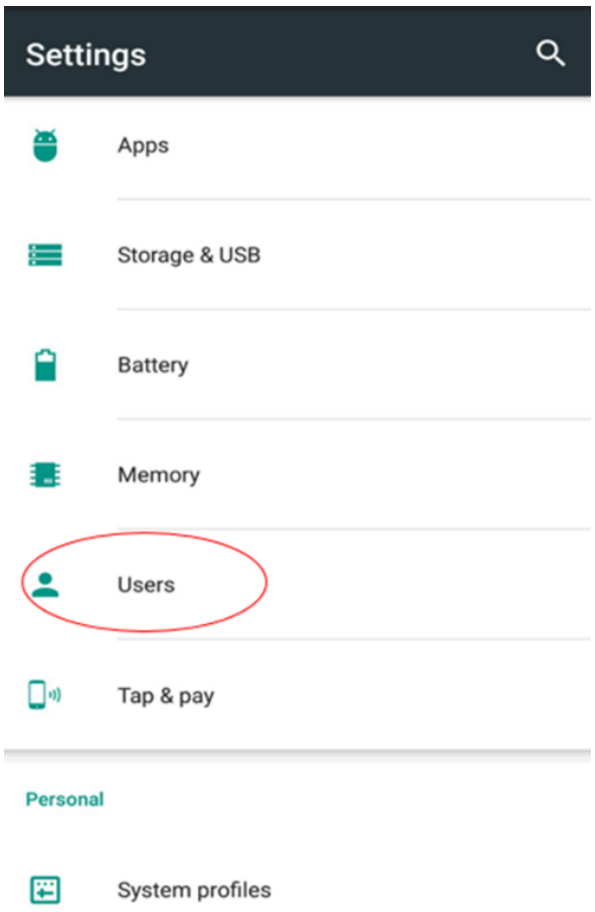

Figure 22 - Menu - UI Design

2.3.5.2. Profile selection menu on System Boot up

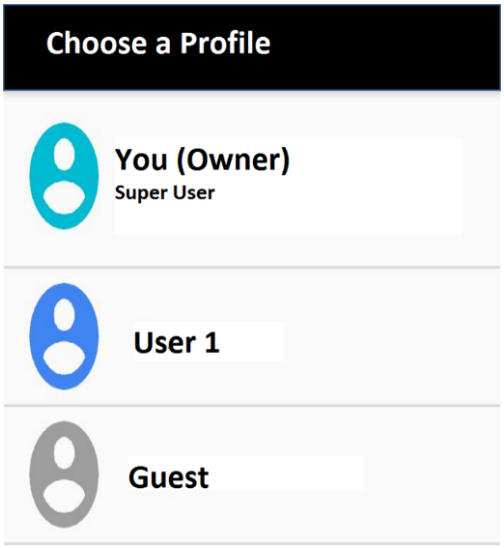

Figure 23 - Profile Selection Menu - UI Design

2.3.5.3. Runtime Access Permission

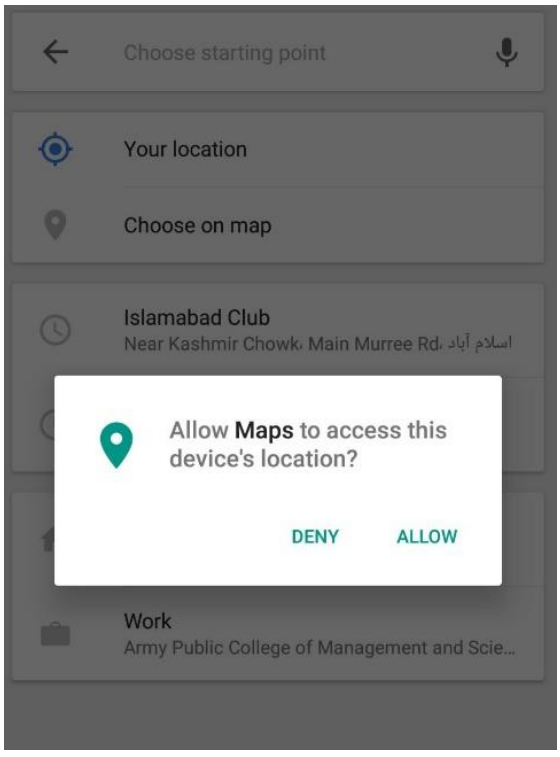

Figure 24 – Runtime Permission Dialog – UI Design

2.3.5.4. Apk Permissions

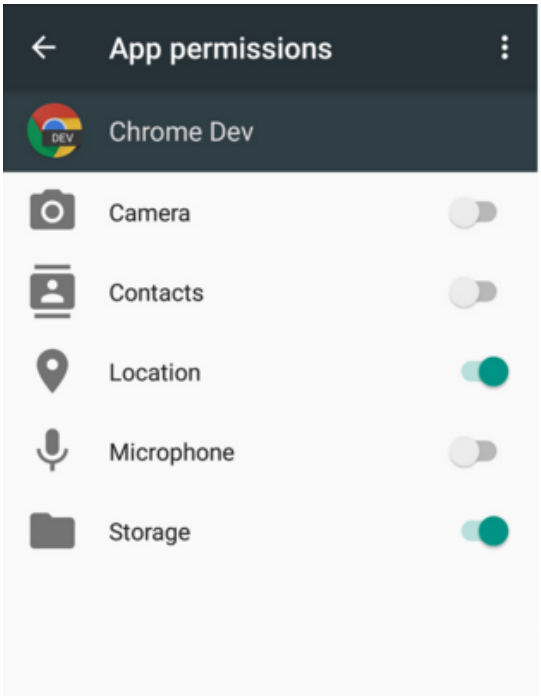

Figure 25 - Apk Permissions Manager- UI Design

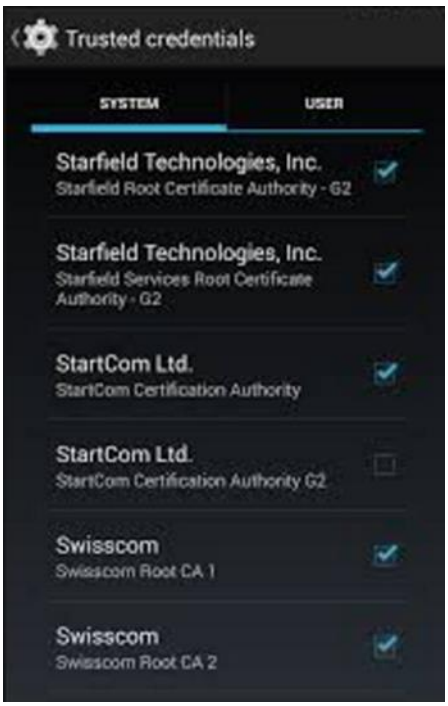

Figure 26 – Certificate Manager - UI Design

3. DETAILED DESCRIPTION OF COMPONENTS

3.1. Settings Menu

Identification	Android Settings Menu
Type	Component
Purpose	User interface to access different settings of the system. Separate menu class instances for each menu screen.
Function	Displays a window of buttons representing different menu choices. For each menu choice, the menu class will call the function "display (button)" to display button onscreen.
Subordinates	An array will hold different buttons, one for each option of the menu.
Dependencies	Calls instances of the button class e.g. , Users
Interfaces	OnScreenTap() [calls the button's callback function]
Resources	An Android-based smart phone
Processing	Display (button) function will display the buttons onscreen
Data	Array of button classes that will be used to hold different buttons in the Menu.

3.2. Certificate Manager

Identification	Certificate manager
Type	Module
Purpose	Allow the user to modify certificates Graphic user interface representation of certificates.
Function	Represents and harnesses delete or enable/disable options available to the user buttonCallback(button) Callback function for when and which button is tapped.
Subordinates	A bitmap will be used for the button's graphic.

Dependencies	Depends on the Settings menu
Interfaces	button->text() [text overlay on the button] button->bitmap() [graphic for the button] button->screenX(), button->screenY() [screen coordinates for the button]
Resources	Database of stored certificates in your phone.
Processing	buttonCallback(button) it's a callback function for the button.
Data	A database of Certificates and a bitmap to hold the visual graphic of the button.

3.3. App Permissions Manager

Identification	Permission Manager
Type	Module
Purpose	Holds all the permissions granted to a particular user profile.
Function	Loads app permissions.
Subordinates	Bitmap images will be used for the button and icon graphic.
Dependencies	Other apps are dependent on this module for getting permissions to access system resources.
Interfaces	LoadBitmap (bitmap_image, width, height) loads the bitmap with dimension width X height Y.
Resources	Bitmap images to be used as icons.
Processing	LoadBitmap (name, x, y) loads resource bitmap on the button with respect to the given dimensions.
Data	Permissions record, Resource bitmap.

3.4. User Manager

Identification	User Management
Type	Module

Purpose	To manage a set of User Profiles. It will manage the creation of child Users, their set of granted permissions and set of Apps allowed in their profile(s). All of this functionality is hidden from the child user - the nitty-gritty details of managing the users are the job of the SuperUser class.
Function	Represents Usermanagement options to the user, i.e. getname ( ) getspace() getresourcepermission() getapppermission()
Subordinates	Create, modify and delete; done by super user
Dependencies	All Usermanagement functions depend on it.
Interfaces	Create(user) Modify(user) Delete(User) Switch(user)
Resources	Database
Processing	signup ( ) signs up the user by valid credentials for onward login login ( ) authentication of the valid user for allocating access rights logout ( ) exits the user from the system editUser (user) enables user to edit user information getLocation ( ) fetches the location of the service being offered or sought searchService ( ) looks for a particular service
Data	Graphics for interface support. Database repository for maintaining the user record.

3.5. Inputs from User(s)

Identification	Inputs
Type	Data input

Purpose	To take actions according to the input provided by the users i.e., profile selection, app permission checking,.. etc)
Function	Get_camera_access ( ) getmapLocation ( ) allowapp( ) denypermission() and many other similar to these
Subordinates	None
Dependencies	All decisions made by secure Android depends on this
Interfaces	Dialog boxes, pop ups, input text boxes
Resources	Android Operating System resources(memory, graphics etc)
Processing	Get_camera_access ( ) request made by an app to permission checker to get an access to camera. getmapLocation ( ) request made by an app to permission checker to get an access to current location in the map. allowapp( ) pop up which asks superuser either to allow an app to be included in a child profile. denypermission() a declaration message which shows that requested permission is denied. and many other similar to these function calls.
Data	Services repository.

4. REUSABILITY & RELATIONSHIPS TO OTHER PRODUCTS

The Secure Android is a new product, Security enhancement is done on already existing operating system of Android lollipop. Therefore, the whole Android lollipop operating system is being reused. Secure Android has potential to be enhanced and have more features added to it. Therefore, right from the architecture, the system is being designed in a modular fashion with a view to have more cohesion and less coupling among the sibling modules.

5. DESIGN DECISIONS AND TRADEOFFS

- Considering the system to be first of its kind in Pakistan, the basic architecture design of Android operating system has been employed in order to leave a room for the future Reusability and enhancement aspects of the modules.
- A simpler user interface of the system has been incorporated, so as to extend the usability of the system to all the novice users of Secure Android
- A mechanism of app permissions has been enforced only to get the users verified and uniquely identified on the app domain.
- The goal is to prevent external mobile apps from granting themselves extra privileges, prevent apps from sharing too much data and to prevent the bypass of security features.

6. PSEUDO CODE FOR COMPONENTS

6.1. User Profile Creation

```
Begin
    If
        SuperUser is logged in
        Check User availability
            If user available
                Create user
                Enter all the credentials
                User created
            else
            Maximum number of user already created.
    else
        Switch/Login to the Super user and create user.
```

End

6.2. Profile Selection

Begin

Phone Boot Up

Display Profile selection menu

If CurrentUser is SuperUser(Admin)

Enter the Password

If password matched

Super user account will open

else

Incorrect Password.

Back to profile selection menu.

else

Back to profile selection menu.

End

6.3. Application Permissions

Begin

Super user or Child user already logged in

Want to use System Resources

If permission is already denied

Display option dialog:

1- Deny

2- Allow

If Allow

got permission to access resource*

ElseIf deny

Resource not granted

End

6.4. Modify profile

Begin

Logged in as super user

Select Child Profile

Select Modify

If edit Profile Title

Enter Profile name

If edit user space

Enter space

If edit user apps

Select apps

If edit user resources

Select resources for user

End

6.5. Certificate Manager

Open Certificate Manager

Select a certificate

Press Enable/Disable Button

If disabled already

Enable it

Else if Enabled already

Disable it

If delete

Certificate Deleted

End

6.6. Apk Installation Permission

Super User Logged In

Begin

Want to install an app

Select app from google store/get apk/get from 3rd party app store

Press install

Open Dialogue with system resource permission*

1- Allow

2- Deny

If allow

System resource usage permission granted.

If Deny

System resource permission will deny to this particular application.

App installed.

End

7. APPENDICES

7.1. Appendix A: Glossary

7.1.1. Android

An operating system designed for mobile devices (i.e. cell phones, tablet computers) by Google, Inc.

7.1.2. App (Application)

An application program (app or application for short) is a computer program designed to perform a group of coordinated functions, tasks, or activities for the benefit of the user. Examples of an application include a word processor, a spreadsheet, an accounting application, a web browser, a media player, an aeronautical flight simulator, a console game or a photo editor.

7.1.3. Recourse (as Referenced for an Android Device)

The resource of a mobile phone include the camera, the internal memory, GPS, Wi-Fi, Mobile network data connection, contacts, messages and other personal files.

7.1.4. Permission

The permissions are controls that we give to an application for using the resources of our mobile phone. Permissions are when we allow or deny an application from accessing e.g. The camera.

7.1.5. Certificate

All APKs (.apk files) must be signed with a certificate whose private key is held by their developer. This certificate identifies the author of the application. The certificate does not need to be signed by a certificate authority; it is perfectly allowable, and typical, for Android applications to use self-signed certificates. The purpose of certificates in Android is to distinguish application authors. This allows the system to grant or deny applications access to signature-level permissions and to grant or deny an application's request to be given the same Linux identity as another application.

7.1.6. Profile

The concept of Profiles is implemented to separate different users and there control on the device there are two types of profiles mentioned in the document Superuser and child user Superuser has all the controls of the device and child user has limited control of the device that is allowed by the superuser to the child user.

7.1.7. Apk

Android application package (APK) is the package file format used by the Android operating system for distribution and installation of mobile apps and middleware.

7.2. Appendix B: Use Cases Description

This section lists the Use Cases for Secure Android. The various user classes identified the following Use Cases and primary actors for the Secure Android:

Actors	Use Cases
Superuser	Create Child Profile(s) Edit permissions of child profile(s) Activate Child Profile(s) Use Android Apps Manage Certificates Install Apk(s)

User	Use Android App(s) Install Apk(s)
Intruder	Security Breach

7.2.1. Create Child Profile

Use Case ID:	1		
Use Case Name:	Create Child Profile		
Actors:	Superuser		
Created By:	Maryam	Last Updated By:	Maryam
Date Created:	17/12/2015	Date Last Updated:	17/12/2015
Description:	The superuser is the only actor that can access the privilege to create a child profile.		
Preconditions:	Actor must be superuser or have all the access that a superuser has		
Post conditions:	A new user profile is created.		
Normal Flow (primary scenario):	The actor enters all the information needed allocates specific space allows permissions.		
Alternative Flows:	None		

7.2.2. Edit permissions of child profile(s)

Use Case ID:	2		
Use Case Name:	Edit permissions of child profile(s)		
Actors:	Superuser		
Created By:	Maryam	Last Updated By:	Maryam

Date Created:	17/12/2015	Date Last Updated:	17/12/2015
Description:	The superuser edits the permissions that are provided to a child profile.		
Preconditions:	The child profile must exist first in order for its permissions to be edited.		
Post conditions:	If the use case was successful, the actor has given the child profile some desired permissions.		
Normal Flow (primary scenario):	The child actor request a resource to be used for which he doesn't have permission. The superuser gets the request and edits the permissions of the child user according to the demands.		
Alternative Flows:	If the superuser doesn't want to give the requested permission , the request is denied and the permissions of the user remain the same.		

7.2.3. Activate child Profile

Use Case ID:	3		
Use Case Name:	Activate Child Profile		
Actors:	Superuser		
Created By:	Maryam	Last Updated By:	Maryam
Date Created:	17/12/2015	Date Last Updated:	17/12/2015
Description:	A user tries to switch from one profile to the other.		
Preconditions:	User has to be a superuser.		
Post conditions:	The new profile is activated and the previous one is inactive now for the current device.		
Normal Flow (primary scenario):	The superuser wants to check or access a child profile The superuser switches to the other profile and the previous profile is deactivated. The superuser requires a password to change back to the superuser profile.		

Alternative Flows:	None

7.2.4. Use Android App

Use Case ID:	4		
Use Case Name:	Use Android App		
Actors:	Superuser, User		
Created By:	Afifa	Last Updated By:	Afifa
Date Created:	17/12/2015	Date Last Updated:	17/12/2015
Description:	A user tries use an Android application.		
Preconditions:	User has install the application first.		
Post conditions:	The System must show the profile and record the changes made therein if the changes were made in a profile other than the superuser profile.		
Normal Flow (primary scenario):	Actor opens the app Actor uses the app.		
Alternative Flows:	The Actor installs the app. The Actor opens the app The Actor uses the app.		
Alternative Flow 2:	The Actor installs the app The Actor opens the app The app crashes		
Alternative Flow3:	The Actor opens the app The app crashes		

7.2.5. Manage Certificates

Use Case ID:	5

Use Case Name:	Manage Certificates		
Actors:	Superuser		
Created By:	Afifa	Last Updated By:	Afifa
Date Created:	17/12/2015	Date Last Updated:	17/12/2015
Description:	Actor will Manage certificates		
Preconditions:	There must be applications for which certificates are provided.		
Post conditions:	The actor is able to enable disable and delete certificates.		
Normal Flow (primary scenario):	This use case starts when the actor wishes to enable, disable or delete a certificate. The actor successfully enables, disables or deletes a certificate.		
Alternative Flows:	No Applications Installed In case there are no applications installed on the device and as a result there are no certificates available for the certificate manager to manage		

7.2.6. Install Apk

Use Case ID:	6		
Use Case Name:	Install Apk		
Actors:	User, Superuser		
Created By:	Afifa	Last Updated By:	Afifa
Date Created:	17/12/2015	Date Last Updated:	17/12/2015
Description:	The system will enable the actor to install an apk into the secure Android.		
Preconditions:	The user has to have an apk file. If the actor is user then it must be allowed by the super user to install an apk.		
Post conditions:	The app must be successfully installed and the app should be fully functional.		

Normal Flow (primary scenario):	Actor installs the apk. The app has all the required permissions for proper functioning. The app is fully functional
Alternative Flows:	The actor installs the apk The App is not able to install.

7.2.7. Security Breach

Use Case ID:	7		
Use Case Name:	Security Breach		
Actors:	Intruder		
Created By:	Shahid	Last Updated By:	Shahid
Date Created:	17/12/2015	Date Last Updated:	17/12/2015
Description:	Intruder will try to access the device.		
Preconditions:	The user must not have any permissions to do the task he/she is doing		
Post conditions:	The owner(Superuser) of the device should be notified through a popup or a security warning.		
Normal Flow (primary scenario):	The intruder tries to access the device without the permission of the superuser by finding a loophole. The intruder might use an app , try to manipulate the certificates of the apps and might try to install new apps or apks on to the device The superuser is notified about the security breach through a security warning or popup.		
Alternative Flows:	The intruder tries to access the device without the permission of the superuser by finding a loophole. The intruder is blocked because it doesn't have enough permissions to access the device.		

7.2.8. Display Security Warning

Use Case ID:	8		
Use Case Name:	Display Security Warning		
Actors:	Intruder		
Created By:	Shahid	Last Updated By:	Shahid
Date Created:	17/12/2015	Date Last Updated:	17/12/2015
Description:	The system notifies the user about the security breach.		
Preconditions:	The Security of the system is breached.		
Post conditions:	A popup or security warning notifies the Superuser about the security breach.		
Normal Flow	The security is breached The display popup or security warning shows the user that a security breach has occurred and the security of the system has been compromised.		
Alternative Flows:	The security is breached The threat is eliminated and the intruder is blocked. The display popup or security warning shows the user that a security breach has occurred and the security of the system has been compromised.		