

PUCIT-Project Coordination Office	Version: 1.0
Final Project Proposal	Date: 12 June, 2015



# **PUCIT**

## **Punjab University College of Information Technology**

---

## **Final Documentation Format Guidelines**

**Version 1.0**

---

## **Vulnerable Apps Detector App**

1



**Team ID BCSF11-20**

**Session: BSCS. Spring 2011**

**Project Advisor: Madam Zobia Suhail**

**Submitted By**

Zaheer Ahmed

BCSF11M044

Talha Sajjad

BCSF11M029

Zeeshan Tahir

BCSF11M008

---

Punjab University College of Information Technology  
University of the Punjab, Lahore.

PUCIT-Project Coordination Office	Version: 1.0
Final Project Proposal	Date: 12 June, 2015

## STATEMENT OF SUBMISSION

This is to certify that **Zaheer Ahmed** Roll No. **BCSF11M044** and **Talha Sajjad** Roll No. **BCSF11M029** and **Zeeshan Tahir** Roll No. **BCSF11M008** have successfully completed the final project named as: **Vulnerable App Detactor App (Android Version)**, at the Punjab University College of Information Technology, University of The Punjab, Lahore, to fulfill the partial requirement of the degree of **Belchers in Computer Science**.

\_\_\_\_\_  
Project Office Supervisor  
PUCIT, Lahore

\_\_\_\_\_  
Project Primary Advisor  
Name: Write name of Project advisor here  
Designation: Write designation of Project Advisor here  
PUCIT

\_\_\_\_\_  
Project Examiner  
Name: Write name of Project Examiner here  
Designation: Write designation of Project Examiner here  
PUCIT

PUCIT-Project Coordination Office	Version: 1.0
Final Project Proposal	Date: 12 June, 2015

## Proofreading Certificate

It is to certify that I have read the document meticulously and circumspectly. I am convinced that the resultant project does not contain any spelling, punctuation or grammatical mistakes as such. All in all I find this document well organized and I am in no doubt that its objectives have been successfully met.

---

Mam Fozia Adeeba  
Business Communication and Technical Writing,  
Lecturer, PUCIT

PUCIT-Project Coordination Office	Version: 1.0
Final Project Proposal	Date: 12 June, 2015

## Acknowledgement

First of all thanks to ALLAH who let us able to work on this Project. We truly acknowledge the cooperation and help make by Name of **Madam Zobia Suhail, Senior Female Faculty, Lecturer of Punjab University Collage of Information Technology**. She has been a constant source of guidance throughout the course of this project. We are also thankful to our families whose silent support led us to complete our project.

**1- Madam Zobia Suhail**

Date:

June 12, 2015

PUCIT-Project Coordination Office	Version: 1.0
Final Project Proposal	Date: 12 June, 2015

## Abstract

Due to ever increasing demand of Security in Communication Devices like Cause our devices stores Data of our Internet Accounts, Bank Accounts and Credit Card Info or maybe data in cell phone which contain personal contents so we choose our Final year project to reduce risks of security violation in Android Devices. We provide our user an user friendly interface to interact with our App and scan for those Android App which are Violating Security Protocols. With our App user can scan and get those App which are Spying his data in any aspect. We make sure our Security App can remain active all the time in user's system from time when he/she boots his device after installing our App. This project not only just working anyway but also we get assure to use minimum system space, memory and processing of user's device.

PUCIT-Project Coordination Office	Version: 1.0
Final Project Proposal	Date: 12 June, 2015

## Preface

As a Computer Scientist our point of view on Security and Data Theft is different from simple users. We know current needs of security in different kind of systems. Computer operating systems industry is mature too much. They fulfill security needs time to time but mobile phone industry is rising and it's Operating systems also free (Linux based) so no extra ordinary Security tools built-in came with any Android Version. It's on user to secure his data by using Apps like our Tool. iPhone, Blackberry and Windows phone having own Operating Systems and uses own App Stores to protect their user from fake or vulnerable Apps. Android is open to everyone for uploading their App so a Cyber Attacker can get user's Data, Accounts Information and also can use user's device as a dummy in Botnet Attack to a website. Our group members also having interest in security field so we choose to develop this Security App for android based mobile devices. Thanks to our supervisor **Madam Zobia Suhail** and friends specially **Junaid Surfraz** they give us suggestions to efficiently do our project.

## Table of Contents

<b>I-Abstract .....</b>	<b>5</b>
<b>II- Preface .....</b>	<b>6</b>
<b>1-Project Title .....</b>	<b>8</b>
<b>2- Feasibility REPORT .....</b>	<b>8</b>
2.1 Technical Feasibility .....	8
2.2 Operational Feasibility .....	8
2.3 Economic Feasibility .....	8
2.4 Schedule Feasibility .....	8
2.5 Specification Feasibility.....	9
2.6 Information Feasibility.....	9
2.7 Motivational Feasibility .....	9
2.8 Legal & Ethical Feasibility .....	9
<b>3- Project Scope.....</b>	<b>9</b>
<b>4- Project Overview Statement .....</b>	<b>10</b>
<b>5-Project Costing .....</b>	<b>10</b>
5.1 Project Cost Estimation by using COCOMO'81 (Constructive Cost Model) ....	10
<b>6- Critical Path Method Analysis (CPM Analysis).....</b>	<b>13</b>
<b>7- Gantt Chart.....</b>	<b>16</b>
<b>8- Introduction to team members .....</b>	<b>16</b>
<b>9- Tools and Technologies .....</b>	<b>17</b>
<b>10- Vision Document.....</b>	<b>18</b>
<b>11-Risk List.....</b>	<b>18</b>
<b>12- Requirements Engineering .....</b>	<b>18</b>
<b>13- Use Case Descriptions .....</b>	<b>21</b>
<b>14- Use Case Diagram .....</b>	<b>23</b>
<b>15- Domain Model .....</b>	<b>24</b>
<b>16- Sequence Diagrams .....</b>	<b>26</b>
<b>17- Collaboration Diagrams .....</b>	<b>30</b>
<b>18- Operation Contracts .....</b>	<b>33</b>
<b>19- Design Class Diagram .....</b>	<b>35</b>
<b>20- Data Model .....</b>	<b>37</b>



PUCIT-Project Coordination Office	Version: 1.0
Final Project Proposal	Date: 12 June, 2015

## 1-Project Title

Vulnerable Apps Detector App

## 2- Feasibility Report

When a project is started the first matter to establish is to assess the feasibility of a project or product. Feasibility means the extent to which appropriate data and information are readily available or can be obtained with available resources such as staff, expertise, time, and equipment. It is basically used as a measure of how practical or beneficial the development of a software system will be to you (or organization). This activity recurs throughout the life cycle.

There are many types of feasibilities:

- Technical
- Operational
- Economic
- Schedule
- Specification
- Information
- Motivational
- Legal and Ethical

### 2.1 Technical Feasibility

With respect to Technical Feasibility of this Project we are dealing with Android features of Installing Apps. We can count total Apps installed in System to run our system after that we just have to collect data from each App's Manifest file so this is what we gonna do which is possible in any Android OS & Devices.

### 2.2 Operational Feasibility

Before working on this Project we (Team Members) are Strong enough for finding Logic and also Developing it. All members of team are of 7<sup>th</sup> semester and also studying/studied

- Mobile Computing(Android)
- EAD(Enterprise Application Development)
- Software Engineering (Subject)

so it's not a big deal to working with Project with help of these courses.

### 2.3 Economic Feasibility

The Cost of this Project is calculated below Cost Estimation heading. Team of this Project is going to manage this project on due time with enough functionality. We paid enough time to maintain this as compared to if this project is developed in a Software house.

## 2.4 Schedule Feasibility

Our Team is scheduling to complete this Project in due time. We had divided whole work to three parts

1. GUI (All Activities of Project)
2. Spy Apps Detection Module
3. Malware/Fake Apps Detection Module

First part GUI we will Code in this semester because we have to work for deliverable of this project and also other courses are here. Other two parts are Backup working of this Project that we will manage in 8<sup>th</sup> semester.

## 2.5 Specification Feasibility

This App just need file system Access as like as every Security Program. Secondly Access required to other Apps. This System can be installed on any Android OS and Device.

## 2.6 Information Feasibility

This is a Security System for Android real time required to protect user from those Scams they are stealing Non-Technical user's info with letting him play a game or an internet App etc.

## 2.7 Motivational Feasibility

This App in today's Need because *Information is worthless nowadays* so we are working on Security System. This is the only motivation for all group members because we are capable of Developing other Projects like simple **Metropolitan Traffic System** or maybe **Image Processing System**

## 2.8 Legal & Ethical Feasibility

Every Security System today is Legal and we also working on a Security App. This System can be used Against Owners of those Apps which are found to be Guilty in Scan Result so this is Totally legal and green for Cyber Environment.

## 3- Project Scope

This Project is for Android End user. User of this App will be facilitated with following features.

- 1: User can find malicious app's on his Android system.
- 2: User can see Apps with those permissions they are vulnerable for a user security.

Goals of this project are to secure Android Devices. With this App user will be able to scan his Device for some malicious App or if an App is using extra permissions which are irrelevant to App's work.

Objectives are following:

- We will perform Checks for those Apps which are using other App's Certificate as a fake certificate.
- Secondly this App will scan system for those Apps which using some extra permissions.

## 4- Project Overview Statement

This Project is for Android End user. User of this App will be facilitated with following features.

- 1: User can find malicious app's on his Android system.
- 2: User can see Apps with those permissions they are vulnerable for a user security.

## 5-Project Costing

A metric is some measurement we can make of a product or process in the overall development process. Metrics are split into two broad categories:

- Knowledge oriented metrics: these are oriented to tracking the process to evaluate, predict or monitor some part of the process.
- Achievement oriented metrics: these are often oriented to measuring some product aspect, often related to some overall measure of quality of the product.

Most of the work in the cost estimation field has focused on algorithmic cost modeling. In this process costs are analyzed using mathematical formulas linking costs or inputs with metrics to produce an estimated output. The formulae used in a formal model arise from the analysis of historical data. The accuracy of the model can be improved by calibrating the model to your specific development environment, which basically involves adjusting the weightings of the metrics.

### 5.1 Project Cost Estimation by using COCOMO'81 (Constructive Cost Model)

Boehm's COCOMO model is one of the mostly used models commercially. The first version of the model delivered in 1981 and COCOMO II is available now. COCOMO 81 is a model that allows one to estimate the cost, effort, and schedule when planning a new software development activity, according to software development practices that were commonly used in the 1970s through the 1980s. It exists in three forms, each one offering greater detail and accuracy the further along one is in the project planning and design process. Listed by increasing fidelity, these forms are called Basic, Intermediate, and Detailed COCOMO. However, only the Intermediate form has been implemented by USC in a calibrated software tool.

Three levels:

**Basic:** Is used mostly for rough, early estimates.

**Intermediate:** Is the most commonly used version, includes 15 different factors to account for the influence of various project attributes such as personnel capability, use of modern tools, hardware constraints, and so forth.

**Detailed:** Accounts for the influence of the different factors on individual project phases: design, coding/testing, and integration/testing. Detailed COCOMO is not used very often.

Each level includes three software development types:

1. **Organic:** Relatively small software teams develop familiar types of software in an in-house environment. Most of the personnel have experience working with related systems.

2. **Embedded:** The project may require new technology, unfamiliar algorithms, or an innovative new method
3. **Semi-detached:** Is an intermediate stage between organic and embedded types.

#### Basic COCOMO

Type	Effort	Schedule
<b>Organic</b>	<b>PM= 2.4 (KLOC)1.05</b>	<b>TD= 2.5(PM)0.38</b>
<b>Semi-Detached</b>	<b>PM= 3.0 (KLOC)1.12</b>	<b>TD= 2.5(PM)0.35</b>
<b>Embedded</b>	<b>PM= 2.4 (KLOC)1.20</b>	<b>TD=</b>
		<b>2.5(PM)0.32</b>

PM= person-month (effort)

KLOC= lines of code, in thousands

TD= number of months estimated for software development (duration)

#### Intermediate COCOMO

Type	Effort
Organic	PM= 2.4 (KLOC)1.05 x M
Semi-Detached	PM= 3.0 (KLOC)1.12 x M
Embedded	PM= 2.4 (KLOC)1.20 x M

PM= person-month

KLOC= lines of code, in thousands

M.- reflects 15 predictor variables, called cost drivers.

YOUR BASIC COCOMO RESULTS!!								
MODE	"A" variable	"B" variable	"C" variable	"D" variable	KLOC	EFFORT, (in person/months)	DURATION, (in months)	STAFFING, (recommended)
undefined	3.6	1.2	2.5	0.32	2	8.27062815597865	4.915326003877275	1.6826204710439692

Explanation: The coefficients are set according to the project mode selected on the previous page, (as per Boehm,81). The final estimates are determined in the following manner:

**effort** =  $a * KLOC^b$ , in person/months, with KLOC = lines of code, (in the thousands), and:


**duration** =  $c * effort^d$ , finally:

**staffing** =  $effort / duration$

For further reading, see Boehm, "Software Engineering Economics", (81)

**WARNING:** If you see "NaN" in any field above, you have entered an **INVALID** value for KLOC!! Hit the "BACK" button on your browser, hit the "RESET" button, and enter a **DECIMAL NUMBER** in the KLOC input text box!

*Thank you, and happy software engineering!*



The schedule is determined using the Basic COCOMO schedule equations.

$$\begin{aligned}
 \text{People Required} &= \text{Effort} / \text{Duration} \\
 &= 8.2706 / 4.9153 \\
 &= 2 \text{ approx.}
 \end{aligned}$$

## 6- Critical Path Method Analysis (CPM Analysis)

CPM models the activities and events of a project as a network. Activities are depicted as nodes on the network and events that signify the beginning or ending of activities are depicted as arcs or lines between the nodes. The following is an example of a CPM network diagram:

Steps in CPM Project Planning

1. Specify the individual activities.
2. Determine the sequence of those activities.
3. Draw a network diagram.
4. Estimate the completion time for each activity.
5. Identify the critical path (longest path through the network)
6. Update the CPM diagram as the project progresses.

### 6.1 Specify the Individual Activities

PUCIT-Project Coordination Office	Version: 1.0
Final Project Proposal	Date: 12 June, 2015

From the work breakdown structure, a listing can be made of all the activities in the project. This listing can be used as the basis for adding sequence and duration information in later steps.

## 6.2 Determine the Sequence of the Activities

Our GUI is dependent on both Modules of Our Project. GUI is the Activity that always dependent on other Modules.

## 6.3 Draw the Network Diagram

Once the activities and their sequencing have been defined, the CPM diagram can be drawn. CPM originally was developed as an activity on node (AON) network, but some project planners prefer to specify the activities on the arcs.

## 6.4 Estimate Activity Completion Time

The time required to complete each activity is estimated using past experience or the estimates of knowledgeable persons. CPM is a deterministic model that does not take into account variation in the completion time, so only one number is used for an activity's time estimate.

## 6.5 Identify the Critical Path

The critical path is the longest-duration path through the network. The significance of the critical path is that the activities that lie on it cannot be delayed without delaying the project. Because of its impact on the entire project, critical path analysis is an important aspect of project planning.

Determining the following six parameters for each activity which can identify the critical path:

**ES:** earliest start time: the earliest time at which the activity can start given that its precedent activities must be completed first.

$ES(K) = \max [EF(J) : J \text{ is an immediate predecessor of } K]$

**EF:** earliest finish time: equal to the earliest start time for the activity plus the time required to complete the activity.

$EF(K) = ES(K) + Dur(K)$

**LF:** latest finish time: the latest time at which the activity can be completed without delaying the project.

$LF(K) = \min [LS(J) : J \text{ is a successor of } K]$

**LS:** latest start time: equal to the latest finish time minus the time required to complete the activity.

$LS(K) = LF(K) - Dur(K)$

**TS:** Total Slack: the time that the completion of an activity can be delayed without delaying the end of the project

$$TS(K) = LS(K) - ES(K)$$

**FS:** Free Slack: the time that an activity can be delayed without delaying both the start of any succeeding activity and the end of the project.

$$FS(K) = \min [ES(J) : J \text{ is successor of } K] - EF(K)$$

The slack time for an activity is the time between its earliest and latest start time, or between its earliest and latest finish time. Slack is the amount of time that an activity can be delayed past its earliest start or earliest finish without delaying the project.

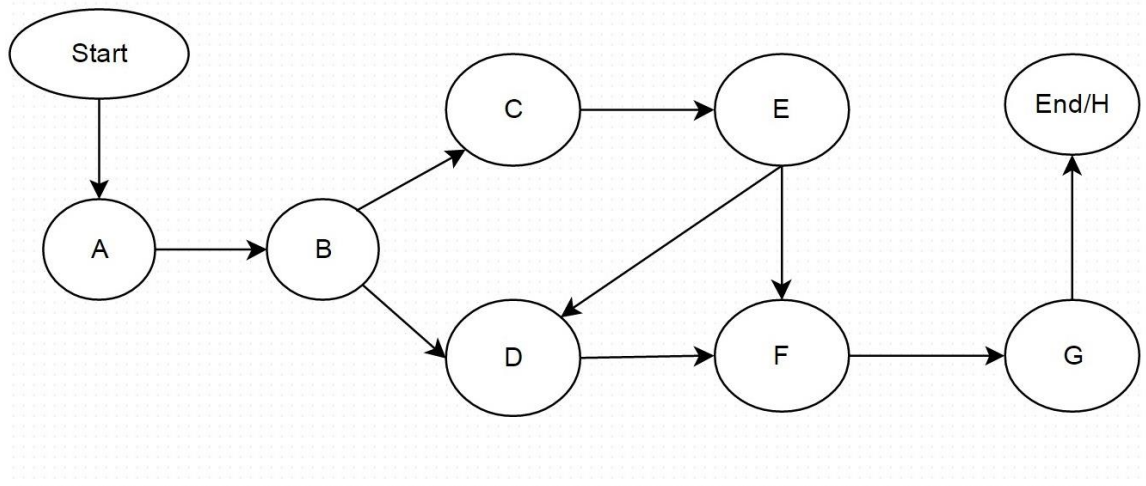
The critical path is the path through the project network in which none of the activities have slack, that is, the path for which  $ES=LS$  and  $EF=LF$  for all activities in the path. A delay in the critical path delays the project. Similarly, to accelerate the project it is necessary to reduce the total time required for the activities in the critical path.

### 6.6 Update CPM Diagram

As the project progresses, the actual task completion times will be known and the network diagram can be updated to include this information. A new critical path may emerge, and structural changes may be made in the network if project requirements change.

**CP for This Project Tabulated Bellow:**

Activity	Immediate Predecessor	Duration (Weeks)
A	None	1
B	A	2
C	B	3
D	B	4
E	C	1
F	D, E	3
G	F	1
H	G	1



• **Network Diagram for the above mentioned Activities**

A	Deliverable 1(Paper Work)
B	Deliverable 2(Paper Work)
C	GUI Designing(Designing Work)
D	Module 1: Spy Apps Detection Sys (Development)
E	GUI Implementation(Development)
F	Module 2: Malware/Fake Apps Detection Sys (Development)
G	Module Testing as Developer(Lab Testing)
H	Project Testing After Deployment on Device (Open Testing)

Activity	Duration	ES	EF	LS	LF	TS	FS
A	1	0	1	1	1	1	1
B	2	0	1	2	2	3	1
C	3	1	2	3	3	2	2
D	4	1	3	4	4	3	2
E	1	0	1	2	4	3	1
F	3	2	2	4	3	2	1
G	1	1	1	5	3	1	1
H	1	1	1	3	2	1	1

The parameters and slacks are calculated as follows:

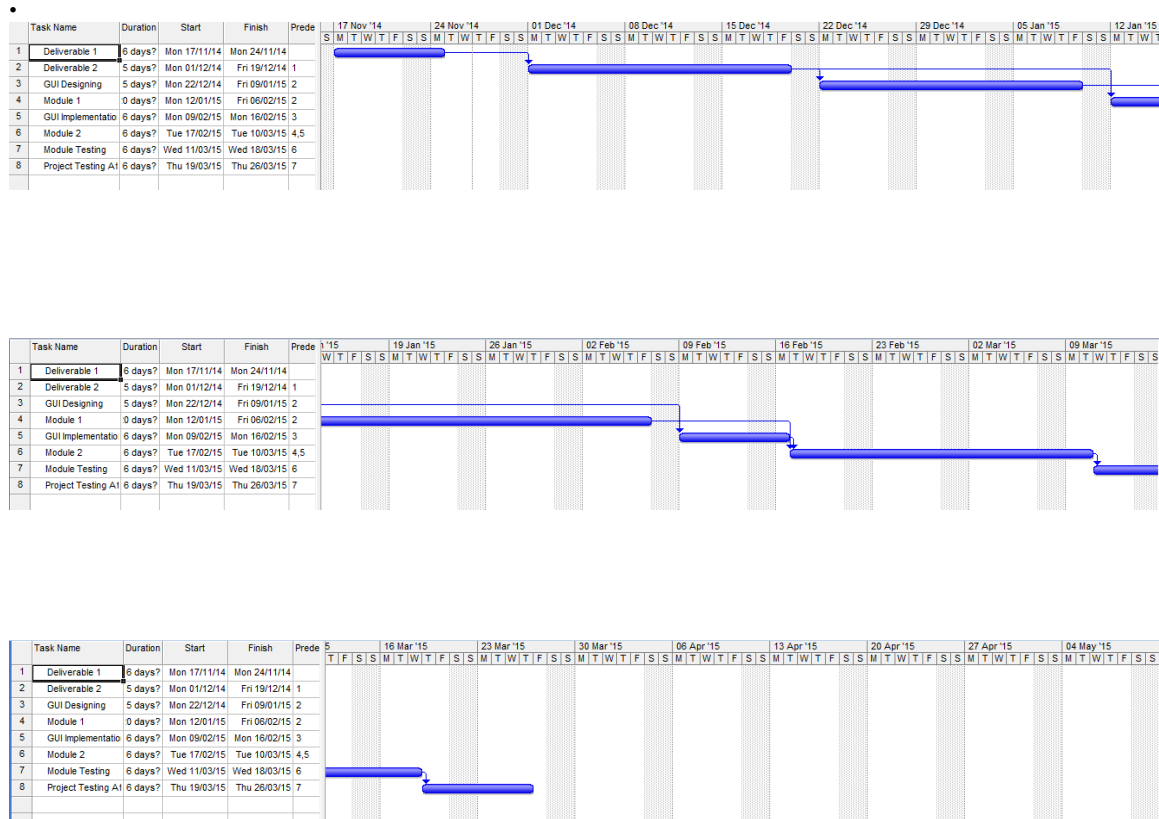
**Here are 2 critical path** A->B-> C->E-> F-> G->H = 12 A->B-> D-> F-> G->H = 12



## 7- Gantt Chart

Timeline of project is shown below in graphical representation with gantt chart.

### Gantt Chart Diagrams



## 8- Introduction to team members

We are three Team Members in total. Project is not divided into parts for each member but used different techniques. Project is consist of

- Paper Work
- Development

Paper work is individually done by each member.

### 1. Zaheer Ahmed (Group Leader):

#### Contribution in Project

Whole Paper work Editor + Work on Project related Details like Scope, Working and time required to complete etc. Idea for Project and concern with authorities are also included in Zaheer's Work.

#### Skill Set(Directly/Conceptually Used in Project)

PUCIT-Project Coordination Office	Version: 1.0
Final Project Proposal	Date: 12 June, 2015

- C/C++ Programming
- Work on HTML, CSS
- Work on Graphic, Web Designing
- Computer Security
- Cryptography
- Work on *SQL Injections*
- Database Operations
- Work on *Penetration Testing/System's Security Flaw Finding*
- Work on *Win Based Operating System Security Hacking*
- Work on Web based Application Security
- Batch Programming/CL Operations
- Mobile Computing(XML)

## 2. Talha Sajjad

### Contribution in Project:

In paper work Talha contributed to make system Diagrams and Finishing Editing. Talha also done with all calculations in *Cost Estimation* and time Calculation.

### Skill Set

- C/C++ Programming
- Internet Programming
- Database Management System
- Operating System
- Mobile Computing

## 3. Zeeshan Tahir

### Contribution in Project:

In paper work Zeeshan contributed for Editing and other working for submission. Zeeshan also done with Project's Idea and Android Related Details.

### Skill Set

- C/C++ Programming
- Internet Programming
- Database Administration
- C# (studying)
- Mobile Computing (studying)

## 9- Tools and Technology with reasoning

Following tools and technology this project required:

- Eclipse as host platform with Android SDK(for Development).
- Android (All versions till now).
- The Java programming language for Development.
- This is a medium scale Security App which can be developed by some Computer Science buddies within 3 to 4 months.
- Time to develop this project shown in Gantt also.
- Existing tools like MS project to visualize project before its development.

PUCIT-Project Coordination Office	Version: 1.0
Final Project Proposal	Date: 12 June, 2015

- This tool will support all Android Versions Released till now October 2014.

Project Need Developer's interest in **Security Application Development** to be completed. First we need to know how Android Security can be **Violated/Hacked** then we can protect it.

## 10- Vision Document

The main Vision/Theme of this project to Secure Android End User. If anyone stealing user's data then Spyware Technology will detect them. If someone used fake certificate to run his malicious code on user's device then malware detector technology will deal with them.

The problem is Security and Data Stealing and we are totally Dealing with that. This Project will also deal with those Apps they maybe Detecting GPS location of user Unnecessarily.

## 11- Risk List

According to risk list we are trying to find some **Flaws** in Android Operating System which may allow us to install or uninstall an App (which is *Detected Malicious/Harmful or Security Violator*) without user permissions with help of our App.

## 12- Requirements Engineering

### Systems Specifications

The following are the clauses that must be included while describing the system specifications.

### Introduction

#### I-Project Title

Vulnerable Apps Detector App

#### II-Project Overview Statement

This Project is for Android End user. User of this App will be facilitated with following features.

- 1: User can find malicious app's on his Android system.
- 2: User can see Apps with those permissions they are vulnerable for a user security.

#### III-Existing System

Goals of this project are to secure Android Devices. With this App user will be able to scan his Device for some malicious App or if an App is using extra permissions which are irrelevant to App's work.

*Objectives are following:*

We will perform Checks for those Apps which are using other App's Certificate as a fake certificate.

Secondly this App will scan system for those Apps which using some extra permissions.

#### IV-Scope of the System

This Project is for Android End user. User of this App will be facilitated with following features.

- 1: User can find malicious app's on his Android system.

2: User can see Apps with those permissions they are vulnerable for a user security.

Goals of this project are to secure Android Devices. With this App user will be able to scan his Device for some malicious App or if an App is using extra permissions which are irrelevant to App's work.

Objectives are following:

- We will perform Checks for those Apps which are using other App's Certificate as a fake certificate.
- Secondly this App will scan system for those Apps which using some extra permissions.

### **Summary of Requirements:(Initial Requirements)**

The purposed system must fulfill following requirements as follow:

This App must be proper installed to work with it. To get secured from malicious Apps or Fake Apps that maybe installed with a Fake Certificate or Spy Apps user must complete scanning process. At the end of Scanning Process this system will list down all Apps which are violating predefined Security Protocols.

## **Identifying External Entities or Actors**

### **Perform Refinement**

We found the following entities more related to our application;

- User
- System

### **12.1 Capture "shall" Statements and the external entities (Actors)**

Para #	External Entity	Initial Requirements
1.0	User	A User "shall" Run System
1.0	User	A User "shall" Select option for Scan type.
1.0	System	The system "shall" display option for scan how Intense Scan user Select
1.0	User	A User "shall" select scan type tor run Scan on his/her Device
1.0	System	A System "shall" Display his Scan Progress to User
2.0	System	System "shall" Listdown Result for Scan to User so User can uninstall Those Apps following Scan Results.

### **12.2 Allocate Requirements**

Para #	Initial Requirements	Use Case Name
1.0	A user "will" select the type to scan	UC_Scan_Type

1.0	A user “shall” select the mode to be scanned	UC_Scan_Intensity
1.0	A user “shall” view his details for verification purposes	UC_Progress_View
1.0	System “will” listdown the results	UC_Scan_Results

### 12.3 Priorities Requirements

Para #	Rank	Initial Requirements	Use Case ID	Use Case Name
1.0	Highest	A user “will” select the type to scan	UC_1	UC_Scan_Type
1.0	Highest	A user “shall” select the mode to be scanned	UC_2	UC_Scan_Intensity
2.0	Lowest	A user “shall” view his details for verification purposes	UC_3	UC_Progress_View
2.0	Highest	System “will” listdown the results	UC_4	UC_Scan_Results

### 12.4 Requirements Traceability Matrix

Sr#	Para #	System Specification Text	Build	Use Case Name	Category
1	1.0	A user “will” select the type to scan	B1	UC_Scan_Type	Common User Security
2	1.0	A user “shall” select the mode to be scanned	B1	UC_Scan_Intensity	Common User Security
3	1.0	A user “shall” view his details for verification purposes	B1	UC_Progress_View	Common User Security
4	1.0	System “will” listdown the	B1	UC_Scan_Results	Common User

		results			Security
--	--	---------	--	--	----------

### 13- Use Case Description

While technically not part of UML, use case documents are closely related to UML use cases. A use case document is text that captures the detailed functionality of a use case. Description of all use case's are written down. Use case description typically contains the following parts:

#### *UC1: Menu*

<b>Use Case Name</b>	Menu
<b>Scope</b>	Android Based Security System
<b>Level</b>	Initial
<b>Primary Actor</b>	User
<b>Secondary Actor</b>	
<b>Goal</b>	To let User choose type of scan
<b>Pre-condition</b>	User must have Licensed copy of Application.
<b>Post condition</b>	New request will be given to user
<b>Trigger</b>	User clicks Scan option

#### **First Scenario:**

1. User shown an option in the menu field.	1. System shows user the select option which user can select to scan.
2. if User selected 'Spy Apps Scan'	2. System let user select scan type.
3. User clicks on 'OK'.	3. System opens another activity that prompts option of "Select Scan Intensity".
4. User clicks on 'Strict' or 'Normal'.	4. System get selected an option from user.

#### **Alternate Path:**

2.a. User selected 'Vulnerable Apps Scan'	2.a.. System let user select scan type.
3.a. User clicks on 'OK'.	3.a. System opens another activity that prompts option of run scan.

#### *UC2: Vulnerable Apps Detector*

<b>Use Case Name</b>	Vulnerable Apps Detector
<b>Scope</b>	Android System Security Flaw Scanning Module.
<b>Level</b>	Intermediate
<b>Primary Actor</b>	User
<b>Secondary Actor</b>	
<b>Goal</b>	To let User scan his/her system for Android

PUCIT-Project Coordination Office	Version: 1.0
Final Project Proposal	Date: 12 June, 2015

	Vulnerabilities.
<b>Pre-condition</b>	User must have Selected a Scan type in Menu..
<b>Post condition</b>	Show result system will show scanning Progress and results to user.

**First Scenario:**

1. User clicks on 'Run Scan'.	1. System runs scanning process for Vulnerable Apps in System.
2. User can see Progress of scan on Screen but can also do Multitasking if clicks 'Minimize'.	2. System let user use other Applications till Scan Process Completes.

**Alternate Path:**

2.a. User clicks 'Cancel'.	2.a.. System close shows a pop-up window showing 'Are you Sure'.
3.a. User clicks on 'yes'.	3.a. System close Application.
4.a. User Clicks on 'No'.	4.a. System let user to previous state.

***UC3: Spy Apps Detector***

<b>Use Case Name</b>	Spy Apps Detector
<b>Scope</b>	Android System Security Flaw Scanning Module.
<b>Level</b>	Intermediate
<b>Primary Actor</b>	User
<b>Secondary Actor</b>	
<b>Goal</b>	To let User scan his/her system for Android System for those Apps that Spying on him..
<b>Pre-condition</b>	User must have Selected a Scan Intensity in Menu..
<b>Post condition</b>	Show result system will show scanning Progress and results to user.

**First Scenario:**

1. User clicks on 'Run Scan'.	1. System runs scanning process for Spy Apps in System.
2. User can see Progress of scan on Screen but can also do Multitasking if clicks 'Minimize'.	2. System let user use other Applications till Scan Process Completes.

**Alternate Path:**

2.a. User clicks 'Cancel'.	2.a.. System close shows a pop-up window showing 'Are you Sure'.
3.a. User clicks on 'yes'.	3.a. System close Application.

4.a. User Clicks on 'No'.	4.a. System let user to previous state.
---------------------------	---

#### ***UC4: Results***

Use Case Name	Results
<b>Scope</b>	Shows Android System's installed Apps that infects user Security (Data).
<b>Level</b>	Intermediate
<b>Primary Actor</b>	User
<b>Secondary Actor</b>	
<b>Goal</b>	To let User View his/her system for Android Vulnerabilities.
<b>Pre-condition</b>	User must have Completed a Scan Process.
<b>Post condition</b>	Show result system will also show Detail results.

#### **First Scenario:**

1. User clicks on running App's Icon.	1. System shows results of scanning to user.
2. User clicks on 'Detail' against each App.	2. System let user know about all aspects which cause App to be shortlisted.

#### **Alternate Path:**

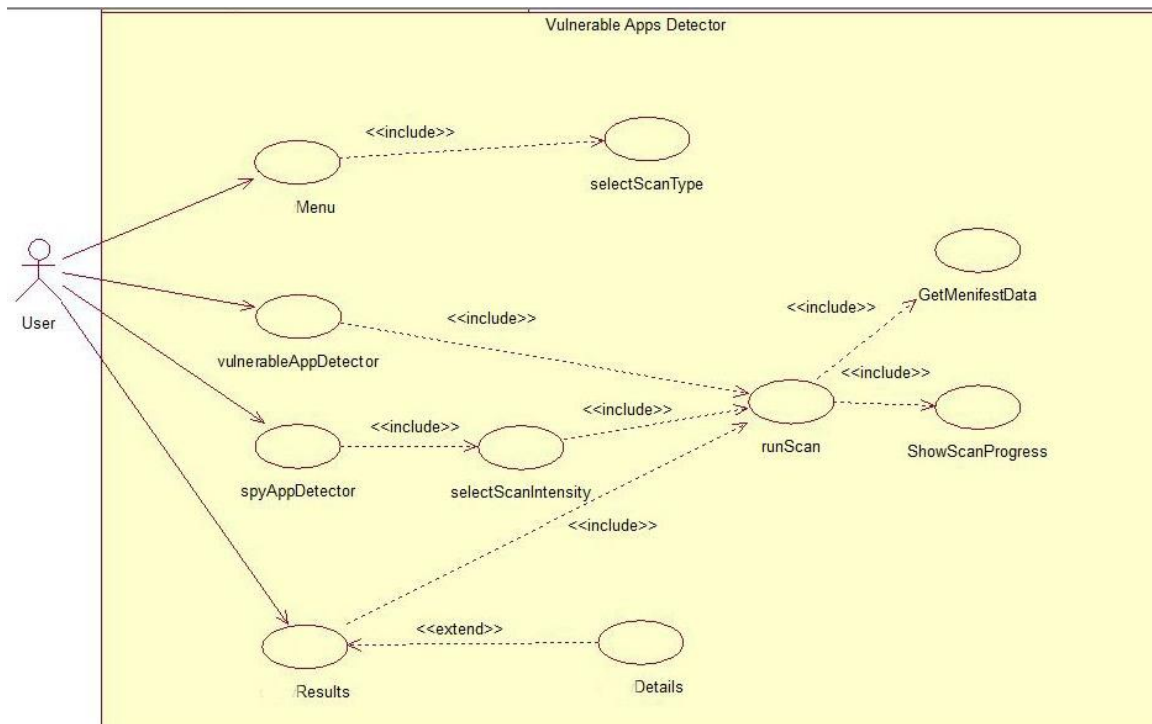
2.a. User clicks 'Exit'.	2.a.. System close all processes and end up App.
--------------------------	--

## **14- Use Case Diagram (Analysis level)**

Analysis level usecase diagram is a refined High level use case diagram and is actually the explanation of high level usecase diagram. In this diagram high level usecases are expanded in a way that exhibit how high level usecases will reach to their functionality. Two types of relationships are used in this diagram. Which are:

- Extend
- Include



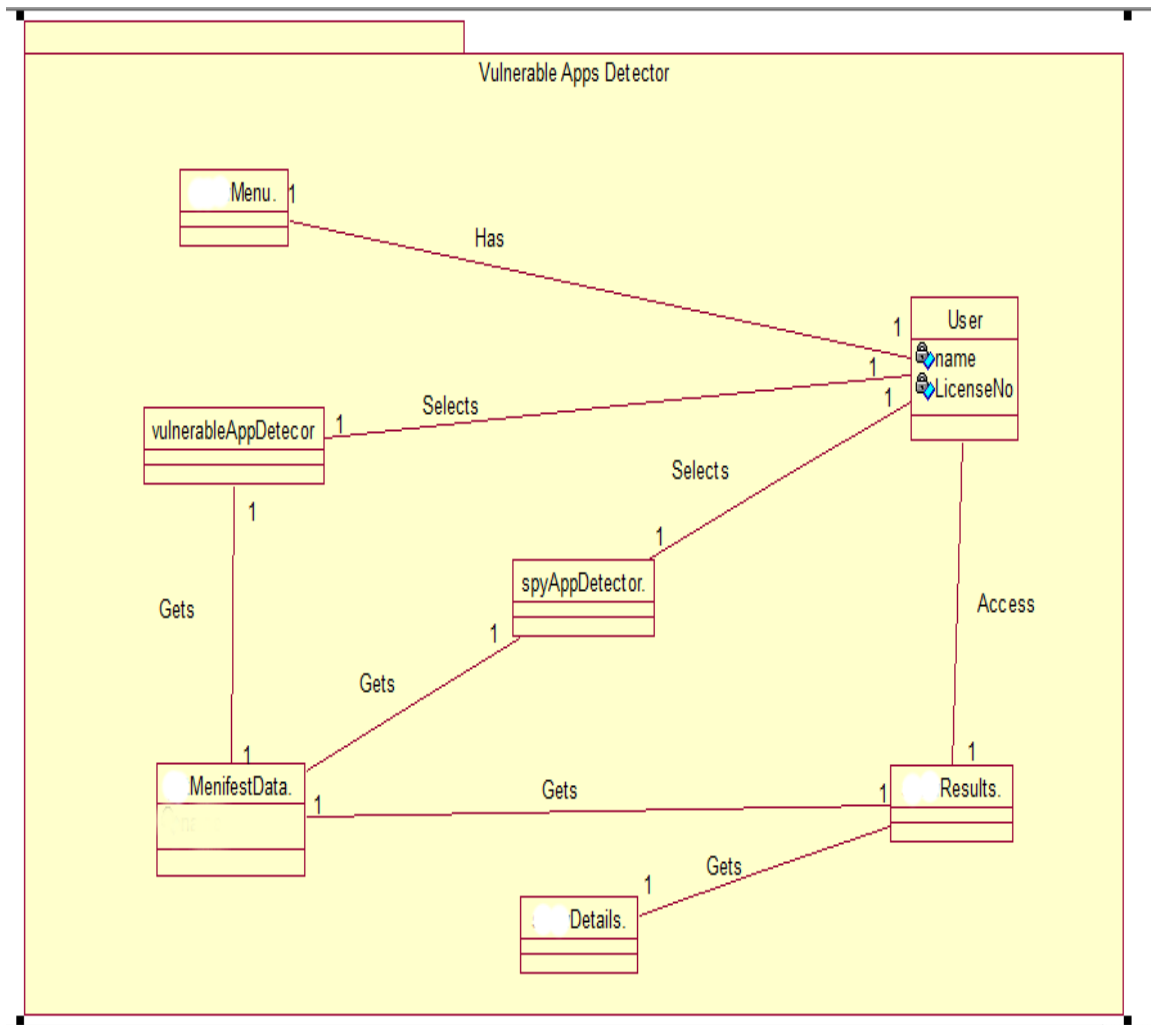


## 15- Domain Model

Domain models represent the set of requirements that are common to systems within a product line. There may be many domains, or areas of expertise, represented in a single product line and a single domain may span multiple product lines. The requirements represented in a domain model include:

- Definition of scope for the domain
- Information or objects
- Features or use cases, including factors that lead to variation
- Operational/behavioral characteristics

A product line definition will describe the domains necessary to build systems in the product line.



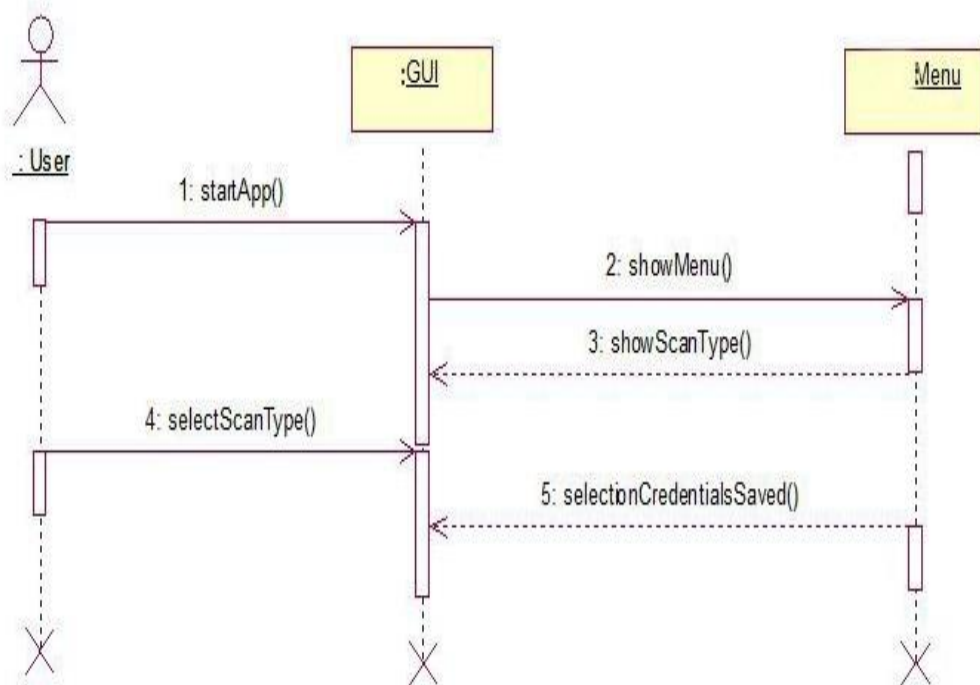
## 16- Sequence Diagrams

A Sequence diagram depicts the sequence of actions that occur in a system. The invocation of methods in each object, and the order in which the invocation occurs is captured in a Sequence diagram. This makes the Sequence diagram a very useful tool to easily represent the dynamic behavior of a system.

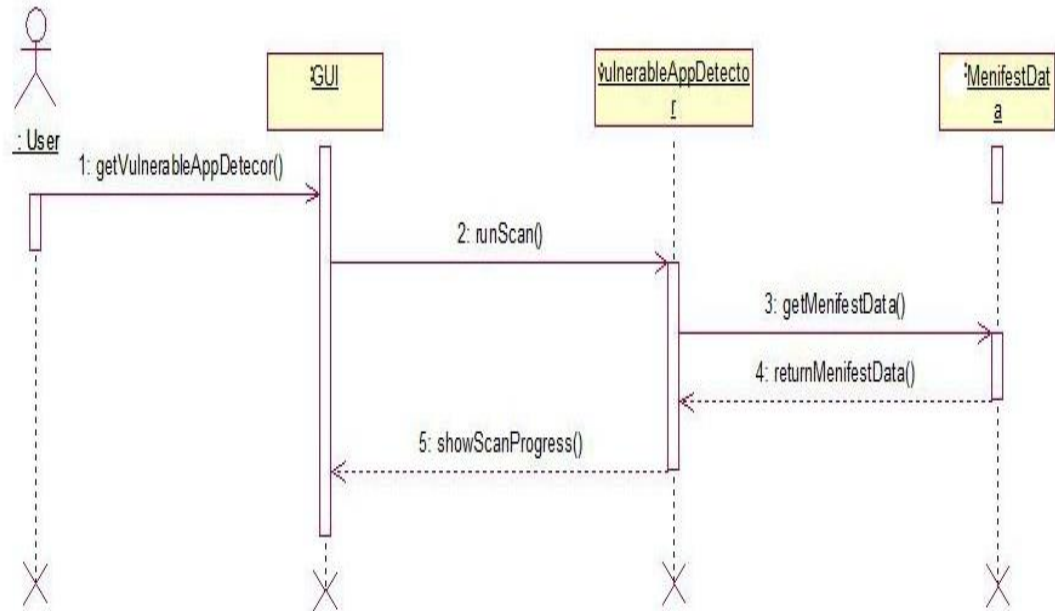
A Sequence diagram is two-dimensional in nature. On the horizontal axis, it shows the life of the object that it represents, while on the vertical axis, it shows the sequence of the creation or invocation of these objects.

Because it uses class name and object name references, the Sequence diagram is very useful in elaborating and detailing the dynamic design and the sequence and origin of invocation of objects. Hence, the Sequence diagram is one of the most widely used dynamic diagrams in UML.

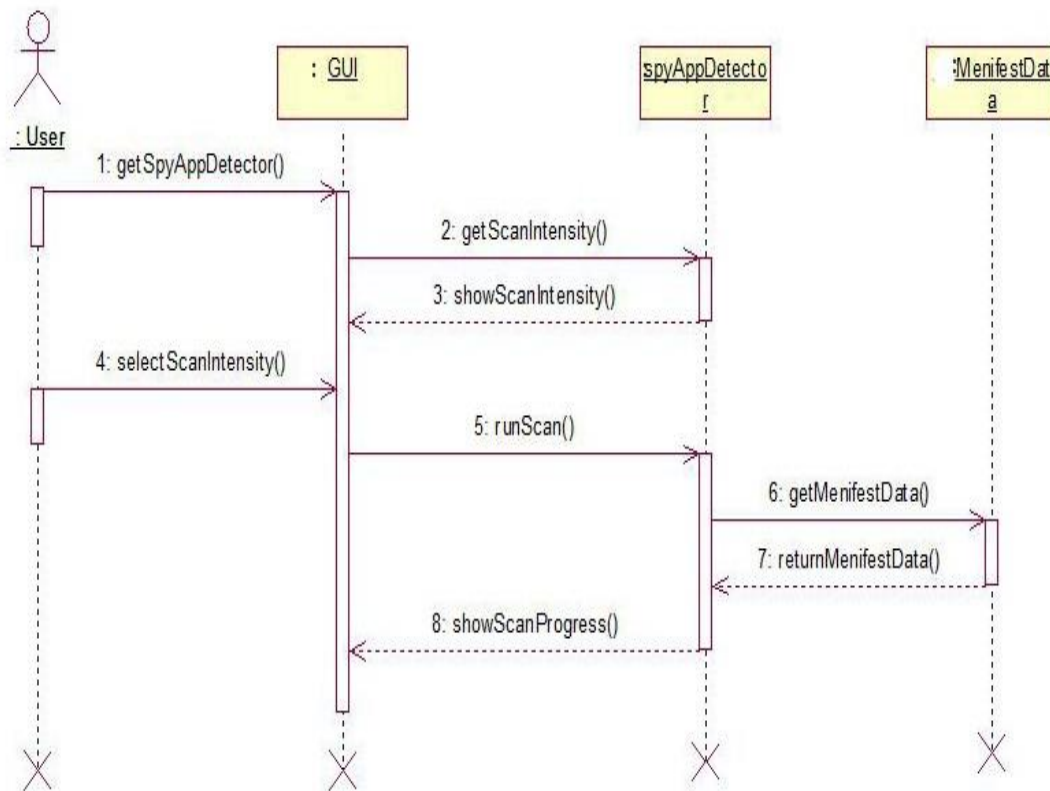
### UC1:Menu



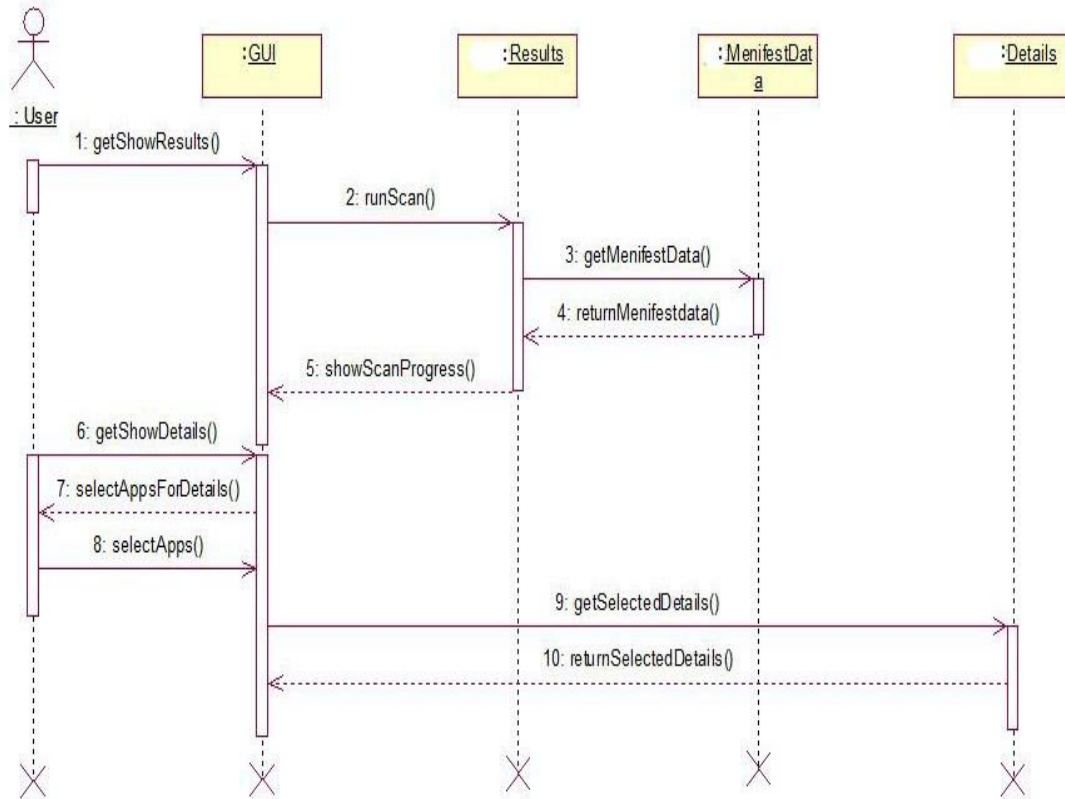
## UC2:VulnerableAppDete



## UC3: SpyAppDetector

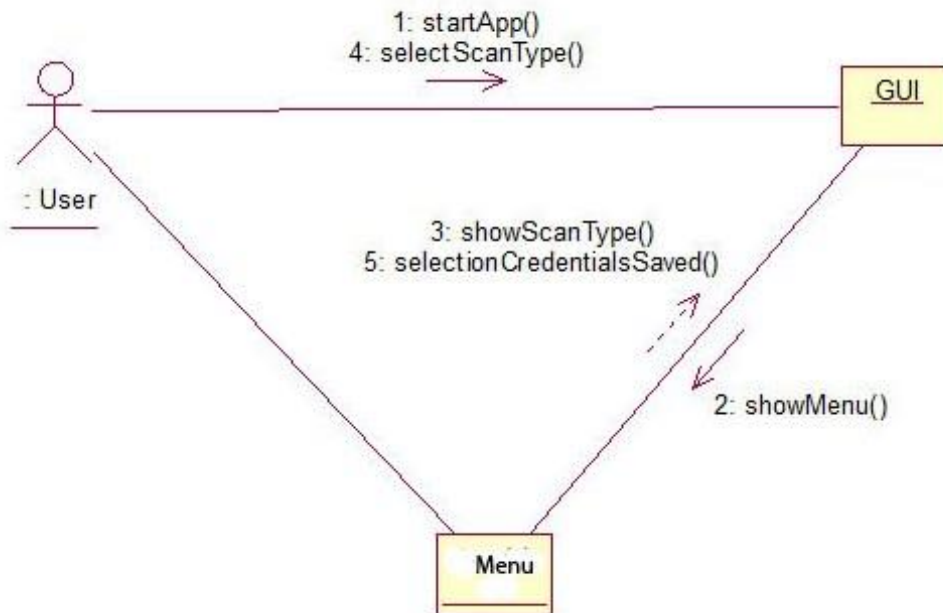


## UC4: Results



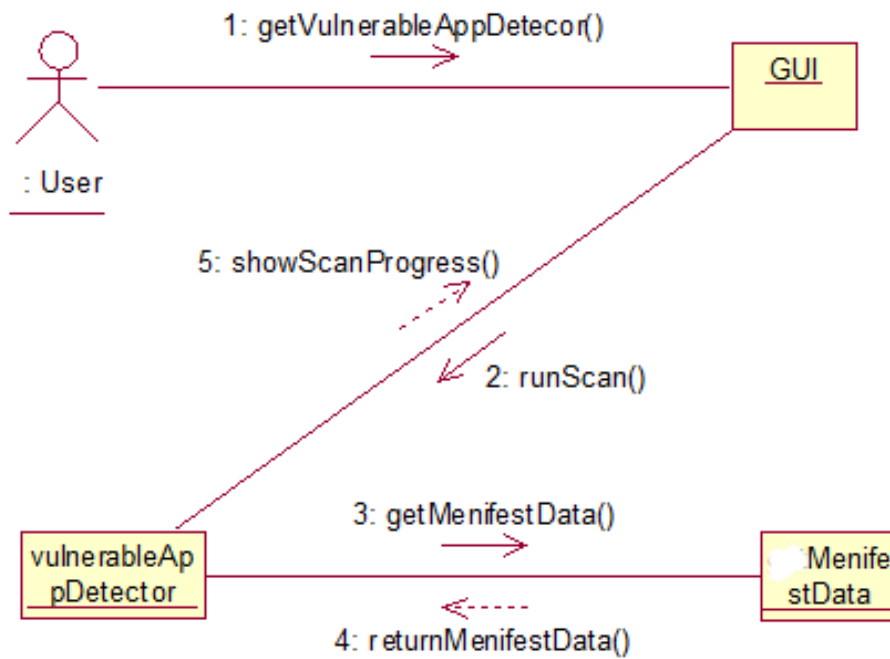
## 17- Collaboration Diagrams

### UC1:Menu



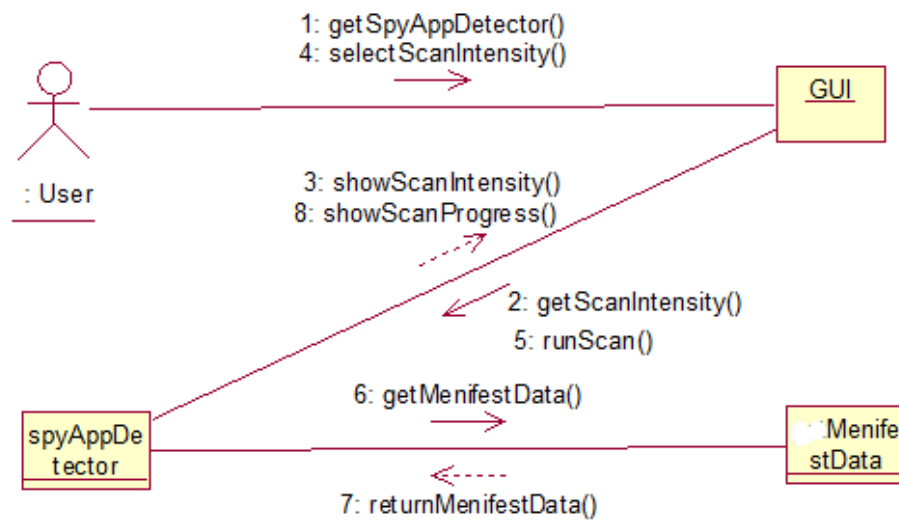
## UC2: VulnerableAppDetector

---

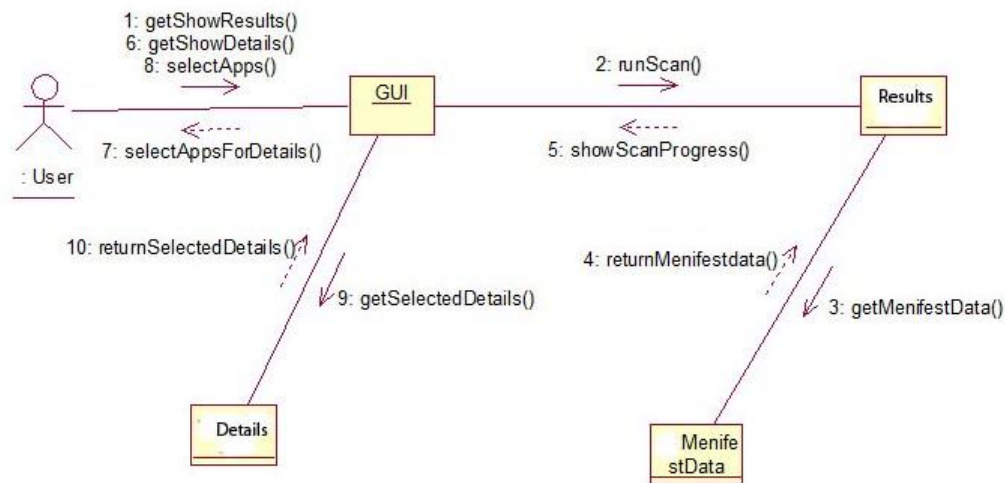




## UC3: SpyAppDetector



## UC4: Results



## 18- Operation Contracts

A UML Operation contract identifies system state changes when an operation happens. Effectively, it will define what each system operation does. An operation is taken from a system sequence diagram. It is a single event from that diagram. A domain model can be used to help generate an operation contract.

### Operation Contract Syntax

1)

#### Name:

showMenu (licenseNo)

#### Responsibility:

PUCIT-Project Coordination Office	Version: 1.0
Final Project Proposal	Date: 12 June, 2015

The responsibility of this operation is show all possible operations.

**Cross Reference:**

UC Show Menu

**Exception:**

None

**Pre-Condition:**

User must have a Licensed copy of Application.

**Post Condition:**

- A Selected Scan Type Begins.

2)

**Name:**

vulnerableAppsDetector ()

**Responsibility:**

The responsibility of this operation is to scan all Apps for fake Manifest Certificate.

**Cross Reference:**

UC Vulnerable Apps Detector

**Exception:**

None

**Pre-Condition:**

A Scan type has been selected by a user.

**Post Condition:**

- A resultant list of Vulnerable Apps was created.

3)

**Name:**

spyAppsDetector ()

**Responsibility:**

The responsibility of this operation is to scan all Apps for Spy Operations with Resettled Intensity Criteria.

PUCIT-Project Coordination Office	Version: 1.0
Final Project Proposal	Date: 12 June, 2015

**Cross Reference:**

UC Spy Apps Detector

**Exception:**

None

**Pre-Condition:**

A Scan Intensity has been selected by a user.

**Post Condition:**

- A resultant list of Spy Apps was created.

4)

**Name:**

showResult ()

**Responsibility:**

The responsibility of this operation is to show scan results to user with an extra Detail option.

**Cross Reference:**

UC Show Result

**Exception:**

None

**Pre-Condition:**

A Successful Scan has been conducted by a user.

**Post Condition:**

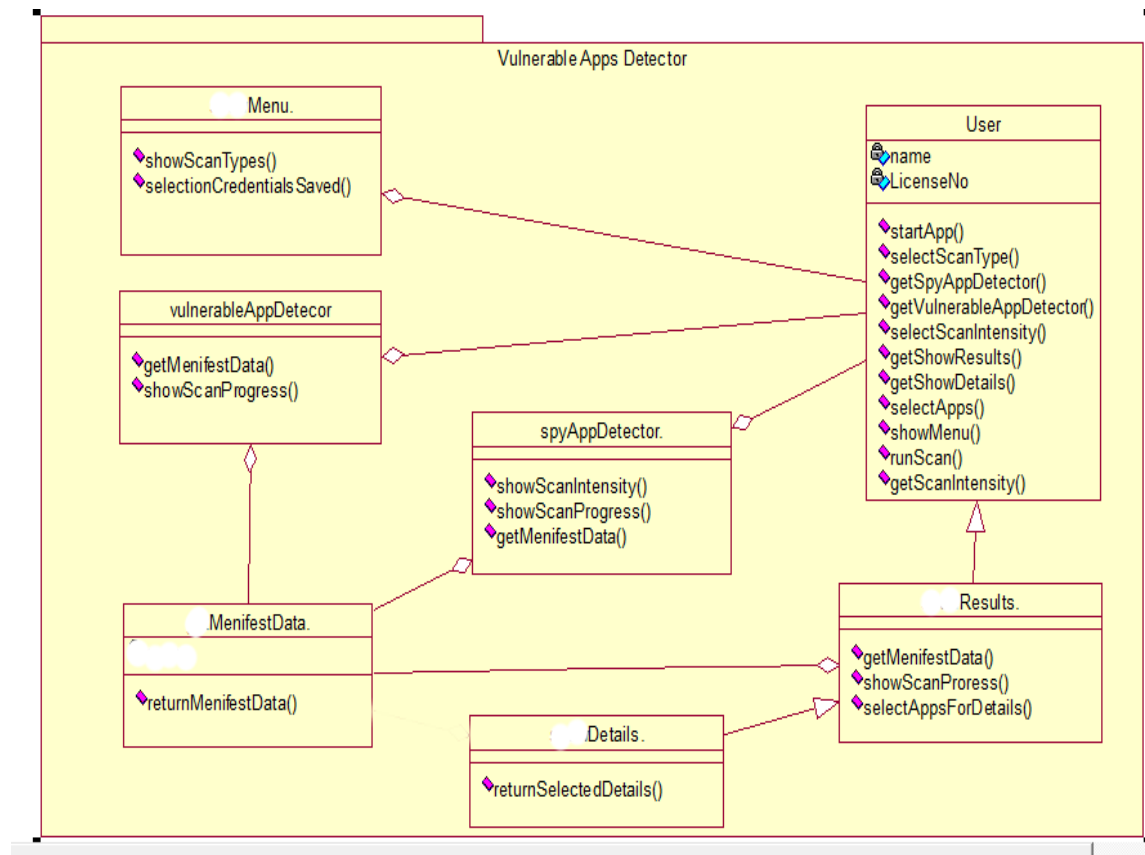
None

## 19- Design Class Diagram

Classes are the work-horses of the design effort—they actually perform the real work of the system. The other design elements—subsystems, packages and collaborations simply describe how classes are grouped or how they interoperate.

Capsules are also stereotyped classes, used to represent concurrent threads of execution in real-time systems. In such cases, other design classes are 'passive' classes, used within the execution context provided by the 'active' capsules. When the software architect and designer choose not to use a design approach based on capsules, it is still possible to model concurrent behavior using 'active' classes.

Active classes are design classes, which coordinate and drive the behavior of the passive classes - an active class is a class whose instances are active objects, owning their own thread of control.



## 20- Data Model

The data model is a subset of the implementation model, which describes the logical and physical representation of persistent data in the system.

