Table of Content

VenturePact

1. COMPANY PROFILE

❖ Name of the Organization

Overview:

VenturePact is an online marketplace that connects businesses with pre-screened software development firms. The marketplace provides a transparent platform where companies can get customized pitches, see a firm's ratings, inspect a verified portfolio, get their standard FAQs answered and make secure payments. It is founded by Mr Pratham Mittal and Randy Rayess.

They have pre-screened hundreds of software development firms in different verticals and provide extensive data around each including client reviews & ratings, validated portfolios and employee profiles. This transparency lets businesses make a better & faster decision regarding which firm to go with for their high risk projects.

Clients:

ESPN, Flip Kart, Adidas, Yale University, BMW, Ashoka

Offices:

Its offices are located in New York City & Philadelphia.

Services:

Mobile applications: Technologies served: IOS, Android, Symbian, Blackberry (6 more)

Web applications: Technologies served: Ruby, php, Python, Java (32 more)

Business applications: Platforms served: SAP, Oracle (16 more)

Wearable& Internet of things Technologies: Google Glass, Galaxy Watch, iBeacon

UI/UX Design categories: SaaS UI/UX, Mobile UI/UX, Mobile & Web Ad design

Big data & data visualization: Technologies served: Hadoop, MapReduce, and R (11 more)

❖ Products of VenturePact

a) MidWife – A Healthcare App

MidWife is a healthcare app, helping maternity in remote locations. App helps ladies who are in remote locations and gives information on their medical history, treatment and follow-ups. App also provides items available with a shopping cart.

b) RetailCo – An Ecommerce App

RetailCo is an iPhone app, which provides details of products in store. The app gives easy access to stores having products. The app gives the user the nearest store location withDistance. App also tells whether offers are available with the store or not.

c) MatchMaker

MatchMaker is a website providing social networking to people.

d) **AirCare** Founder : George Xiangwen Zend

Aircare is a medical app where doctor and patient can communicate directly using video chat.

e) **Boupp** Founder : Parisa

Boupp is a website providing fashion products, Multiple franchise companies can put their products. Companies can get advanced data results for the products.

f) **Accompliss**Founder : Justin

Accompliss is a hotel management app, App allows customers to call room service, food, spa, room booking etc.

g) **FireFly** Founder : Aziz

FireFly is browser sharing website, where two users can share there browser in real-time and can view the live actions.

h) **SmartyPal** Founder : Prasanna

SmartyPal is a kids app where kids can read stories and app gives animation affects for the characters involved.

2. Profile of Problem

Existing System

Current forensic expert systems are based on manual gathering of evidences that are found at the crime location and storing the details in the papers. These systems have drawback of manipulation of the evidences before they are available to the forensic experts.Expert systems are computer programs that contain subject-specific knowledge. Expert systems can analyse a given set of data with the same skill as a human expert. In the forensic community, an expert system can assist forensic analysts with the routine processing of single source samples and can streamline sample processing workflow by alerting the analyst to those samples that require more thorough manual review as specified in the user defined rules.

Disadvantage

- Unauthorised modification of the Evidences
- Waste of time in collecting and storing the evidences manually
- Difficulty in searching cases

Introduction to "Forensic Expert System"

Our proposed system will have interface available to collect and capture all the details of the evidences and store them at single place called server dynamically. These evidences can be viewed and used by the forensic experts to solve the cases. This system helps to resolve the issue of manipulation of collected evidences by automatically uploading all the collected evidences that can be accessed by only authorized person only like the forensic experts head.

The goal of this forensics app is to perform a structured investigation while maintaining a documented chain of evidence to find out exactly what happened to different cases and who was responsible for it.

Advantage

- Ease and quick access to the evidences
- Generation of reports of each cases and their evidences
- Different module/form to capture each type of evidence
- Synchronization of collected evidences to save on server
- Easy maintenance of records compare to manual system

3. Project Analysis

□ □PRODUCT DEFINITION

“Forensic Expert System” is an android App which helps the forensic experts and investigatorsto easily manage different type of cases. It provides user friendly interface with separate user account facility and modification rights. The app also facilitates users to easily add different type of evidence for each case. In this project we use Java and for designing purpose XML language and SQLite as database. So it’s a fully functional application having nice user interface and best functionality.

❖ FEASIBILITY ANALYSIS

The feasibility study is an evaluation and analysis of the potential of the proposed project which is based on extensive investigation and research to support the process of decision making. Various issues of feasibility analysis considered in project are as follows:-

➢ Technical Feasibility Study

Technically we analyses that it is possible to develop such system with Java Because Java is pure object oriented language which provides enough functions to implement various functionalities. Various design issues of app can be resolved by using XML and front-end validations can be implement with the help of java. MYSQL Can is used as database which is compatible with PHP for implementing web services at server side.

➢ Economic Feasibility Study

This app is also economically feasible because it can easily be deployed on Google play.Google offers the Google Play service in which programmers can offer their Android application toAndroid users. Google phones include the Google Play application which allows installing applications.Google Play also offers an update service, e.g. if a programmer uploads a new version of hisapplication to Google Play, this service will notify existing users that an update is available and allow toinstall it.

➢ Legal Feasibility Study

Since the project needs no copyright, patenting, and doesn’t intent to have any relation with anybody else’s intellectual property rights, it can be considered as a legally feasible project

➢ Time Feasibility Study

As it has been more probable (as per the requirements, functions, and performance specifications of the system) that the project can be completed within the given time frame, it is considered that the undertaking this project is feasible in the context of time.

❖ PROJECT PLAN

Project planning is performed basically in the large organization where it requires effective management to control and to find the desired result. Project planning for an organization has following steps:

- Acquiring and organizing the tools and resources for the project.
- Preparation of well-defined schedule for events of the project.
- Proper evaluation of progress of project development.
- Establishing various standards for the project by which we can find the standard output.

We had the time from August 2014 to November 2014 for the project preparation thus within this duration we had to estimate everything to prepare this project. According to the first step we acquired the resources and organize them in well-defined manner so that there we should not face any problem. The second thing is distributing the time stamp into small time unit according to the module of the project. This is the term as schedule the task according to the event happened in the project. We distributed the time into different module of the whole part of the project development.

Evaluating the progress of project development is to measure the functionality of the project by comparing with the requirement criteria. It is told that we cannot do anything if we have not properly measure the task which you are going to do. This is a project which has everything in it so the proper plan is much necessary. Initially, the project must establish the objectives of each phase of development. Each phase must be of a controllable size, and every task within the project must be performed with responsibility. We first established the objectives of each phase and at last we guessed that the objectives that have been established are met in the proper time.

4. SOFTWARE REQUIRMENT ANALYSIS & TOOLS

Introduction:

One of the most difficult tasks is that the selection of the software, once system requirement is known is determining whether a particular software package fits the requirements. After initial selection further security is needed to determine the desirability of particular software compared with other candidates. This section first summarizes the application requirement question and then suggests more detailed comparisons:

GENERAL DESCRIPTION

❖ Android

Android is a modern, open source operating system and SDK for mobile devicesdeveloped by Google. With it you can create powerful mobile applications. This becomes even more attractive when your applications can access Web services, which means you; need to speak the language of the Web: XML. In this article, you will see different options for working with XML on Android and how to use them to build your own Android applications.

-The Linux kernel 2.6-which includes useful drivers that allow for example Wi-Fi or Bluetooth.

-The library written in C and C + + that provide higher level functionality such as an HTML engine, or adatabase (SQLite).

-A runtime environment for applications based on a virtual machine, made for inefficient machines such astelephones. The aim is to translate JAVA in machine language understood by Android.

-A JAVA framework that allows applications running on the virtual machine to organize and cooperate.

Why Android is better..?

This is a list of features in the Android operating system.

➢ Applications

Android includes most of the time many Google applications like Gmail, YouTube or Maps.These applications are delivered with the machine most of the time, except in certain cases, such as somephones running android on which the provider has replaced Google applications by its own applications.

➢ Widgets

With android, it is possible to use widgets which are small tools that can most often getinformation. These widgets are directly visible on the main window.

- Android Market

This is an online software store to buy applications. Developers who created applicationscan add them into the store, and these applications can be downloaded by users, they can be both free andpaid.

- Multitasking

Android allows multitasking in the sense that multiple applications can run simultaneously.With Task Manager it is possible view all running tasks and to switch from one to another easily.

- Messaging

SMS and MMS are available forms of messaging, including threaded text messaging and Android Cloud to Device Messaging (C2DM) and now enhanced version of C2DM, Android Google Cloud Messaging (GCM) is also a part of Android Push Messaging service.

- Voice based features

Google search through voice has been available since initial release.[4] Voice actions for calling, texting, navigation, etc. are supported on Android 2.2 onwards.[5] As of Android 4.1, Google has expanded Vce Actions with ability to talk back and read answers from Google's Knowledge Graph when queried with specific commands.[6] The ability to control hardware has not yet been implemented.

- Connectivity

Android supports connectivity technologies including GSM/EDGE, Wi-Fi, Bluetooth, LTE, CDMA, EV-DO, UMTS, NFC, IDEN and WiMAX.

- Bluetooth

Supports voice dialing and sending contacts between phones, sending files (OPP), accessing the phone book (PBAP), A2DP and AVRCP. Keyboard, mouse and joystick (HID) support is available in Android 3.1+, and in earlier versions through manufacturer customizations and third-party applications.[12]

- Media support

Android supports the following audio/video/still media formats: WebM, H.263, H.264, AAC, HE-AAC (in 3GP or MP4 container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), MP3, MIDI, Ogg Vorbis, FLAC, WAV, JPEG, PNG, GIF, BMP, WebP.[3]

- External storage

Most Android devices include microSD slot and can read microSD cards formatted with FAT32, Ext3 or Ext4 file system. To allow use of high-capacity storage media such as USB flash drives and USB HDDs, many Android tablets also include USB 'A' receptacle. Storage formatted with FAT32 is handled by Linux Kernel VFAT driver, while 3rd party solutions are required to handle other popular file systems such as NTFS, HFS Plus and exFAT.

- Java support

While most Android applications are written in Java, there is no Java Virtual Machine in the platform and Java byte code is not executed. Java classes are compiled into Dalvik executables and run on Dalvik, a specialized virtual machine designed specifically for Android and optimized for battery-powered mobile devices with limited memory and CPU. J2ME support can be provided via third-party applications.

- Storage

SQLite, a lightweight relational database, is used for data storage purposes.

The basics of creating Android apps:

To begin to program for Android I needed some basics, because some elements are very different, even if programming an application in Android uses the Java language, therefore, an object oriented language. Firstly, in an Android application, there is no main method:

public static void main(String[] args){...}

This method that allows to launch a program in java is not present in an application android. This example is only the first of a long list.

Activity:

An activity is a user interface that allows the user to interact with the screen, to perform actions.For example, a text messaging application could have an activity that displays a list of contacts to sendmessages. Once the contact is selected , activity could send information to a second activity that couldserve to send the message to the contact.

When an application is launched, what it displays is the result of an activity. At the code level, forcreate an activity; you must create a class that extends the Activity class. An activity has a requiredonCreate () method. It is the main method. To interact with the program, through the activity, there must besomething displayed, that is why the activity, contains what is called views.

View:

a View is the basic building block for user interface components. A View occupies a rectangular areaon the screen. View is the base class for widgets, which are used to create interactive UI components(buttons, text fields, etc.).There is differents kinds of views, for example a ListView is able to display only aninteractive list of what you want to display, while a WebView allows you to to display a web page. As saidbefore, A view occupies a rectangular area on the screen. To organise these rectangles on the screen, thereis a text file written in XML for every different screen.

- **Android SDK**

The Android software development kit (SDK) includes a comprehensive set of development tools.[8] These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows XP or later. For the moment one can also develop Android software on Android itself by using the AIDE - Android IDE - Java, C++ app and the Java editor app. The officially supported integrated development environment (IDE) is Eclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box,[9] and NetBeans IDE also supports Android development via a plugin.[10] Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely)

❖ Java

Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacentres, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere.

If you want to get started with application development, Google provides a Java API to get started and compiles your files into classes. Why did Android prefer Java for its development platform? There are multiple reasons such as; Java is a commonly used language and many programmers know it, it can run on a virtual machine (VM) so no need to recompile for different phones, better security, many development tools available for Java, and Java is a known industry language with most phones compatible with it.
Though Google provides the Java API, Android does not use JVM to execute class files. Rather, it uses Dalvik Virtual Machine (DVM). The class files are compiled into Dalvik Executable (DEX) format, and bundled as Android Package (APK) along with other resources.With Java, if you are aware of object-oriented programming principles, creating applications for android will be much simpler than IOS app development.

❖ XML on Android

Xml means Extensible Mark-up Language. Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses. The goal of using Android's XML vocabulary is to quickly design UI layouts and the screen elements they contain, in the same way that creating web pages in HTML with a series of nested elements.

The Android platform is an open source mobile development platform. It gives you access to all aspects of the mobile device that it runs on, from low level graphics, to hardware like the camera on a phone. With so many things possible using Android, you might wonder why you need to bother with XML. It is not that working with XML is so interesting; it is working with the things that it enables. XML is commonly used as a data format on the Internet. If you want to access data from the Internet, chances are that the data will be in the form of XML

Intent:

An activity can of course start another one, even if it but to do this, it will need a special objectcalled Intent. Intent is basis description of an operation to be performed. It can launch an Activity, send abroadcastIntent to any interested Broadcast Receiver components, and communicate with a backgroundService. An Intent performs binding between the code in different applications. it can be thought of as thelink between activities. It is possible to add some informations to an Intent, thanks to an object calledbundle, that you add to the intent thanks to the method :

Intent.putExtras(Bundle objetbunble);

Android Manifest:

AndroidManifest.xml file is necessary for all android applications and must have this name in itsroot directory. In the manifest you can find essential information about the application for the Androidsystem, information's that the system must have before it can run any of the application's code. Here iswhat you can find in the Android manifest:

-The name of the Java package for the application. The package name serves as a unique identifierfor the application.

-The description of the components of the application : the activities, services, broadcast receivers,and content providers that the application is composed of and under what conditions they can belaunched .

-The processes that will host application components.

-The permissions the application must have in order to access protected parts of the API andinteract with other applications.

-The permissions that others are required to have in order to interact with the applicationscomponents.

-The minimum level of the Android API that the application requires.

-The list of the libraries that the application must be linked against.

❖ IDE for Android

Android Studio is a new Android development environment based on IntelliJ IDEA. It provides new features and improvements over Eclipse ADT and will be the official Android IDE once it's ready. On top of the capabilities you expect from IntelliJ, Android Studio offers:

- Flexible Gradle-based build system
- Build variants and multiple APK generation.
- Expanded template support for Google Services and various device types.
- Rich layout editor with support for theme editing.
- Lint tools to catch performance, usability, version compatibility, and other problems.
- ProGuard and app-signing capabilities.

- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

❖ SQLite for databases

SQLite is an in-process library that implements a self-contained, server less, zero-configuration, transactional SQL database engine. It is the one database, which is zero-configured, that means like other database you do not need to configure it in your system. SQLite engine is not a standalone process like other databases; you can link it statically or dynamically as per your requirement with your application. The SQLite accesses its storage files directly.

Android provides several ways to store user and app data. SQLite is one way of storing user data. SQLite is a very light weight database which comes with Android OS. In this tutorial I'll be discussing how to write classes to handle all SQLite operations.

- SQLite does not require a separate server process or system to operate (server less).
- SQLite comes with zero-configuration, which means no setup or administration needed.
- A complete SQLite database is stored in a single cross-platform disk file.
- SQLite is very small and light weight, less than 400KiB fully configured or less than 250KiB with optional features omitted.
- SQLite is self-contained, which means no external dependencies.
- SQLite transactions are fully ACID-compliant, allowing safe access from multiple processes or threads.
- SQLite supports most of the query language features found in the SQL92 (SQL2) standard.
- SQLite is written in ANSI-C and provides simple and easy-to-use APl

To use a database, we will create a class called Helper. This class will allow us to manipulate thedatabase from any other class that has instantiated the object Helper. This class has elements and methodsvery specific . First specific objects: a SQLitedatabase, and a class called openHelper that we will also create.

```
private SQLiteDatabase db;
OpenHelper openHelper = new OpenHelper(this.context);
this.db = openHelper.getWritableDatabase();
```

The class OpenHelper extends SQLiteOpenHelper. This class is used to actually create one table orseveral tables in a database, and fill the table in the moment of its creation, al this in the method onCreate(). this class allows to update the version of the table with the method onUpgrade (). The method OnCreatewill be called only once, after that the table is created this method will no longer serve. For the class Helper,you can add all the methods used to select, add, edit or delete entries in the table.

❖ Access to Web Services (JSON)

Web service is software functionality that can be invoked through the internet using common protocols

- like a remote function(s) you can call by contacting a program on a web server
- many web services accept parameters and produce results
- can be written in PHP and contacted by the browser in HTML and/or Ajax code
- service's output might be HTML but could be text, XML, JSON or other content

The ability to describe sets of data in JSON format is a natural extension to the JavaScript language. Service Now supports a web service interface that operates on the JSON object as the data input and output format.

The JSON web service is provided by a platform-level processor similar to the services for SOAP, WSDL, CSV, Excel, and XML. Like those services, the JSON service is triggered by the standalone JSON URL parameter.

❖ E-draw Max for diagrams

E-draw Max enables students, teachers and business professionals to reliably create and publish various kinds of diagrams to represent any ideas. E-draw Max is an all-in-one diagram software that makes it simple to create professional-looking flowcharts, organizational charts, network diagrams, business presentations, building plans, mind maps, science illustration, fashion designs, UML diagrams, workflows, program structures, web design diagrams, electrical engineering diagrams, directional maps, database diagrams and more.

❖ SQLite browser

DB Browser for SQLite is a light GUI editor for SQLite databases, built on top of Qt. The main goal of the project is to allow non-technical users to create, modify and edit SQLite databases using a set of wizards and a spread sheet-like interface.

❖ Specific Requirements

Software Requirements

For developing the application the following are the Software Requirements:

Operating System: Windows 7
Language: SDK, Java
Tools: Android Studio
Technologies used: Java, XML.
Debugger: Android Dalvik Debug Monitor service For running the application the following are the Software Requirements:
Operating System : Android 2.4.1 or higher versions
Network : Wi-Fi Internet or cellular Network 3
Tools : Different API's and JSON data

Hardware Requirements

For developing the application the following are the Hardware Requirements:
Processor : Adreno or higher
RAM : 256 MB
Space on disk : minimum 512MB For running the application:
Device : Android version 2.4.1 and higher
Minimum space to execute : 8.0M

5. DESIGN

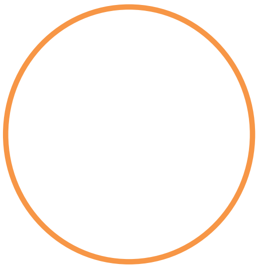
Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term "design" is defined as "the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization".

❖ Data Flow Diagram (DFD)

Notation Used

- **Process**

A function is represented using a circle. This sign is called a process or a bubble. Bubbles are annotated with the names of the corresponding functions.



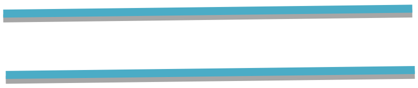
- **External Entity**

An external entity such as a librarian, a library member, etc. is represented by a rectangle. The external entities are essentially those physical entities which is external to the software system that interact with the system by inputting data to the system or by consuming the data produced by the system. In addition to the human users, the external entity symbols can be used to represent external hardware and software such as application software

- **Data Flow**

A directed arc or an arrow is used as a data flow sign. A data flow sign represents the data flow occurring between two processes, or between an external entity and a process, in the direction of the data flow arrow. Data flow signs are usually annotated with the corresponding data names.

- **Data Store**



A data store represents a logical file. It is represented using two parallel lines. A logical file can represent either a data store sign, which can represent either a data structure, or a physical file on disk. Each data store is connected to a process by means of a data flow sign. The direction of the data flow arrow shows whether data is being read from or written into a data store. A arrow flowing in or out of a data store implicitly represents the entire data of the data store and hence connecting to a data store need not be annotated with the name of the corresponding data items.

- **0 Level DFD**

A 0 level DFD defines the boundary between the system and outer environment. Such diagrams represent the system at the centre with no details of its interior structure. The main objective of this diagram is to focus attention on external factors and events that are important during development of a set of system requirements.

DFD Level - 0:

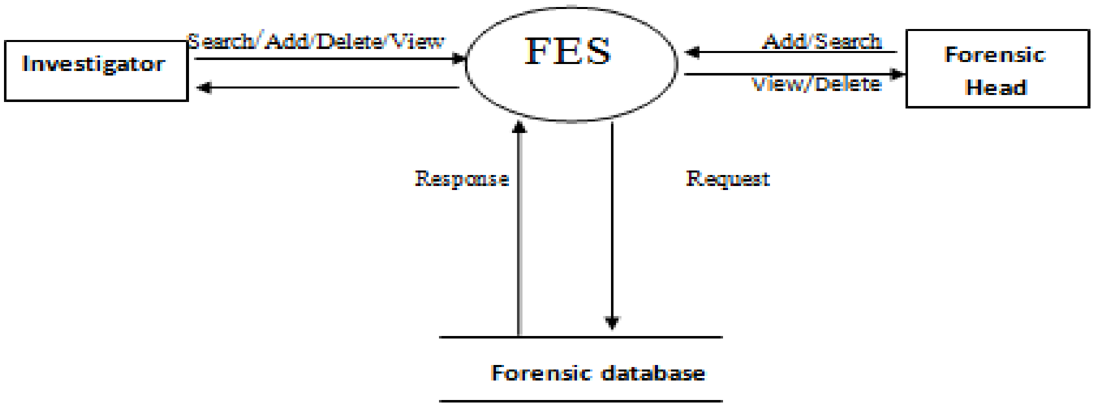


Fig1:DFD Level0

- **Data flow diagram (level 1)**

This shows the system divides into sub systems each of which deals with one or more of the data flow or form an external agent and which together provide all of the functionality of the system as a whole.

DFD for Admin (Forensic Head/Expert):

DFD Level 1

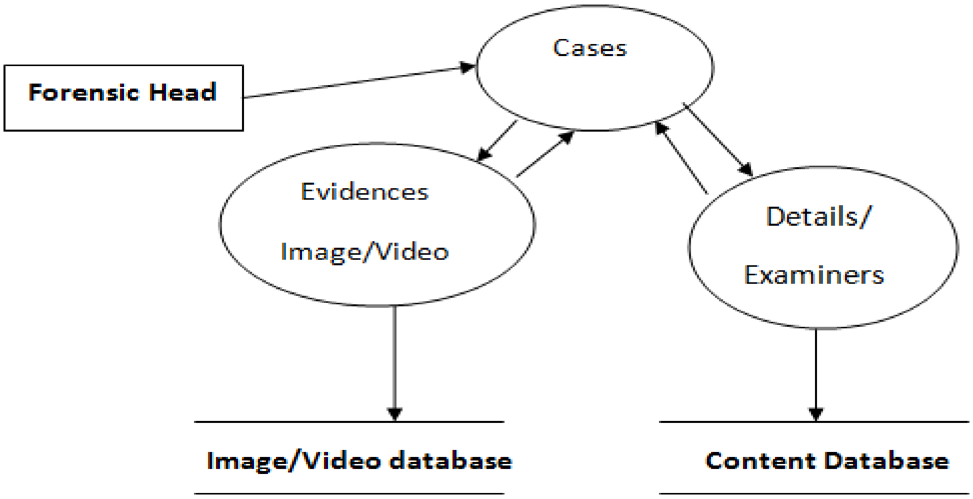


Fig2:DFD Level1

DFD of Admin (Forensic Head/Expert) for Cases Module:

DFD Level 2:

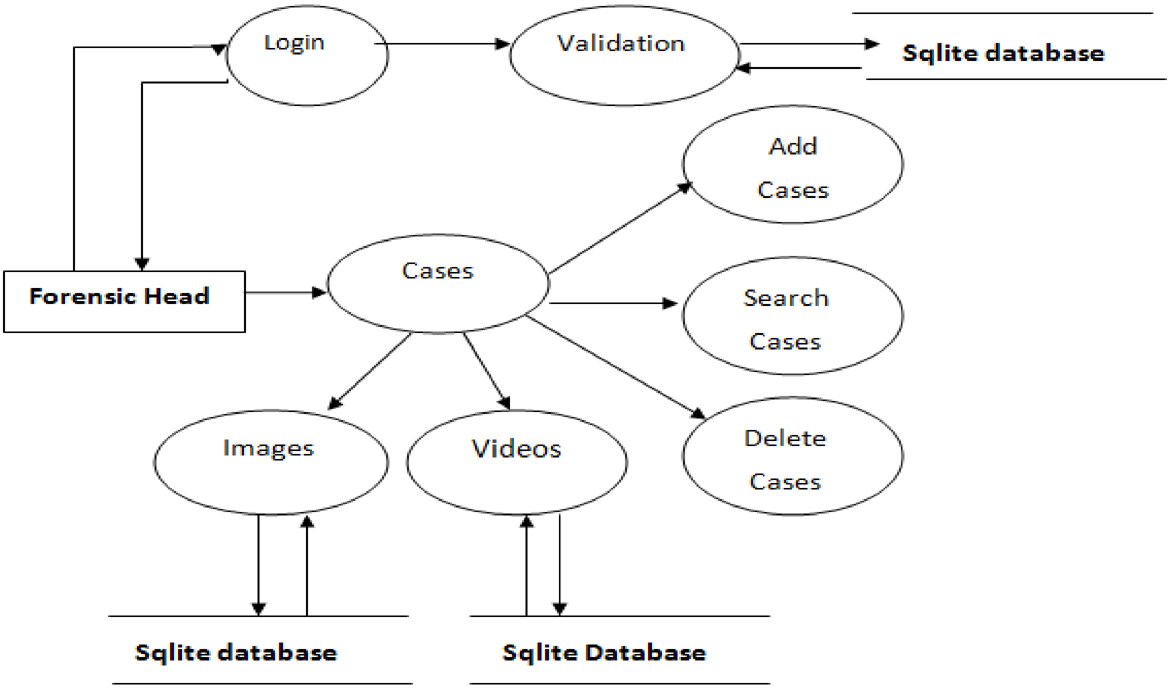


Fig3:DFD Level2

DFD of Admin (Forensic Head/Expert) for Users Module:

DFD Level 2:

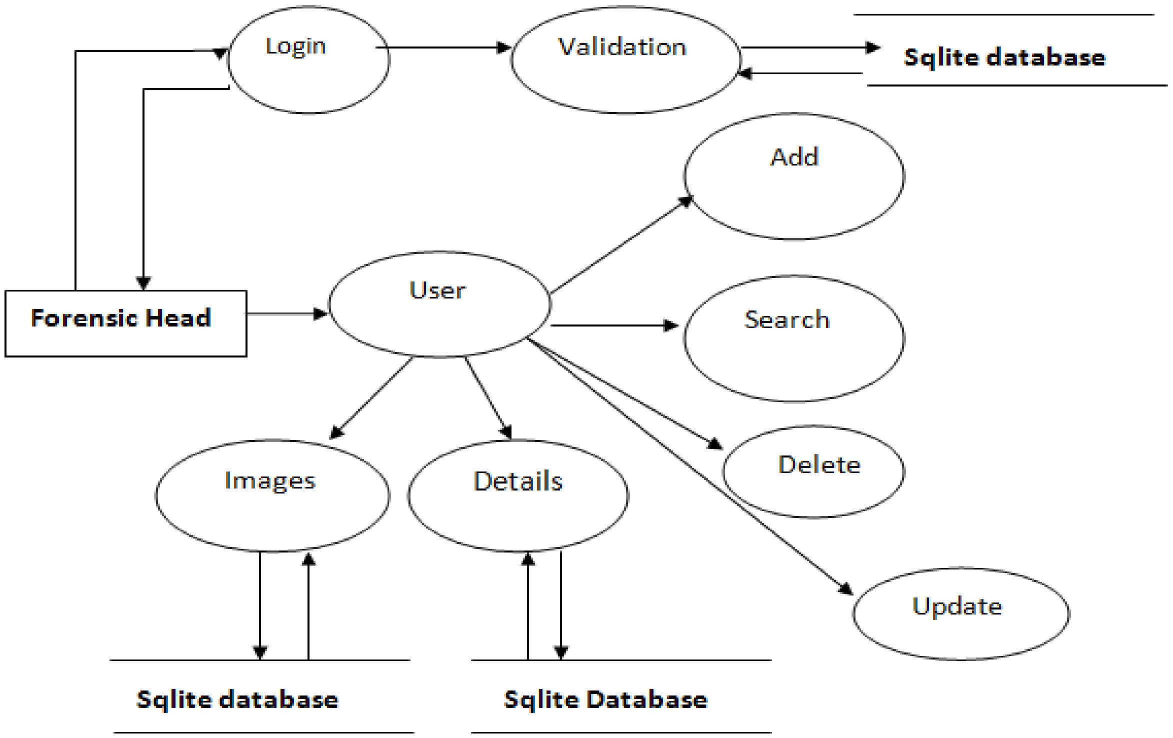


Fig4: DFD Level2

DFD of Investigators for Users Modules:

DFD Level 1

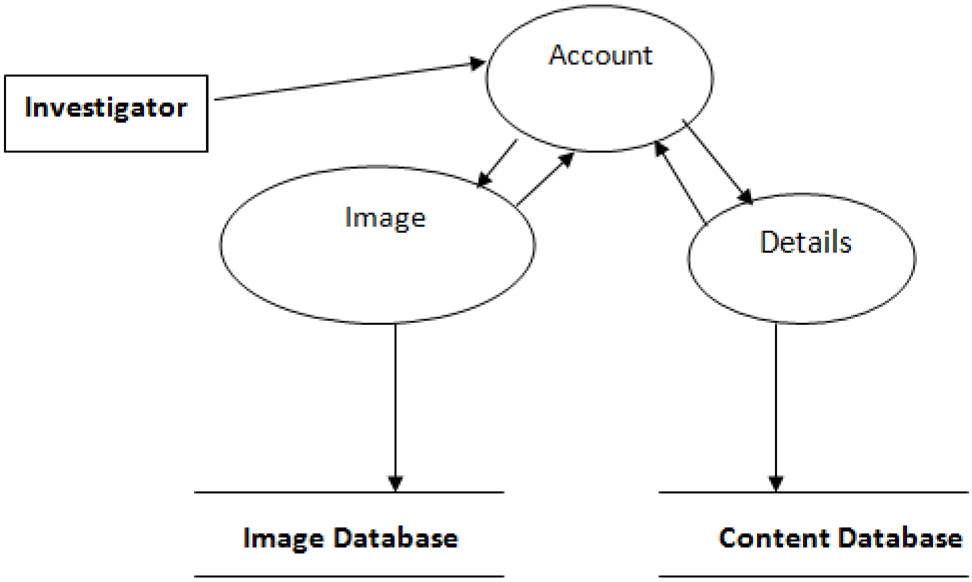


Fig5: DFD Level1

DFD Level 2:

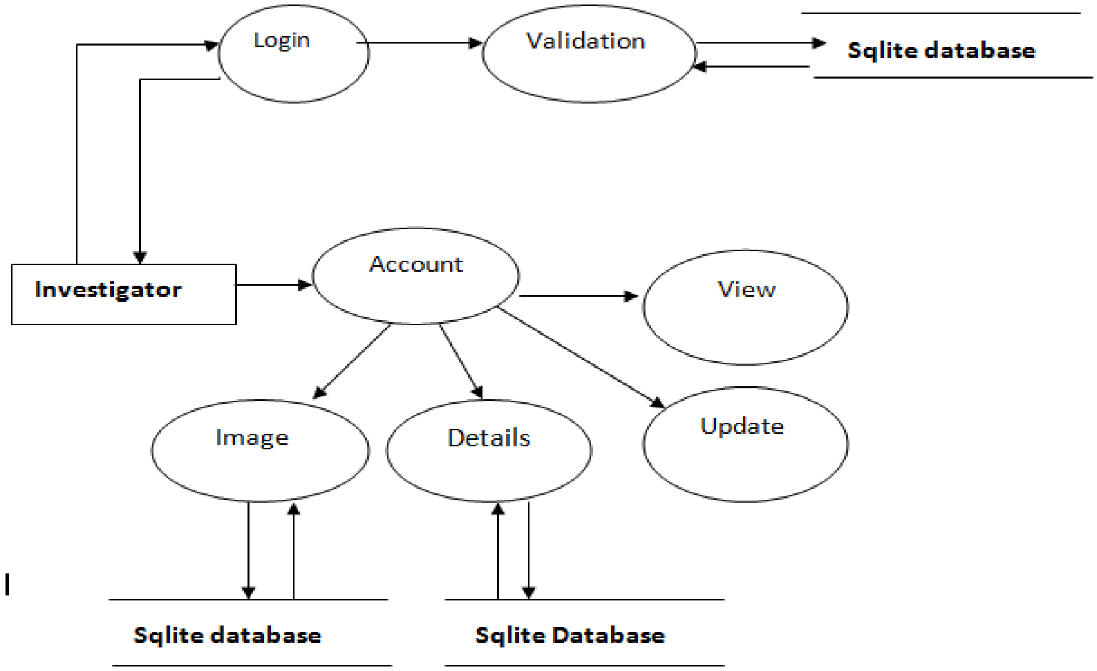


Fig6: DFD Level2

DFD Level 2:

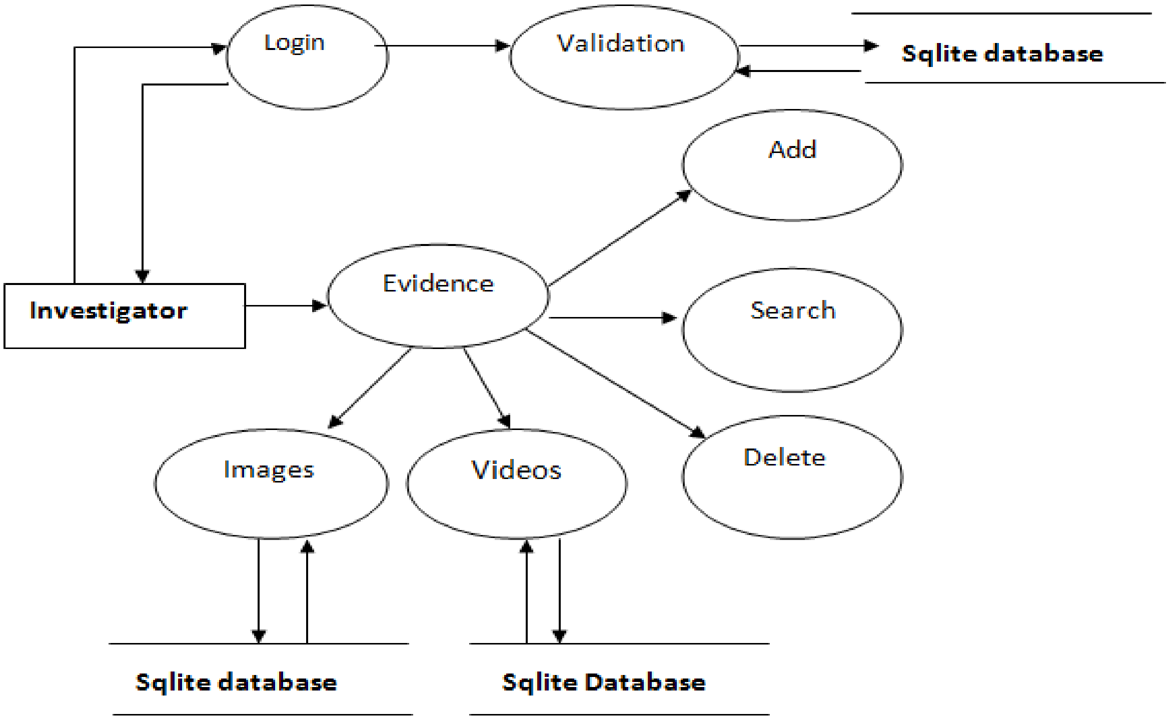


Fig7: DFD Level2

DFD for Admin Module:

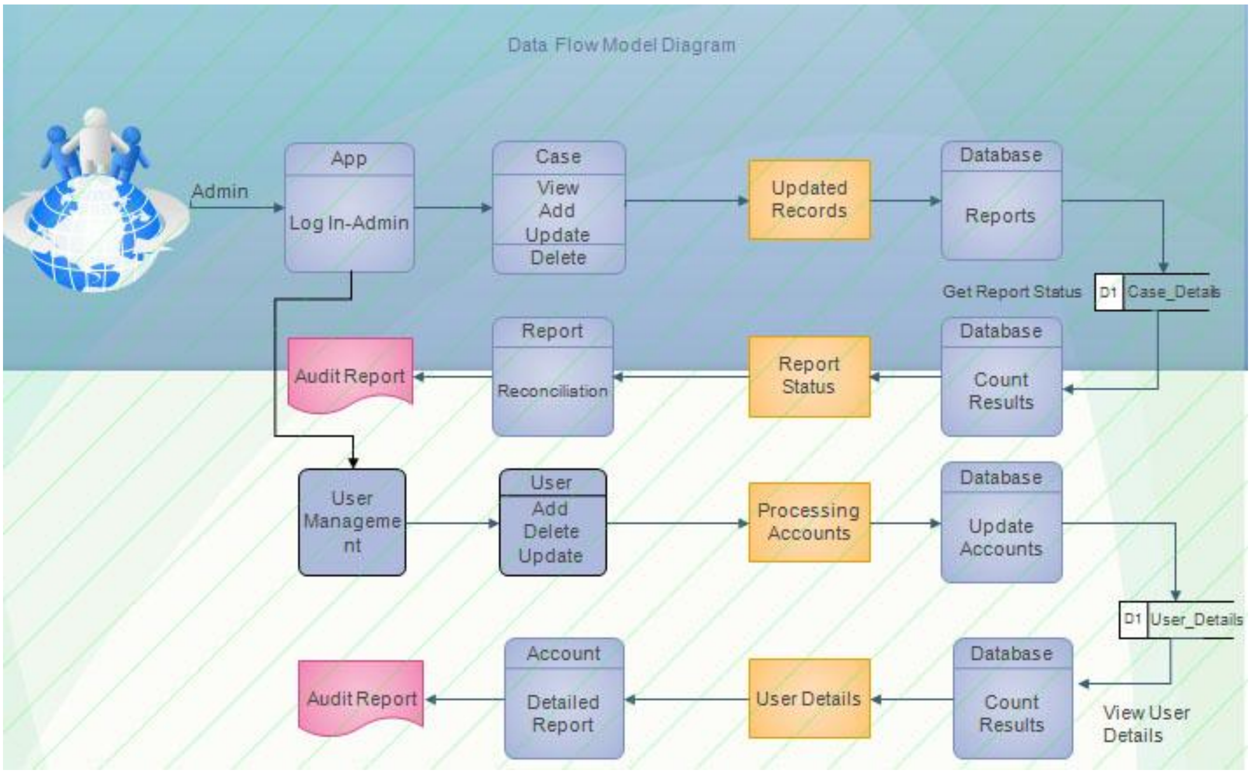


Fig8: DFD of Admin

DFD for Investigator Module:

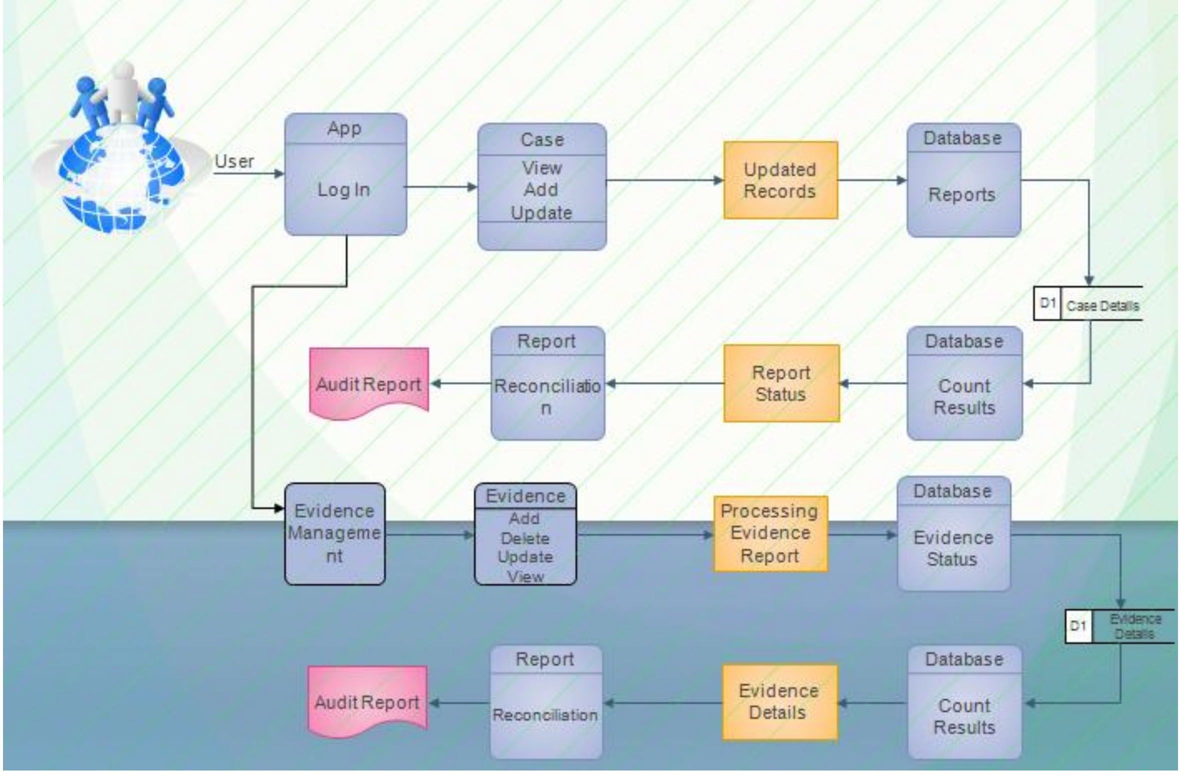


Fig9: DFD of User

❖ FLOW CHART

A Flow Chart Represent is an algorithm or process showing a step by step execution detail. Flow charts are help to visualize what is going on and thereby help the viewer to understand a process, find the flows, and the bottlenecks. The two most or common type of boxes in flow charts are

- ✓ A Process step usually called activity and donated as a rectangular box.
- ✓ A decision usually denoted as a diamond.

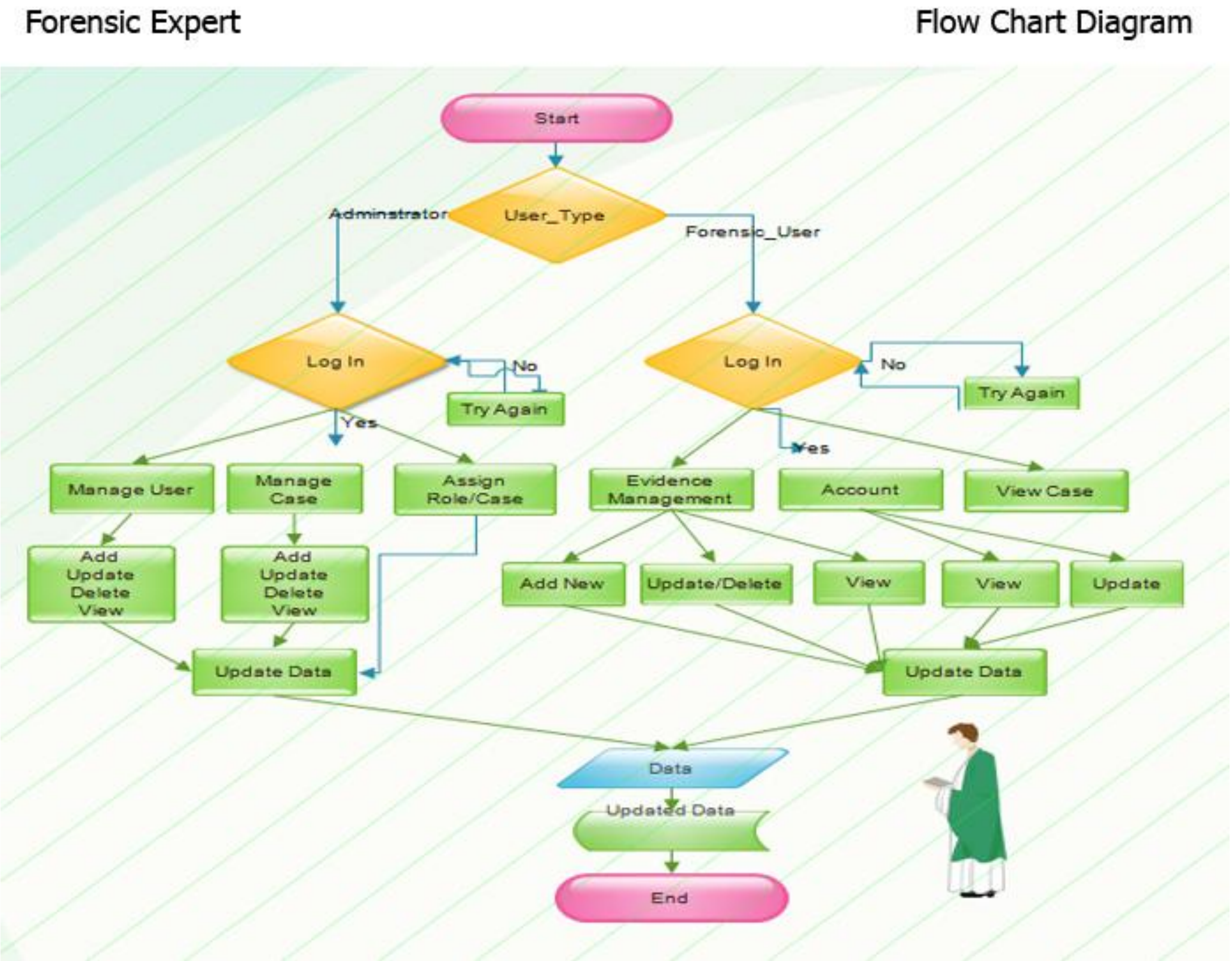


Fig10: Flowchart of Application

❖ USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

USE CASE DIAGRAM for Investigators:

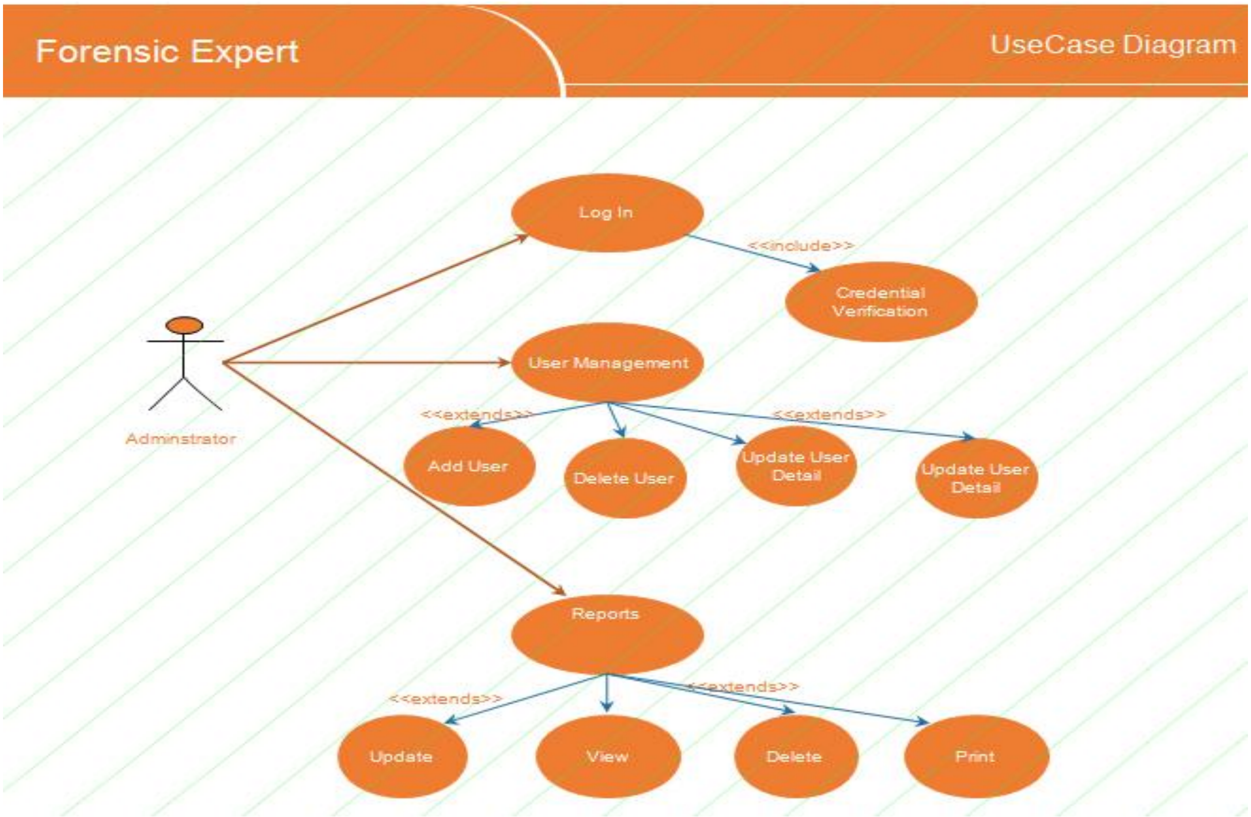


Fig11: Use Case of Admin

USE CASE DIAGRAM for Admin:

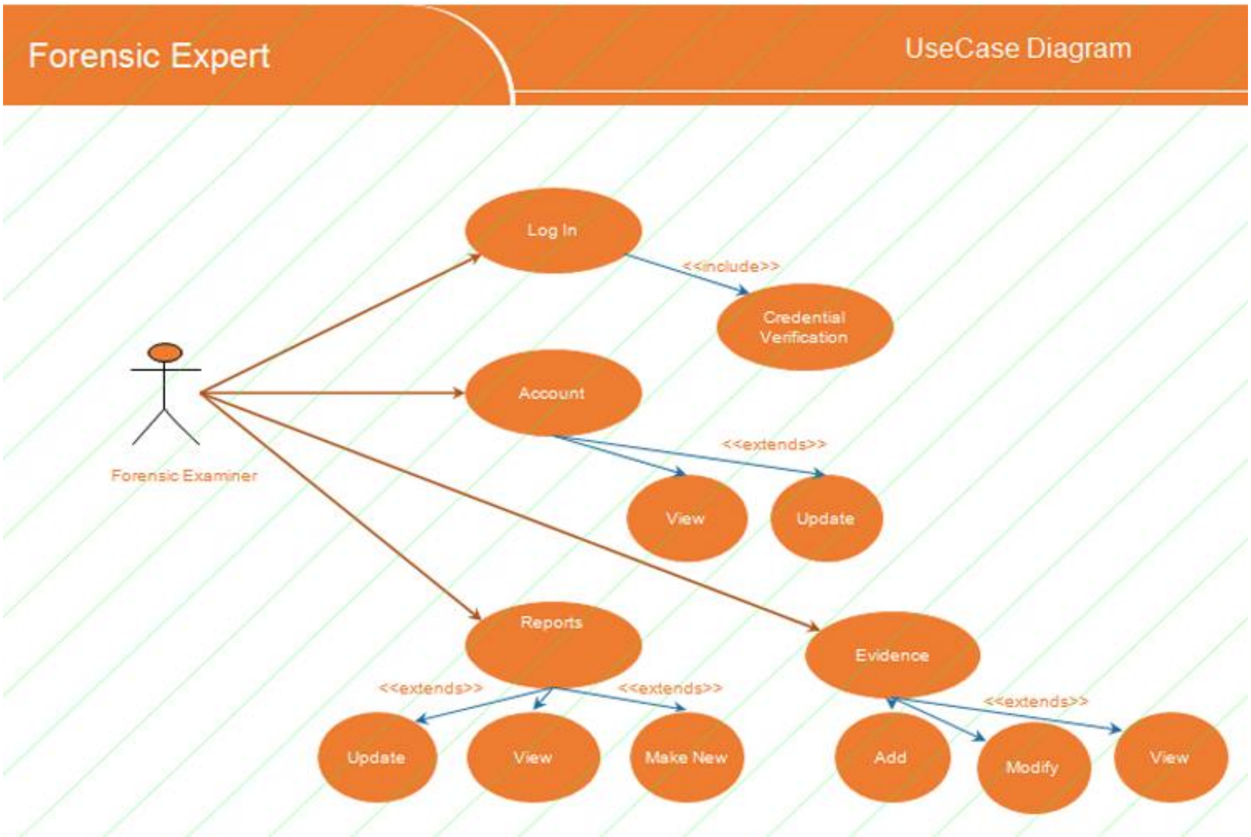


Fig12: Use Case of Forensic Examiner

❖ E-R DIAGRAM

For Investigators to add evidences:

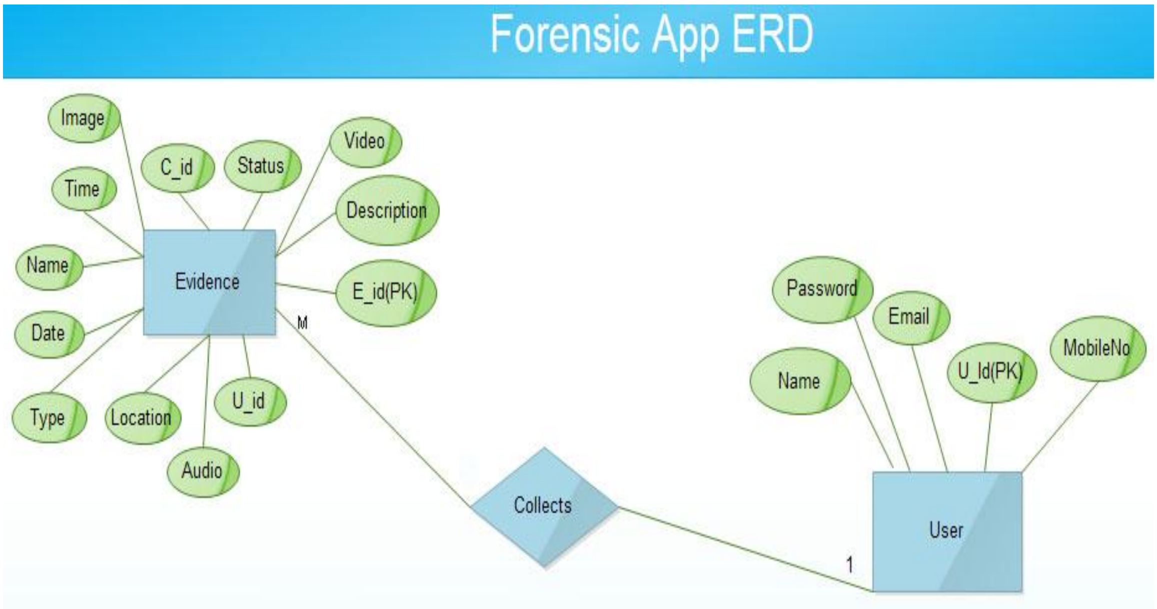


Fig13: ERD

For Investigators to add images:

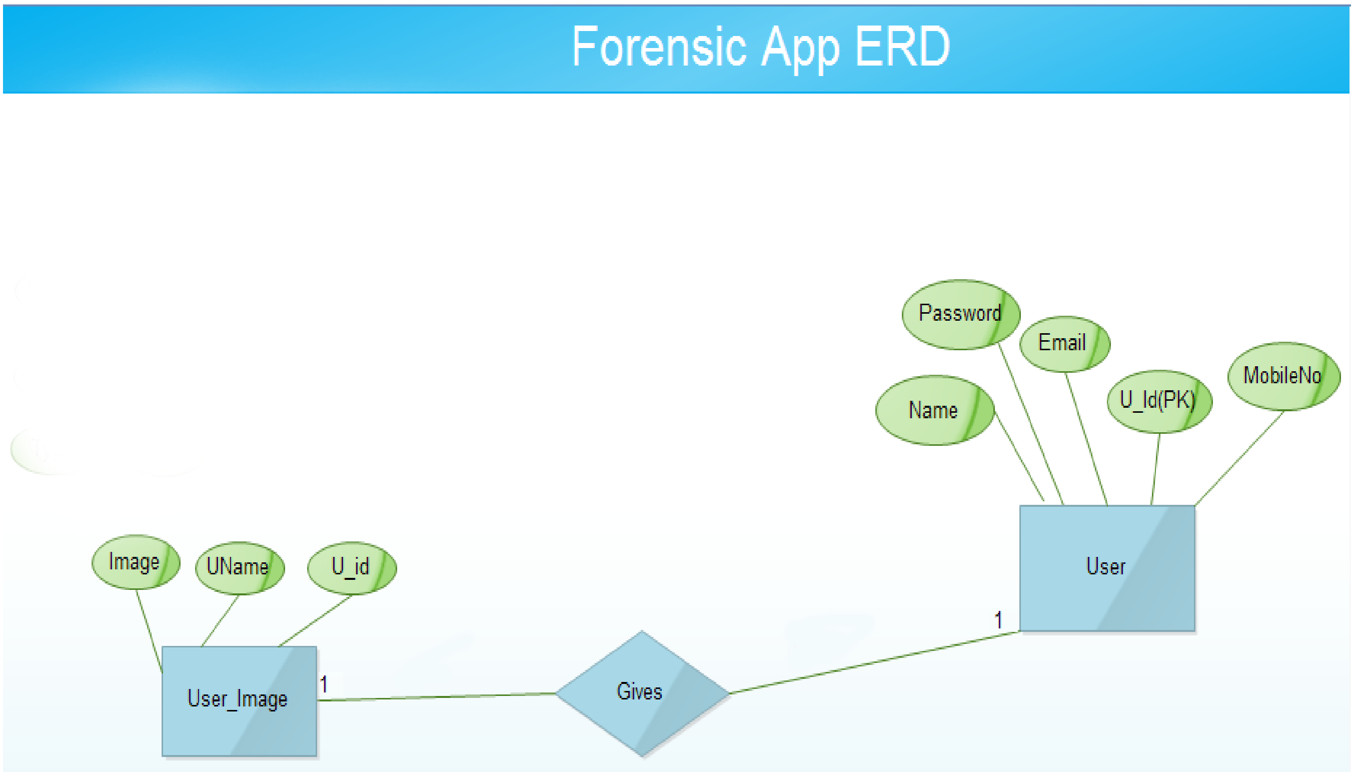


Fig14: ERD

For Admin to add cases:

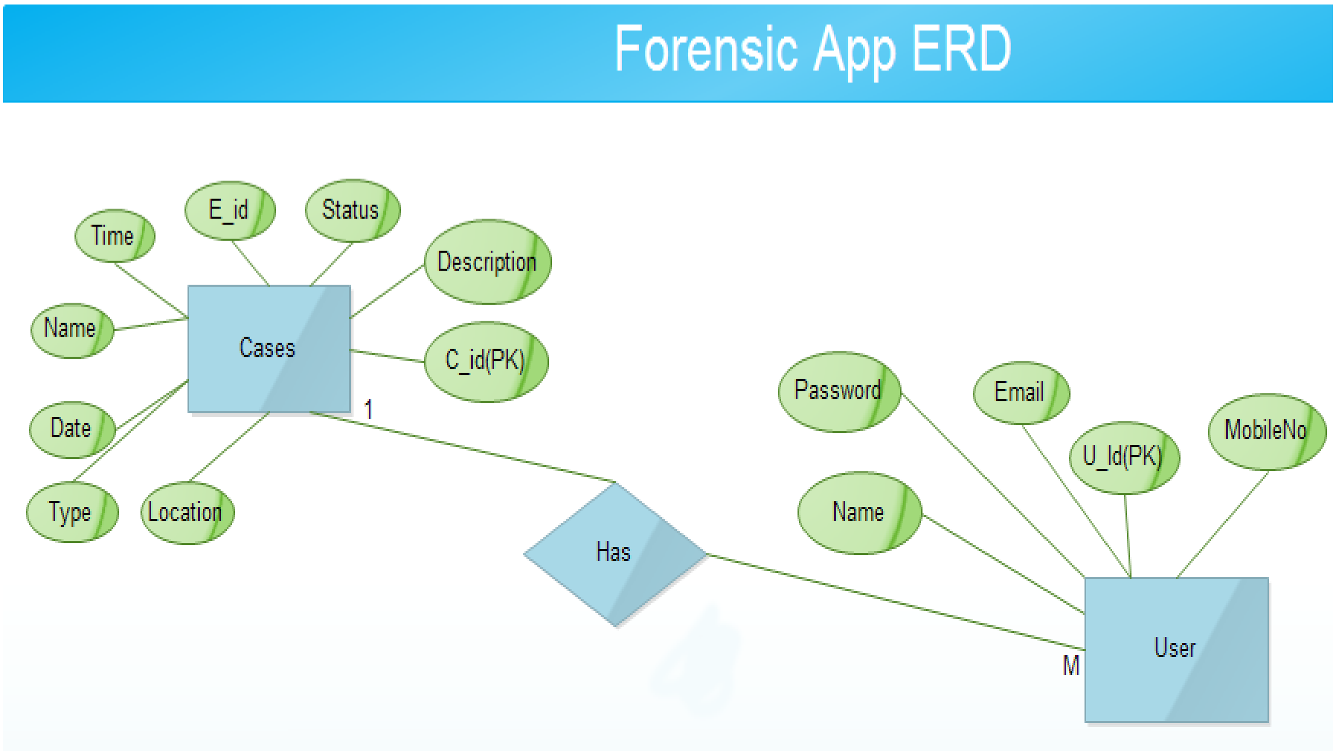


Fig15: ERD

For cases to add evidences:

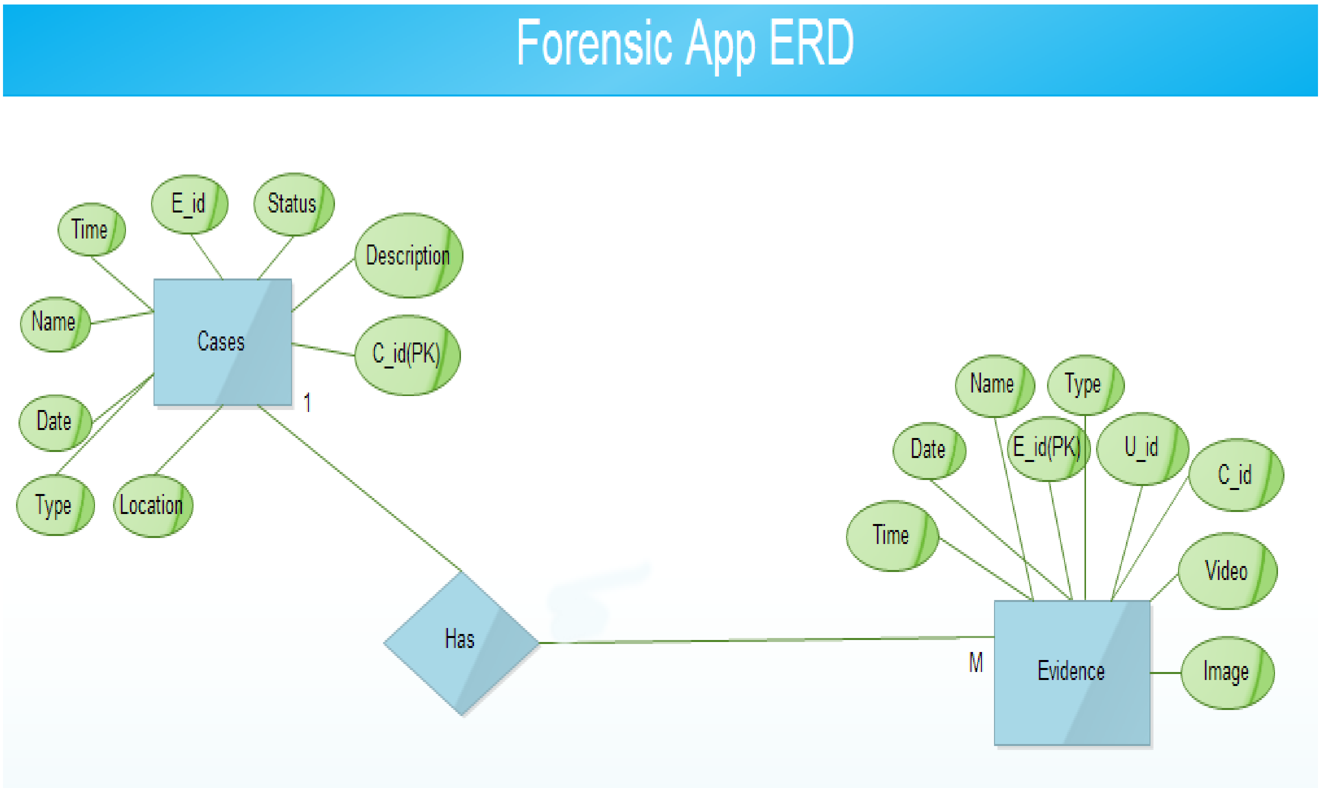


Fig16: ERD

ER – Diagram:

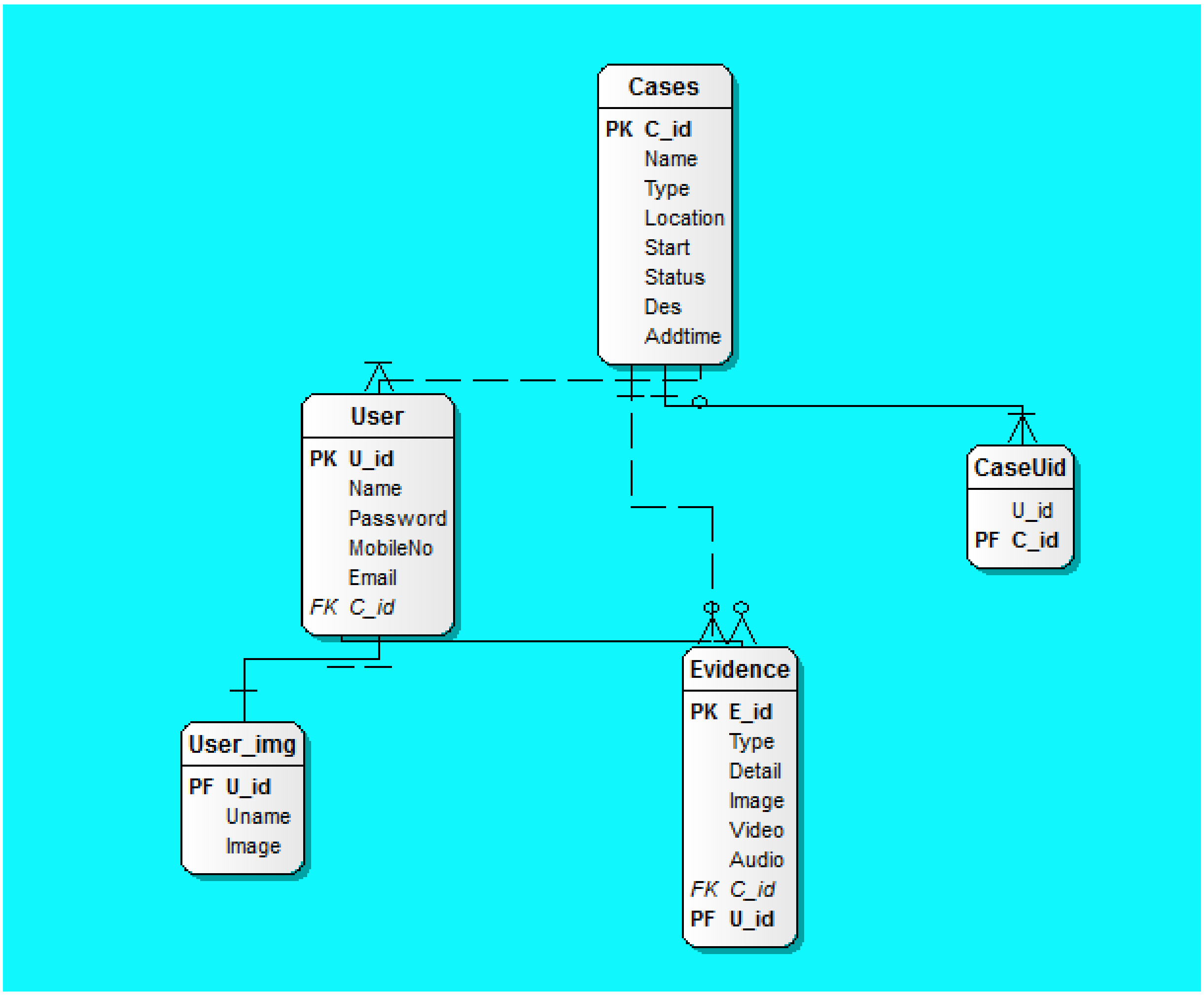


Fig17: ERD-Detail

❖ ACTIVITY DIAGRAM

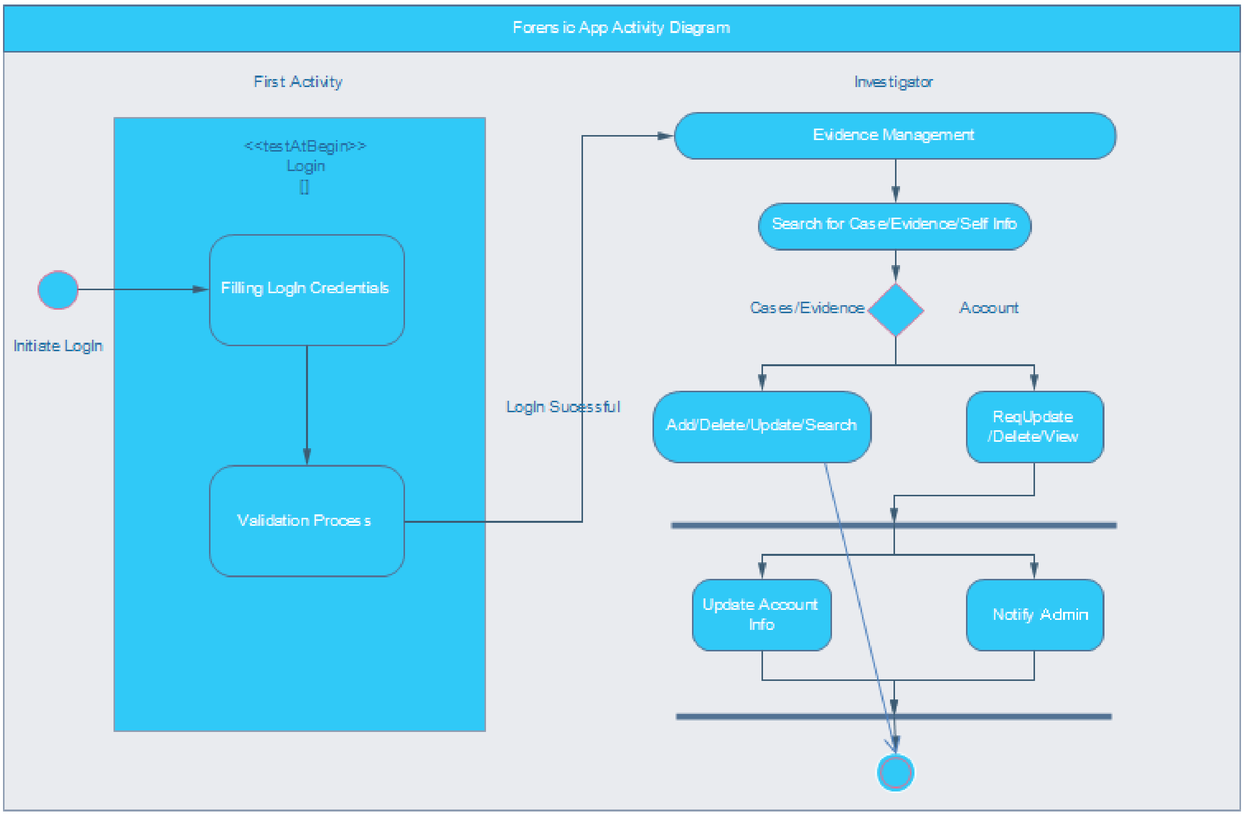


Fig18: Activity Diagram of Investigator

DATABASE DESIGN

Database structure:

admin_details

Column Name	Column Type	Description
user_id	integer	Primary_key
Name	Text	
Password	Text	
Mob no	Integer	
email	Text	

Table- **admin_details**

user_datail

user_id	integer	Primary_key
user_name	Text	
image	BLOB	

Table- **user_detail**

Evidence_detail

Evidence_id	integer	Primary_key
Caes_id	Text	
Investigator_id	integer	Foreign key
type	Text	
detail	Text	
image	Text	
Start	Text	

end	Text	
video	Text	

Table- **Evidence_detail**

case_detail

Case_id	integer	Foreign key
name	varchar	Primary_key
type	varchar	
location	varchar	
Status	Text	
start	text	
adddtime	Text	
Desr	Text	

Table- **case_detail**

victim_details 1

Case_no	Integer	Primary key
Victim_name	Varchar	
Ssn.	Varchar	unique
DOB	Varchar	
Contact	Varchar	

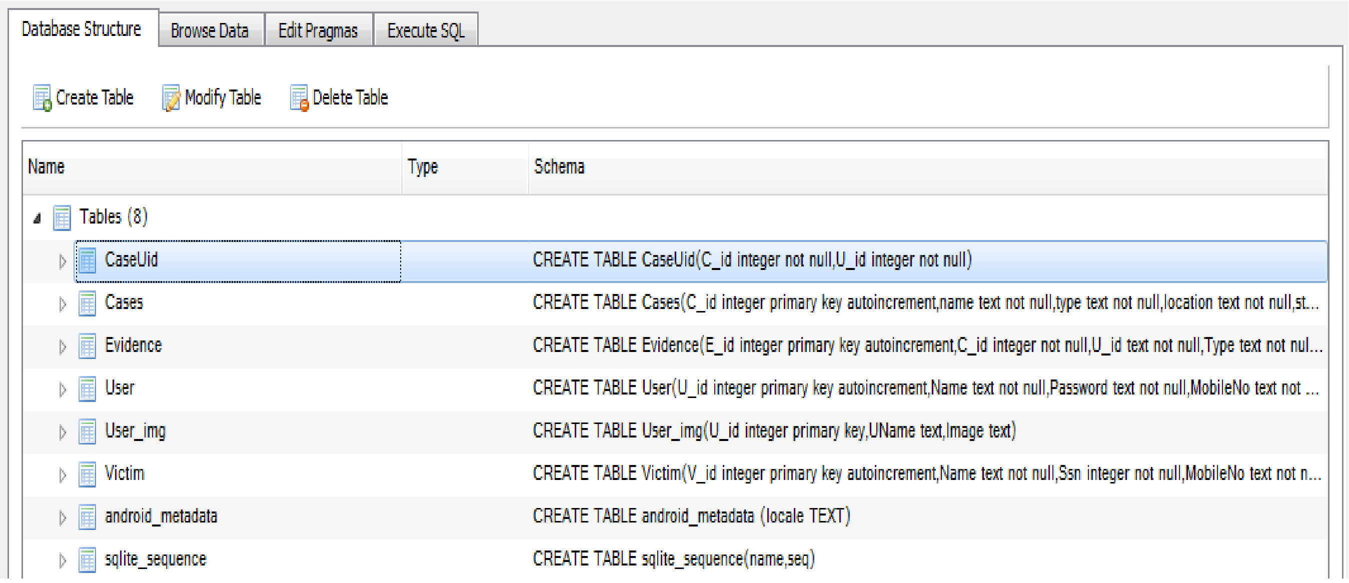
Table- **victim_details1**

Case_user 1

case_id	integer	Primary_key
user_id	Text	

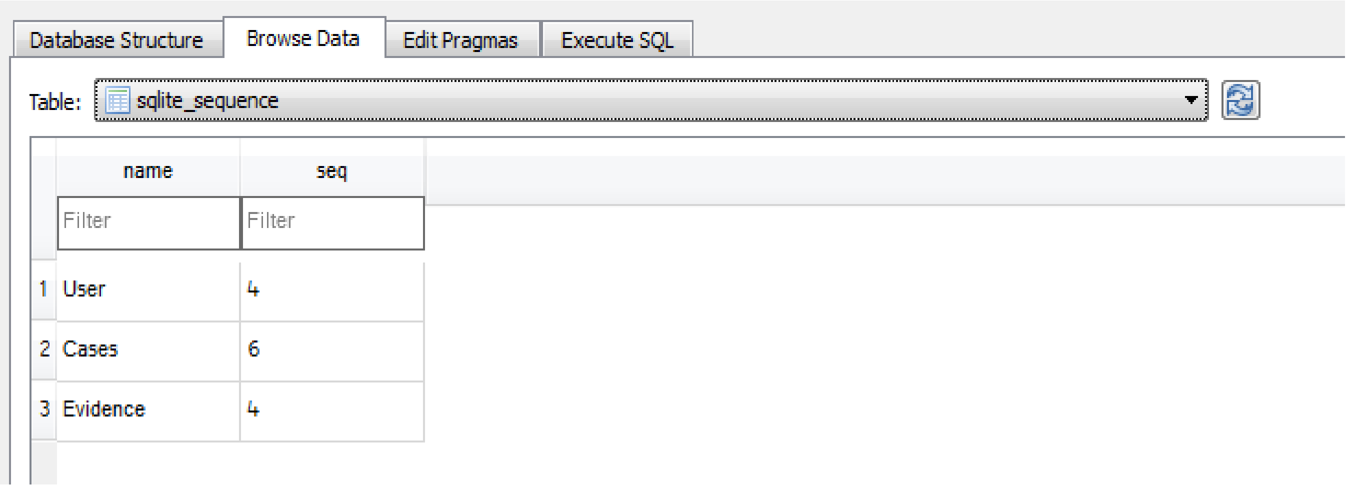
Table- **Case_user 1**

□ □**Table structure**



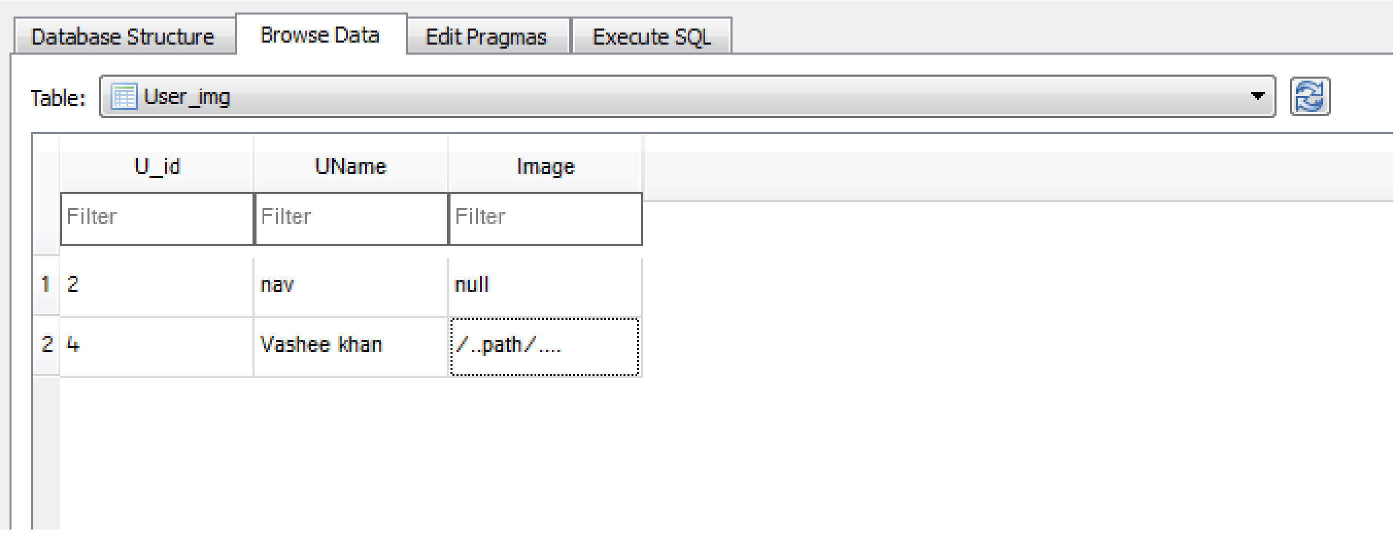
Name	Type	Schema
Tables (8)		
CaseUid		CREATE TABLE CaseUid(C_id integer not null,U_id integer not null)
Cases		CREATE TABLE Cases(C_id integer primary key autoincrement,name text not null,type text not null,location text not null,st...
Evidence		CREATE TABLE Evidence(E_id integer primary key autoincrement,C_id integer not null,U_id text not null,Type text not nul...
User		CREATE TABLE User(U_id integer primary key autoincrement,Name text not null,Password text not null,MobileNo text not ...
User_img		CREATE TABLE User_img(U_id integer primary key,UName text,Image text)
Victim		CREATE TABLE Victim(V_id integer primary key autoincrement,Name text not null,Ssn integer not null,MobileNo text not n...
android_metadata		CREATE TABLE android_metadata (locale TEXT)
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)

Fig 19: DB Structure



	name	seq
1	User	4
2	Cases	6
3	Evidence	4

Fig 20: sqlite_sequence



	U_id	UName	Image
1	2	nav	null
2	4	Vashee khan	/..path/....

Fig 21: User_img

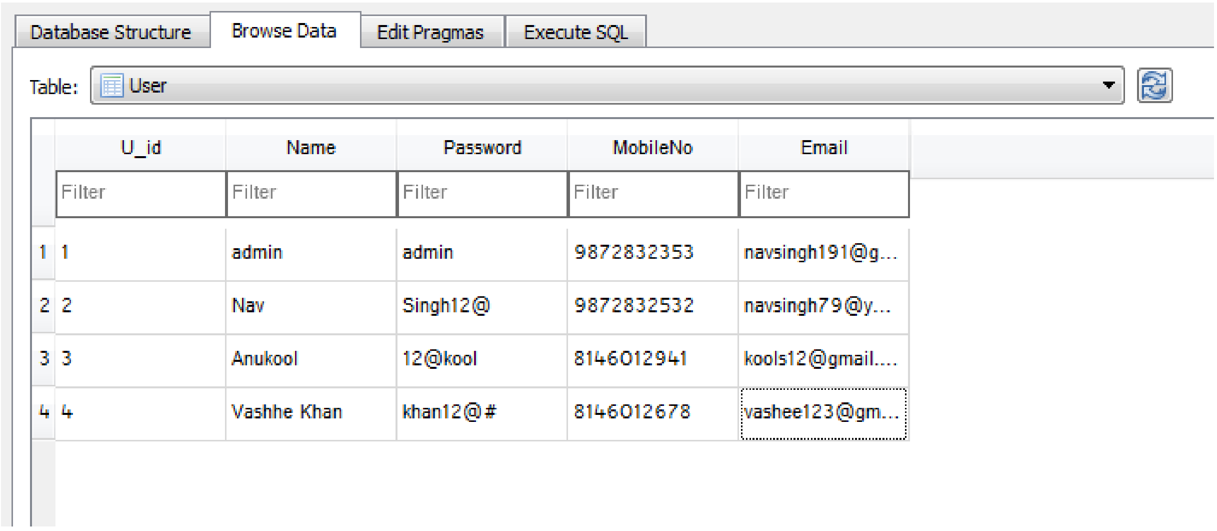


Table: User

	U_id	Name	Password	MobileNo	Email
	Filter	Filter	Filter	Filter	Filter
1	1	admin	admin	9872832353	navsingh191@g...
2	2	Nav	Singh12@	9872832532	navsingh79@y...
3	3	Anukool	12@kool	8146012941	kools12@gmail....
4	4	Vashhe Khan	khan12@#	8146012678	vashee123@gm...

Fig 22: user

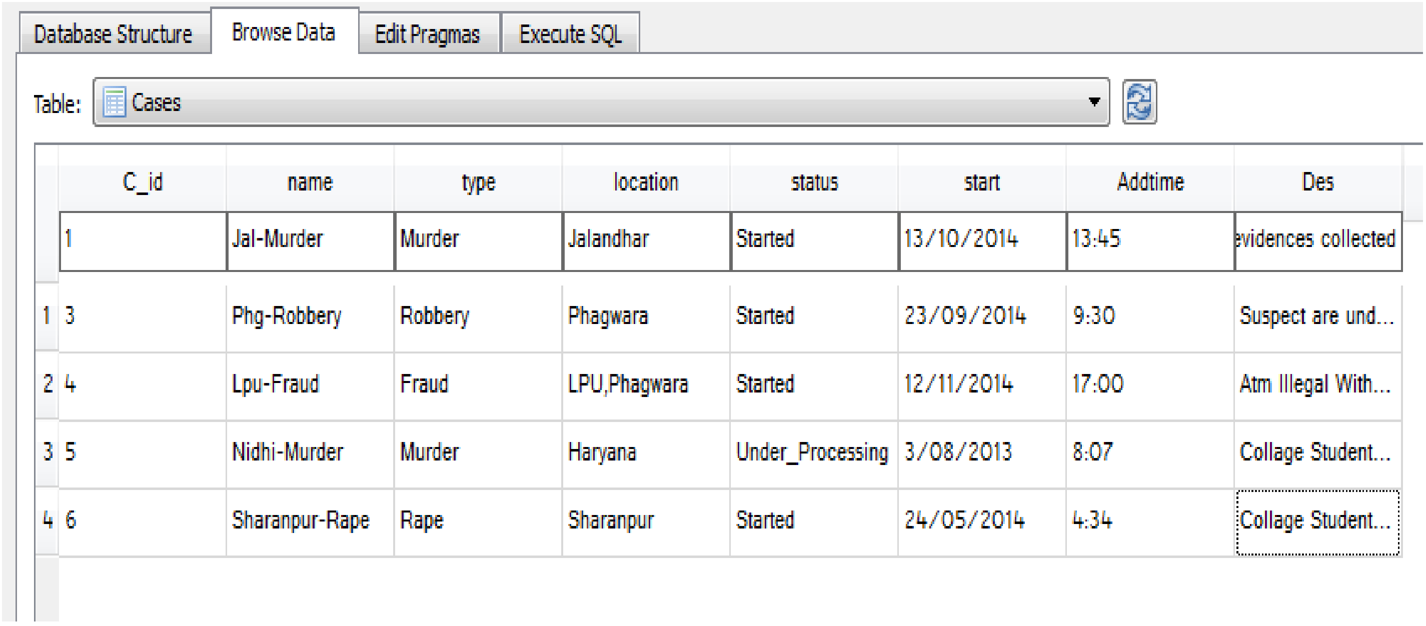
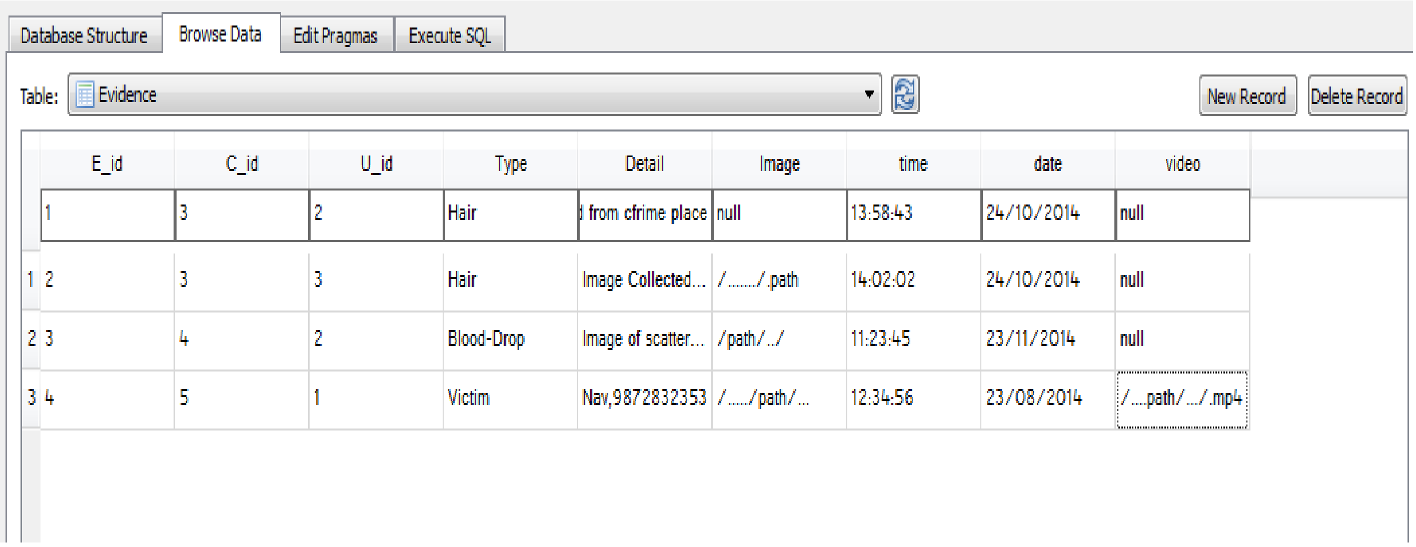


Table: Cases

	C_id	name	type	location	status	start	Addtime	Des
	1	Jal-Murder	Murder	Jalandhar	Started	13/10/2014	13:45	evidences collected
1	3	Phg-Robbery	Robbery	Phagwara	Started	23/09/2014	9:30	Suspect are und...
2	4	Lpu-Fraud	Fraud	LPU,Phagwara	Started	12/11/2014	17:00	Atm Illegal With...
3	5	Nidhi-Murder	Murder	Haryana	Under_Processing	3/08/2013	8:07	Collage Student...
4	6	Sharanpur-Rape	Rape	Sharanpur	Started	24/05/2014	4:34	Collage Student...

Fig23: Cases



	E_id	C_id	U_id	Type	Detail	Image	time	date	video
	1	3	2	Hair	d from cfrime place	null	13:58:43	24/10/2014	null
1	2	3	3	Hair	Image Collected...	/......./.path	14:02:02	24/10/2014	null
2	3	4	2	Blood-Drop	Image of scatter...	/path/../	11:23:45	23/11/2014	null
3	4	5	1	Victim	Nav,9872832353	/...../path/...	12:34:56	23/08/2014	/....path/.../.mp4

Fig 24: Evidence

6. TESTING

Software testing is an investigation conducted to provide information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks at the software implementation. Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Software Testing is the process of executing a program or system with the intent of finding errors. Software testing can also be stated as the process of validating and Adding that a software program/application/product.

Testing Objectives:

The main objectives of testing is to uncoCase a host of errors, systematically and with minimum effort ant time, starting formally, we can say,

- Testing is the process of executing a program with the intent of finding errors.
- A successful test is one that uncoCases an as yet of finding errors.
- A good test case is one that has a high probability of finding errors, if it exits.
- The tests are independent to detect possible errors.
- The software more or less confirms to the quality and reliable standards.

❖ Functional Testing

Functional Testing also called behavioural testing focuses on the functional requirements of the software. That is, it enables the software engineer to derived sets of input conditions that will fully exercise all functional requirements for a program.

Black Box Testing is a testing strategy, which does not need any knowledge of internal design or code etc. As the name "black box" suggests, no knowledge of internal logic or code structure is required. The types of testing under this strategy are totally based/focused on the testing for requirements and functionality of the work product/software application. It attempts to find errors in the following categories:

Incorrect or missing functions

- ✓ Interface errors
- ✓ Errors in data structures or external data base access
- ✓ Behaviour or performance errors

❖ Structural Testing

Structural Testing, also called glass-box testing, is a test case design philosophy that uses the control structure described as part of component level design to derive test cases. Structural testing strategy deals with the internal logic and structure of the code. This testing is also called as glass, structural, open box or clears box testing. In order to implement white box testing, the tester has to deal with the code and hence is needed to possess knowledge of coding and logic i.e. internal working of the code. White box test also needs the tester to look into the code and find out which unit/statement/chunk of the codes malfunctioning. Using white-box testing methods, one can derive test cases that:

- ✓ Guarantee that all independent paths within a module have been exercised at least once
- ✓ Exercise all logical decisions on their true and false sides

❖ Unit Testing

A Unit is the small testable piece of software that can be compiled for example functions, procedures, classes, and interface. It is generally written by the programmer and is done after coding Rigorous use of unit tests helps catch defects prior to customer testing and minimizes the cost of correcting defects by identifying them when they are the cheapest to fix. In computer programming, a **unit test** is a procedure used to validate that a particular module of source code is working properly. The procedure is to write test cases for all functions and methods so that whenever a change causes a regression, it can be quickly identified and fixed.

Unit testing focuses verification effort on the smallest unit of software design-the software component or module. Using the component-level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and the errors those tests uncover is limited by constrained scope established for unit testing. The unit test focuses on the internal processing logic and data structures within the boundaries of a component. This type of testing can be conducted in parallel for multiple components.

❖ Integration Testing

It is a systematic technique for constructing the software architecture while at the same time conducting tests to uncover errors interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design.

It is the testing phase, in which whole project developed in J2EE will be tested with integrating it with the various modules for the correct functionality. If working fine, then it will be integrated together with the other modules for purpose of creation of the whole System.

❖ System Testing

It is a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that systemelements have been properly integrated and performed allocated functions. It is the type of testing in which all the modules are integrated together to form the whole system, then this whole system is tested against various efficiency based tests, that also test the accuracy of the system, working as a whole.

- **Acceptance Testing**

This testing is performed with realistic data of the user to demonstrate that the software is working satisfactory. Testing here focuses on the external behaviour of the system, the internal logic of the program is not emphasized. Unlike the other test phases, an objective of acceptance testing is not to actively look for faults.

(a) Alpha Testing:

Alpha testing is testing of an application when development is about to complete. Minor design changes can still be made as a result of alpha testing. Alpha testing is typically performed by a group that is independent of the design team, but still within the company, e.g. in-house software test engineers, or software QA engineers.

(b) Beta Testing:

Beta testing is testing of an application when development and testing are essentially completed and final bugs and problems need to be found before the final release. Beta testing is typically performed by end-users or others, not programmers, software engineers or test engineers.

7. TEST CASES OF PROJECT

MODULE-1 (Add USER Activity)-

TEST SUIT ID	**CREATED BY**		**CREATED ON**
Forensic Expert App	Navjot Singh		31-10-2014
IMPLEMENTED BY: Navjot Singh		IMPLEMENTED ON: 02-11-2014	

Pre-Requisites:

Name:

Valid Classes: A-Z,a-z,spaces,Numeric(0-9).

Valid Range: Min-1, Max- 25. Please Enter a Name.

Invalid Classes: Special characters,Alphanumeric values,Float values,Fraction values.

Invalid range: Less than 1 & more than 25.

Email:

Valid Classes: A-Z,a-z,Numeric,Alphanumeric values,Float values,Special characters: . _ @

Valid Range: Max:50.Email id is mandatory.

Invalid Classes:Special characters except: . _ @,Fraction values, Email Id without domain,Upper or lower case with space,Registration through one id is done then,again user can't use that id.

Invalid Range: More than 50.

Contact Number:

Valid Classes:Numeric values(0-9).

Valid Range: Fix-10.Contact number is mandatory.

Invalid Classes:Alphanumeric,Special characters,Float values,Fraction values,A-Z,a-z,Alphanumeric with special characters.

Invalid Range:Less than 10 or more than 10.

Description:

Valid Classes:A-Z,a-z,Spaces,Numeric,Float Values ,Alphanumeric,Special characters: # - . / ( ) [ ],Alphanumeric with special characters.

Valid Range: Min-1, Max:200.Address is Required.

Invalid Classes:Fraction values,Special characters except: # - . / ( ) [ ]

Invalid Range: Less than 1 & more than 200.

Account Details-

Login Id:

Valid classes: Numeric(0-9).

Valid Range: Numeric Auto-Generated.

Password:

Valid Classes:A-Z,a-z,Spaces,Numeric,Alphanumeric values,Float values,Numeric(0-9).

Valid Range: Min-6,Max-20.Password is mandatory.

Invalid Class: Spaces,Decrypted Password.

Invalid Range: Not be Empty.

Upload Image:

Valid Class: User may select an image.

Add Button:

Valid Class: After filling all information click on register button.

Invalid Classes: Without selecting any information or one information.

Table: Add User Test Case

Test Case ID	**Description**	**Expected Result**	**Actual Result**	**Remarks**
TC Reg01	Input any characters less than 1 .	It must not be allowed to enter less than 1 .	It is not allowed to enter less than 1 .	Pass.
TC Reg02	Input any upper case character value(e.g.- AD) in "Name" field.	Upper case must be accepted.	Upper case are allowed.	Pass.
TC Reg03	Input any lower case character(e.g.- ar) value in "Name" field.	Lower case must be accepted.	Lower case are allowed.	Pass.
TC Reg04	Input any lower case or upper case multiple strings with one spaces(e.g.- sw ty) in "Name" field.	Lower or upper case multiple strings with one spaces must be accepted.	Lower or upper case multiple strings with one spaces are allowed.	Pass.
TC Reg05	Input numeric values(0-9)(e.g.- 45) in the "Name" field.	It must be allowed to enter numeric value(0-9) in "Name' field.	It is allowed to enter numeric value(0-9) in "Name' field.	Pass.
TC Reg06	Input any characters within range of(1-25) in "Name" field.	It must be allowed to be entered only range of (1-25) characters.	It is allowed to be entered only range of (1-25) characters.	Pass.
TC Reg07	Input only upper or lower case character with spaces in "Email" field.	Lower or upper case characters with spaces must not be accepted. It must shown a validation error message.(Enter Email in correct format).	It is not accepting the lower or upper case with spaces. It is displaying an error.(Enter Email in Correct Format).	Pass.
TC Reg08	Input any special characters: other than . _ @ with alphabetic values(e.g.-a%) in "Email" field.	Special characters other than .,_& @ must not be accepted and it must shown a validation error message.(Enter Email in correct format).	Special characters other than .,_& @ are not allowed and it is displaying a validation error message.(Enter Email in correct format).	Pass.
TC Reg09	Input any number of character less than 1 in Email Field.	Characters must not be accepted less than 1 .It must display a validation error message.(Enter Email in correct format).	Characters are not allowed less than 1 .It is displaying a validation error message.(Enter Email in correct format).	Pass.
TC Reg10	Registration through one Email id is done earlier. Then, again user is registering with same Email Id.	It must display a error message. It will not be allowed to register more than one time with same Email Id .	It is not allowing to register more than one time with same Email Id.	Pass.

TC Reg11	Input any Fraction values(e.g.-6/5) in “Email Id” field.	Fraction values must not be accepted. It must display a validation error message.(Enter Email In Correct Format).	It is not allowing fraction values and is display a validation error message.(Enter Email In Correct Format).	Pass.
TC Reg12	Input only lower case character values(e.g.-er) in “Email Id” field.	Lower case must be accepted.	Lower case are accepted.	Pass.
TC Reg13	Input only upper case character values(e.g.-ET) in “Email Id” field.	Upper case must be accepted.	Upper case are accepted.	Pass.
TC Reg14	Input any(0-9) number (e.g.- 432) in “Email Id” field.	Numbers must be allowed to be entered.	Numbers are allowed to be entered.	Pass.
TC Reg15	Input special characters: . _ @ with alphabetic values(e.g.-se@gmail.com) in “Email Id” field.	Special characters : . _ @ must be allowed to be entered.	Special characters : . _ @ are allowed to be entered.	Pass.
TC Reg16	Input any alphanumeric values(e.g.- as23) in “Email Id” field.	Alphanumeric values must be allowed to be entered.	Alphanumeric values are allowed to be entered.	Pass.
TC Reg17	Input any Float values(e.g.-4.6) in “Email Id” field.	Float values must be accepted.	Float values are allowed.	Pass.
TC Reg18	Input any number of character within range (1-50) in “Email Id” field..	Characters must be accepted.	Characters are allowed.	Pass.
TC Reg19	Entering special characters (e.g.-*&^)in the “Contact No.” field.	It must not be allowed to enter special characters. It must display a validation error message.(Please Enter Correct Contact Number)	It is not accepting special characters. It is displaying a validation error message.(Please Enter Correct Contact Number)	Pass.
TC Reg20	Try to enter more than or less than 10 numeric values in “Contact No.” field.	User can not enter more than or less than 10 numeric values in the “Contact No.” field.	User are not allowed to enter more than or less than 10 numeric values in the “Contact No.” field.	Pass.
TC Reg21	Input float values(e.g.-3.8) in “Contact No.”	It must not be allowed to enter float values in	It is not allowed to enter float values in “Contact No.” filed.	Pass.

	field.	“Contact No.” filed. It must display a validation error message.(Please Enter Correct Contact Number)	It is displaying a validation error message.(Please Enter Correct Contact Number)	
TC Reg22	Input fraction values(e.g.- 2/7) in “Contact No.” field.	It must not be allowed to enter fraction values in “Contact No.” filed. It must display a validation error message.(Please Enter Correct Contact Number)	It is displaying a validation error message.(Please Enter Correct Contact Number)	Pass.
TC Reg23	Entering lower case character(e.g.-tery) in the “Contact No.” field.	It must not be allowed to enter lower case character. It must display a validation error message.(Please Enter Correct Contact Number)	It is not accepting lower case character. It is displaying a validation error message.(Please Enter Correct Contact Number)	Pass.
TC Reg24	Entering upper case character (e.g.- EWSA) in the “Contact No.” field.	It must not be allowed to enter upper case character. It must display a validation error message.(Please Enter Correct Contact Number)	It is not accepting upper case character. It is displaying a validation error message.(Please Enter Correct Contact Number)	Pass.
TC Reg25	Input alphanumeric values (e.g.-Sa3)in the “Contact No.” field.	It must not be allowed to enter alphanumeric values. It must display a validation error message.(Please Enter Correct Contact Number)	It is not accepting alphanumeric values. It is displaying a validation error message.(Please Enter Correct Contact Number)	Pass.
TC Reg26	Input alphanumeric with special characters (e.g.-As3@)in “Contact No.” filed.	It will not be allowed to enter alphanumeric with special characters in “Contact No.” field. It must display a validation error message.(Please Enter Correct Contact Number)	It is not accepting alphanumeric with special characters values. It is displaying a validation error message.(Please Enter Correct Contact Number)	Pass.
TC Reg27	Input Numeric values(0-9)(e.g.-32) in “Contact No.” field.	Numeric values(0-9) must be accepted.	Numeric values(0-9) are accepted.	Pass.
TC Reg28	Input 10 digit Number in “Contact No” field.	Input must be accepted.	It is accepted.	Pass.
TC Reg29	Entering more than 200 characters in	It must not be allowed to enter more than 200	It is allowed to enter more than 200 characters.	Pass.

	“Description” field.	characters to be entered.		
TC Reg30	Input special characters except : # - . /( ) [ ] in the Description” field.	It must only allow # - . / ( ) [ ] these characters.	It allows all the special characters.	Fail.
TC Reg31	Input float values(e.g.- 8.4) in “Description” field.	It must allow to enter float values in “Description” field.	It is allowing to enter float values in “Description” field.	Pass.
TC Reg37	Input fraction values(e.g.- 3/2) in “Description” field.	It must be allowed to enter fraction values in “Description” field.	It is allowing to enter fraction values in “Description” field.	Pass.
TC Reg38	Input lower case characters(e.g.-rt) in “Description” field.	Lower case characters must be allowed.	Lower case characters are allowed.	Pass.
TC Reg39	Input upper case characters(e.g.- SDE) in “Description” field.	Upper case characters must be allowed.	Upper case characters are allowed.	Pass.
TC Reg40	Input alphanumeric values(e.g.- aS2) in “Description” field.	Alphanumeric values must be allowed.	Alphanumeric values are allowed.	Pass.
TC Reg41	Input alphanumeric values with special characters (e.g.- As2@)in “A Description” field.	It must allow all the alphanumeric values with special characters in the multi line text field.	It allows all alphanumeric with special characters.	Pass.
TC Reg42	Input only lower or upper case with spaces in “Password” field.	It must not be allowed.	It is allowing in “Password” field..	Fail
TC Reg43	Input any characters less than 6 or more than 20 in “Password” field.	Characters must not be accepted less than 6 or more than 20 in “Password” field.	Characters are not accepted less than 6 or more than 20 in “Password” field.	Fail.
TC Reg44	Input only lower case(wqb) in “Password” field.	It must be allowed to enter lower case in the field.	It is allowing to enter lower case in the field.	Pass.
TC Reg45	Input only upper case(e.g.- AS) in “Password” field.	It must be allowed to enter upper case in the field.	It is allowing to enter upper case in the field.	Pass.
TC Reg46	Input Special characters with alphabetic values(e.g.- asD@) in “Password” field.	It must be to enter Special characters with alphabetic values allowed.	It is allowing to enter Special characters with alphabetic values allowed.	Pass.
TC Reg47	Input any(0-9) numbers (e.g.- 234) in “Password” field.	Numbers(0-9) must be allowed to be entered.	Numbers are allowed.	Pass.

TC Reg48	Input any alphanumeric values(as2S) in “Password” field.	Alphanumeric must be allowed.	Alphanumeric are allowed.	Pass.
TC Reg49	Input any float values(e.g.- 5.7) in “Password” field.	Float values must be allowed to be entered.	Float values are allowed to be entered.	Pass.
TC Reg50	Input any characters within range of(0-n) in “Password” field.	Characters must be accepted within range of(0-n).	Characters are accepted within range of(0-n).	Pass.
TC Reg51	Password No Empty.	A message must appear that “Password not providing”.	A message is displaying that “Password not providing”.	Pass.
TC Reg52	Select button to browse image in(PNG,JPG,BMP,JPEG) “Upload Image” field.	It must select an image of format(PNG,JPG,BMP,JPEG).	It is accepting image of format(PNG,JPG,BMP,JPEG)..	Pass.
TC Reg53	Don't Select button to browse image in “Upload Image” field.	It must display a default image.	It is displaying a default image.	Pass.
TC Reg54	Click on “Add” button without selecting “Image button”.	There is message account added without image.	Image not provided.	Pass.
TC Reg83	Click on “Add” button without selecting “Name”.	It is an error & message must be highlighted.(Please Enter Name).	It is an error & message is highlighted.(Please Enter Name).	Pass.
TC Reg84	Click on “Add” button without selecting “Email id”.	It is an error & message must be highlighted(Email Id is Mandatory).	It is an error & message is highlighted.(Email Id is Mandatory)	Pass.
TC Reg85	Click on “Add” button without selecting “Contact No”.	It is an error & message must be highlighted.(Contact Number is Mandatory)	It is an error & message is highlighted.(Contact Number is Mandatory)	Pass.
TC Reg86	Click on “Add” button without selecting “Description”.	It is an error & message must be highlighted.(Description is Required)	It is an error & message is highlighted.(Description is Required)	Pass.
TC Reg87	Click on “Add” button without selecting “Password”.	It is an error & message must be highlighted.(Password is Mandatory)	It is an error & message is highlighted.(Password is Mandatory)	Pass.
TC Reg88	Click on “Add” button after filling all the information.	Toast displayed that account is added.	Account Added.	Pass.
TC	Click on “Add” button	It must clear all the	It is clearing all the	Pass.

Reg89	for filling information again.	information and reset the page.	information and reset the page.	
User Interface (UI) and Graphical User Interface (GUI) Testing-				
User Interface: Spellings are properly checked and are in proper format.				
Graphical User Interface: Logo,Textboxes,Buttons, and colors are properly arranged.				

MODULE-2 (Add Case FORM)-

TEST SUIT ID	**CREATED BY**	**CREATED ON**
Forensic Expert App	Navjot Singh	1-11-2014
IMPLEMENTED BY: Navjot Singh		IMPLEMENTED ON: 02-11-2014

Pre-Requisites:

Type:
Valid Classes: Select from Auto Textbox.
Valid Range: Selected Index Only.
Invalid Classes: Special characters except: **._ @**. Fractions.
Invalid Range: Editing Selected Case Name.

Name:
Valid Classes: Alphanumeric.
Valid Range: Min-1.
Invalid Range: Less than 1.

Location:
Valid Classes: Alphanumeric.
Valid Range: Min-1.
Invalid Range: Less than 1.

Status:

Valid Classes: Alphanumeric.
Valid Range: Min-1.Status is mandatory.
Invalid Range: Less than 1.

Investigators:
Valid Classes: Select from Auto text either by name or Id.
Valid Range: Min-1.
Invalid Range: Less than 1.

Start-Date:
Valid Classes: Select from Date Time Picker.
Valid Range: Must be Selected.
Invalid Range: Not Selected.

Table: Add Case Test Case

Test Case ID	**Description**	**Expected Result**	**Actual Result**	**Remarks**
TC Case 01	Click on “Add” button without selecting Any Field.	It is an error & Validation message must be highlighted (Invalid Field).	It is displaying a validation error message (Invalid Field)	Pass
TC Case 02	Click on “Add” button without selecting Case Type.	It is an error & Validation message must be highlighted (Select Case Type).	It is displaying a validation error message (Select Case Type.)	Pass
TC Case 03	Click on “Add” button after filling all the information.	“Case successfully added” alert message indicating successful registration must be displayed.	It is displaying message “Case successfully added” .	Pass
TC Case 04	Input any special characters: other than selected with alphabetic values in “Type” field.	Special characters other than selected must not be accepted and it must shown an validation error message.(Select type in correct	It is not allowing characters except. and it is displaying a validation error message.(Select Case type in correct format).	Pass

		format).		
TC Case 05	Input only upper or lower case or numeric character with spaces in “Investigator” field.	Lower or upper case characters with spaces will be accepted & Validation message must be highlighted.	It is allowing lower and upper case.	Pass
TC Case 11	Input Any special characters: with alphabetic values in “Location” field.	It must be allowed to be entered.	It is allowing special characters with alphabetic values.	Pass
TC Case 12	Input only upper case character values in “Location” field.	Upper case will be accepted.	It is allowing upper case characters.	Pass
TC Case 13	Input only lower case character values in “Location” field.	Lower case will be accepted.	It is allowing lower case characters.	Pass
TC Case 14	Display CaseId after Adding a Case.	Toast will be Displayed with CaseId.	Toast will be Displayed with CaseId.	Pass
User Interface (UI) and Graphical User Interface (GUI) Testing-				
User Interface: Spellings are properly checked and are in proper format.				
Graphical User Interface: Logo,Textboxes,buttons,datetime picker and colors are properly arranged.				
Device Compatibility Testing-				
Test Case ID	**Description**	**Expected Result**	**Actual Result**	**Remarks**
TC Case 15	Check whether the Activity is loaded on different Devices.	Activity must be loaded on given Device.	It is loaded on given Device.	Pass.

MODULE-3 (USER LOGIN Activity)-

TEST SUIT ID	**CREATED BY**	**CREATED ON**
Forensic Expert App	Navjot Singh	03-11-2014
IMPLEMENTED BY: Navjot Singh		IMPLEMENTED ON: 04-11-2014

Pre-Requisites:

Login Name:
Valid classes:A-Z,a-z,Numeric(0-9),Alphanumeric.
Valid Range: Min-1.Enter login id.
Valid Classes: Any valid user name.
Invalid Range: Empty.

Password:
Valid Classes:A-Z,a-z,Spaces,Numeric,Alphanumeric values,Float values,Numeric(0-9).
Valid Range: Min-1.Password is mandatory.
Invalid Range: Less than 1.

Test Case ID	**Description**	**Expected Result**	**Actual Result**	**Remarks**
TC Log 01	Input only lower case characters in “Name” field.	Lower case must be accepted.	It is Acceptable, Lower case is accepted.	Pass
TC Log 02	Input only upper case characters in “Name” field.	Upper case must be accepted.	It is Acceptable, Upper case is accepted.	Pass
TC Log 03	Input any special characters with alphabetic values in “Name” field.	Special characters can be accepted.	It is Acceptable, special characters are accepted.	Pass
TC Log 04	Input any(0-9) numeric values in “Name” field.	It must be allowed to enter.	It is Acceptable and it is allowing to enter.	Pass
TC Log 05	Input any float values in “Name” field.	Float values must allowed.	It is Acceptable,	Pass
TC Log 06	Input Fraction values in “Name” field.	Fraction values must be allowed to be entered.	It is Acceptable, Fraction values is accepting.	Pass
TC Log 07	Input any number of	It must be	It is Acceptable and	Pass

	characters within range of(1-n) in “Name” field..	accepted only this range(1-n).	it is allowing to enter.	
TC Log 08	Input only upper case and lower case in “Password” field.	It must allowed lower and upper case in the field.	It is Acceptable and it is allowing to enter.	Pass
TC Log 09	Input only Lowercase an Upper case in “Password” field.	It must allowed lower and upper case in the field.	It is Acceptable and it is allowing to enter.	Pass
TC Log 10	Input only lower or upper case with spaces in “Password” field.	It must not be allowed.	It is Acceptable and Spaces is not accepting and alert is displaying (“Unauthorized Login”)	Pass
TC Log 11	Input Special characters with alphabetic values in “Password” field.	It must be allowed.	It is Acceptable and special characters with alphabetic values in “Password” field entered.	Pass
TC Log 12	Input any(0-9) number or alphanumeric values in “Password” field.	Numbers(0-9) must be allowed to be entered.	It is Acceptable and Numbers(0-9) is allowing to be entered.	Pass
TC Log 13	Input any alphanumeric values in “Password” field.	Alphanumeric must be allowed.	It is Acceptable and Alphanumeric is allowing to be entered.	Pass
TC Log 14	Input any float values in “Password” field.	Float values must be allowed to be entered.	It is Acceptable and Float values is allowing to be entered.	Pass
TC Log 15	Input any fraction values in “Password” field	fraction values must be allowed to be entered.	It is Acceptable and fraction values is allowing to be entered.	Pass
TC Log 16	Input any characters within range of(1-n) in “Password” field.	Characters must be accepted within range of(1-n).	It is Acceptable and Characters be accepted within range of(1-20).	Pass

TC Log 17	Input any characters less than 1 “Password” field.	Input must not be accepted and error message must be display.	It is Acceptable and characters less than 1 “Password” field is not accepting and alert is displaying (“Unauthorized Login”)	Pass
TC Log 18	Click on “Login” button without providing Credentials.	It is an error & message must be highlighted.	It is Acceptable and Click on “Login” button without selecting Name It is an error & validation message is highlighting(“Enter Name”).	Pass
TC Log 19	Click on “Login” button without selecting Password.	It is an error & message must be highlighted.	It is Acceptable and Click on “Login” button without selecting Password It is an error & validation message is highlighting(“Enter Password”).	Pass
TC Log 20	Click on “Login” button after filling all the information.	User must be logged into the system.	It is Acceptable and fraction values is allowing to be entered and User is logging into the system.	Pass

User Interface (UI) and Graphical User Interface (GUI) Testing-

User Interface: Spellings are properly checked and are in proper format.

Graphical User Interface: Logo,Textboxes,Buttons and colors are properly arranged.

Device Compatibility Testing-

Test Case ID	**Description**	**Expected Result**	**Actual Result**	**Remarks**
TC Log21	Check whether the App is working properly on every device.	App must be Installed on given device.	It is installed on given device.	Pass.

MODULE-4 (Add Evidence Activity)-

TEST SUIT ID	CREATED BY	CREATED ON
Forensic Expert App	Navjot Singh	04-11-2014
IMPLEMENTED BY: Navjot Singh		IMPLEMENTED ON: 04-11-2014

Pre-Requisites:

Case Name

Valid Classes: Select from Auto Textbox.

Invalid Classes: Altering the selected case name.

Valid Range: Only Selected Name.

Invalid Range: Less than 1 .

Evidence Type

Valid Classes: Select from Auto Textbox or type your own.

Invalid Classes: only numeric or special characters.

Valid Range: Must Provide.

Invalid Range: Less than 1.

Evidence Name

Valid Classes:A-Z,a-z,Spaces,Special characters,Alphanumeric.

Valid Range:Max 200.

Invalid Range:More than 200.

File

Valid Classes:Only “Image /Video/Audio” file can be Uploaded.

Invalid Classes:Other than “Image/Video/Audio” file .

Valid Range: Select only one at a time either image/video/audio.

Invalid Range try to selecting multiple at a same time.

Table: Add Evidence Test Case

Test Case ID	**Description**	**Expected Result**	**Actual Result**	**Remarks**
TC Evi01	Click on “Add” button without selecting any field.	It is an error & Validation message must be highlighted.(Please Provide Information.)	It is Acceptable(user can not live these field blank and Validation message is highlighting.(Please Provide Information.)	Pass
TC Evi 02	Click on “Add” button without selecting Case	It is an error & Validation message must be highlighted(Please Enter Case Name).	It is Acceptable (Validation message is highlighting(Please Enter Case Name).)	Pass
TC Evi 03	Click on “Add” button without selecting Type .	It is an error & Validation message must be highlighted(Please select type).	It is Acceptable Validation message is highlighting(Please select type).	Pass
TC Evi 04	Click on “Add” button without entering name .	It is an error & Validation message must be highlighted(Please Provide name.).	It is Acceptable Validation message is highlighting(Please Provide name.).	Pass
TC Evi 05	Input Numeric, Alphanumeric in “Name” field.	It must be accepted any numeric and Alphanumeric values.	It is Acceptable.(User can enter any numeric and Alphanumeric values because it is allowed).	Pass
TC Evi 06	Input Other than selected case name in “Case” field..	It must not be accepted any alteration in selected case name.	It is Acceptable.(User can not enter any values because it is not allowed).	Pass
TC Evi 07	Input A-Z,a-z,Spaces,Dot in “Type” field..	A-Z,a-z,Spaces,Dot must be allowed to be entered.	It is Acceptable.(User can enter A-Z,a-z,Spaces,Dot in “Qualification” field).	Pass
TC Evi 08	Input more than 50 characters in “Description” field..	It is accepted more than 50 characters .	It is Acceptable(User can enter more than 50 characters because it is allowed) .	Pass
TC Evi 09	Input characters within (1-50) range in	It is acceptable and user must be enter within (1-	It is Acceptable.(User can not exceeded these	Pass

	“Qualification” field..	50) range.	range and it not accept more than 50 characters because it is not allowed).	
TC Evi 10	Click on “Add” button without selecting “Type”.	It is an error & Validation message must be highlighted((Please enter type).).	It is Acceptable(Validation message is highlighting(Please enter Type).	Pass
TC Evi 11	Input Numeric,Alphanumeric in “UniCasesity”	It must not accept any numeric and Alphanumeric values.	It is Acceptable(It is not allowed to enter through user).	Pass
TC Evi 12	Input Other than A-Z,a-z,Spaces, Dot in “UniCasesity ” field..	It is not accept any numeric and Alphanumeric values..	It is Acceptable(It is not allowed to enter through user).	Pass
TC Evi 13	Input A-Z,a-z,Spaces,Dot in “UniCasesity ” field..	A-Z,a-z,Spaces,Dot must be allowed to be entered.	It is Acceptable(A-Z,a-z,Spaces,Dot is allowed to be entered by user.)	Pass
TC Evi 14	Input more than 50 characters in “Qualification” field..	It must not accepted more than 50 characters .	It is Acceptable(It is not allowed to enter through user).	Pass
TC Evi 15	Input characters within (1-50) range in “Qualification” field..	It must be acceptable and user must be enter it in (1-50) range.	It is Acceptable(It is not allowed to enter through user more than (1-50) range).	Pass
TC Evi 16	Click on “Submit” button without selecting Document Description .	It must be an error & Validation message must be highlighted(Please Describe your Qualification.)..	It is Acceptable Validation message is highlighting(Please Describe your Qualification.).	Pass

TC Evi 17	Input A-Z,a-z,Spaces,Special characters,Alphanumeric in Document Description ..	A-Z,a-z,Spaces,Special characters,Alphanumeric must be allowed to enter.	It is Acceptable and user have permission to enter A-Z,a-z,Spaces,Special characters,Alphanumeric.	Pass
TC Evi 18	Input more than 200 characters in “Description”.	more than 200 characters in “Description” is allowed and more than 200 characters are acceptable.	It is Acceptable(more than 200 characters in “Description” is allowed and more than 200 characters are acceptable.).	Pass
TC Evi 19	Input characters within (1-N) range in “Description”.	characters within (1-N) range in “Document Description” must be allowed.	It is Acceptable(only characters within (1-N) range in “Document Description” are allowed.)	Pass
TC Evi 20	Click on “Add” button without selecting “File”.	It must be accepted .	It is Acceptable.	Pass
TC Evi 21	Upload Other than “Image/Video/Audio” file then Click on “Add” Button.	It must be an error .	It is an Error.	Pass
TC Evi 22	Upload more than 500kb Image file then Click on “Add” Button.	It must be allowed.	It is Acceptable.	Pass

TC Evi 23	Upload more than 500kb image file then Click on “Add” Button.	It must be allowed.	It is Acceptable	Pass
TC Evi 24	Upload Image file Within(1kb-Nkb) the click on “Submit” button.	Image file Within(1kb-Nkb) must be Accepted and user can upload.	It is Acceptable(Image file Within(1kb-Nkb) must be Accepted and user can upload.)	Pass
TC Evi 25	Upload Video file Within(1kb-Nkb) the click on “Add” button.	Video file Within(1kb-Nkb) must be Accepted and user can upload.	It is Acceptable(Video file Within(1kb-Nkb) is Accepted and user can upload.)	Pass
TC Evi 26	Click on “Add” button after selecting all field correctly.	Alert must be displayed. ”Evidence Successfully Added”.	It is Acceptable(Alert will be displayed. ” Evidence Successfully Added”).	Pass

User Interface (UI) and Graphical User Interface (GUI) Testing-

User Interface: Spellings are properly checked and are in proper format.

Graphical User Interface: Logo,Textboxes,Buttons and colors are properly arranged.

Device Compatibility Testing-

TestCase ID	**Description**	**Expected Result**	**Actual Result**	**Remarks**
TC Qua 27	Check whether the App is working properly on every device.	App must be Installed on given device.	It is installed on given device.	Pass.

8. IMPLIMENTATION

❖ Implementation of the Project

Implementation is the process of having system personally checked out and put new equipment into use, trained users, installed the new application and construct and file of data needed to use it. Depending on the size of the organization that will be involved in using the application and the risk associated with its users, system developers may choose to pilot (test) the operation in only one area of the firm, say in one department or with only one or two persons. Regardless of the implementation strategy use, developers strive to ensure that the system's initial use in trouble. Once installed, applications are often used for many years. However, both the organization and the user will change, and the environment will be different over weeks and months. Therefore, the application will undoubtedly have to be maintained, modifications and change will be made to software's, file or procedures to meet emerging user requirement. In the sense, Implementation is an on-going process.

Following activities will be performed for implementation:

- Study Requirements and Design document
- Decide methodology of development
- Select and customize coding standards
- Develop units
- Code
- Self-test will be done before submitting it to Code
- Review
- Code review will be done by peers, by the method of walkthrough
- Unit test by test engineers.
- Integration of components
- Integration testing
- Preparation of Installable
- Release to testing team
- Outputs
- Integrated and tested units
- Peer Review
- Code Review Log
- Defect List

❖ Conversion Plan

The Conversion Plan describes the strategies involved in converting data from an existing system to another hardware or software environment. It is appropriate to re-examine the original system's functional requirements for the condition of the system before conversion to determine if the original requirements are still valid. Various Types of conversion are as follows:-

- Intra language conversion is a conversion between different versions of the same computer language or different versions of a software system, such as a database management system (DBMS), operating system, or local area network (LAN) management system.
- Inter language conversion is the conversion from one computer language to another or from one software system to another.
- Same compiler conversions use the same language and compiler versions. Typically, these conversions are performed to make programs conform to standards, improve program performance, convert to a new system concept, etc. These conversions may require some program redesign and generally require some reprogramming.

❖ Post Implementation and Software Maintenance

Software Maintenance is the last part of the System Development Life Cycle which is actually the implementation of the post-implementation review plan. Software maintenance is a very broad activity that includes error corrections, enhancements of capabilities, deletion of obsolete capabilities, and optimization. Because change is inevitable, mechanism must be developed for evaluation, controlling and making modifications. There are four types of maintenance:

□ □ Corrective Maintenance

□ □ Adaptive Maintenance

□ □ Perfective Maintenance

□ □ Preventive Maintenance

When this system is installed it is used for long period. The average life of a system is 4 to 6 years and maximum used for 10 years. However, this period of use brings with it the need to continually maintain the system, but this system can be modified and new technologies can be used which are prevalent in market at that period of time. Software maintenance requires an accurate maintenance plan to be prepared during the software development. It should specify how users will request modifications or report problems. The budget should include resource and cost estimates. A new decision should be addressed for the developing of every new system feature and its quality objectives. Software maintenance takes more effort than all other phases of software life cycle, but it has not been given as much importance as it deserved. It is an admitted fact that approximately 60 to 70% effort is spent on maintenance phase of software development life cycle.

9. PROJECT LEGACY

CURRENT STATUS OF THE PROJECT

The Designing phase has been almost completed of the project which consists of following modules. The different modules of the project are:

- Investigator Module
- Admin Module

In User Module, Anonymous user can visit all the links like adding, viewing, updating and deletion of the evidences. In the Register User module, new investigator can be added by only admin to get access to app. Can post their requirements and know about current status of the request.

The Investigator cannot register other. For deletion of any evidence it first needs to get access s from admin/expert.

The Administrators module will be protected by user id and password. Ordinary users of the software will not be permitted to enter into this area of the software. This module will be focusing on the security of each evidence of the cases.

❖ FUTURE SCOPE AND ENHANCEMENT

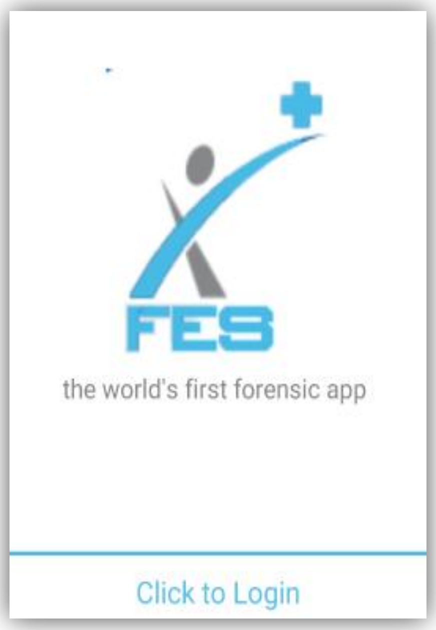
- This website could be created to access all the data available on server side.
- Message Alert Facility Could be included when any new investigator is added
- Analysis module can to create to determine the location of most crimes
- Increase the ratio of cases solved per year
- Web services can be enhanced to collect more information of project.

❖ TECHNICAL SKILL LEARN

- We have learnt XML, Advanced Java.
- Android Mobile operating system development
- In Sqlite database, we have learned about Normalizing Data, Query handling and some new features of Sqlite
- Working with MYSQL for back-end database on web server.
- In managerial Skills, we have learned about works, Group effort and Coordination in group, and the way to work on the Live Projects.

10.User manual: A complete document of the app

Fig24: Login Screen Fig25: Admin Screen



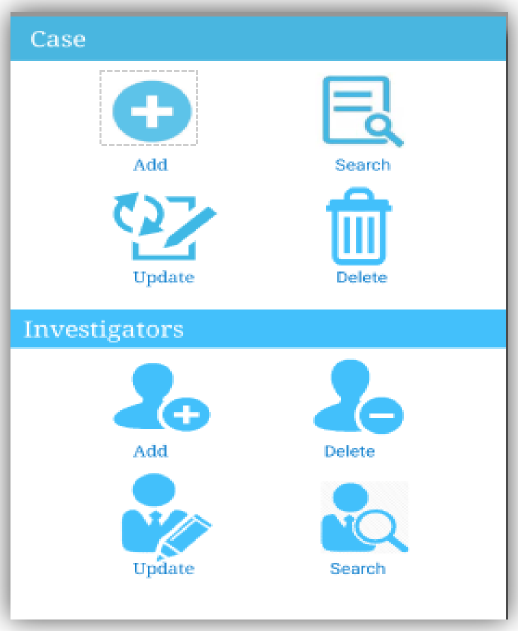


Fig24: Login Screen

Fig25: Admin Screen

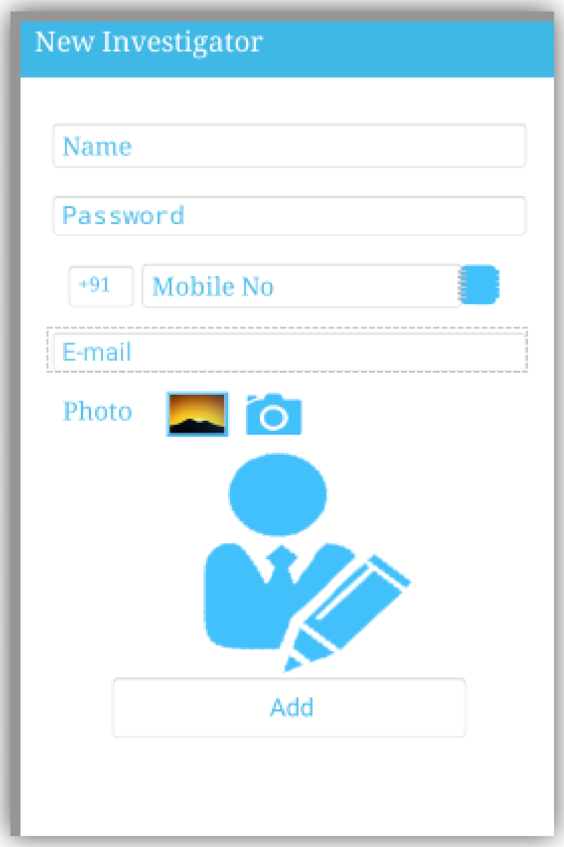


Fig26: Add User

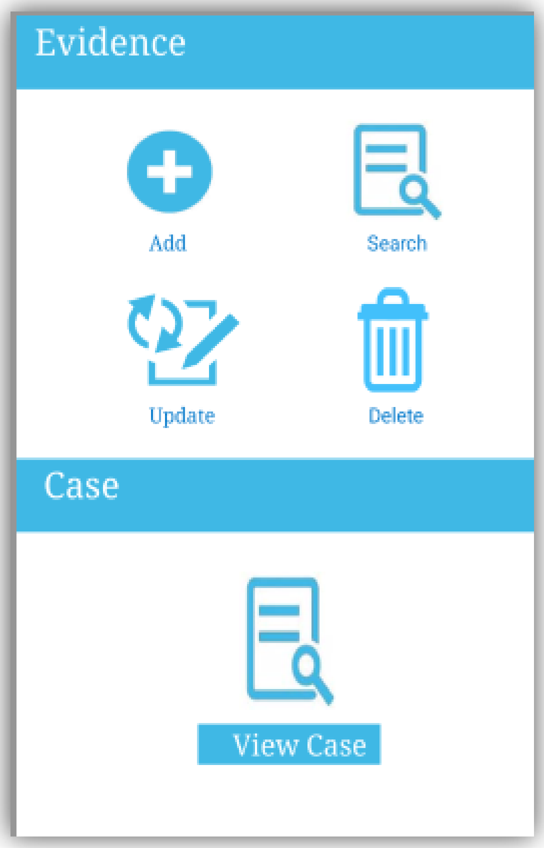


Fig27: User Screen

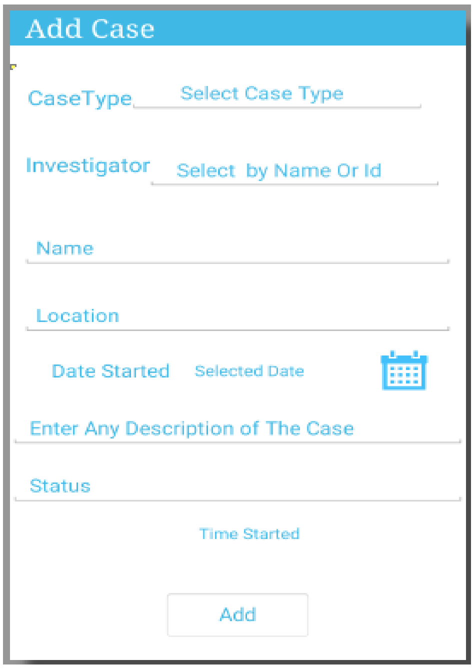


Fig28: Add Case Screen

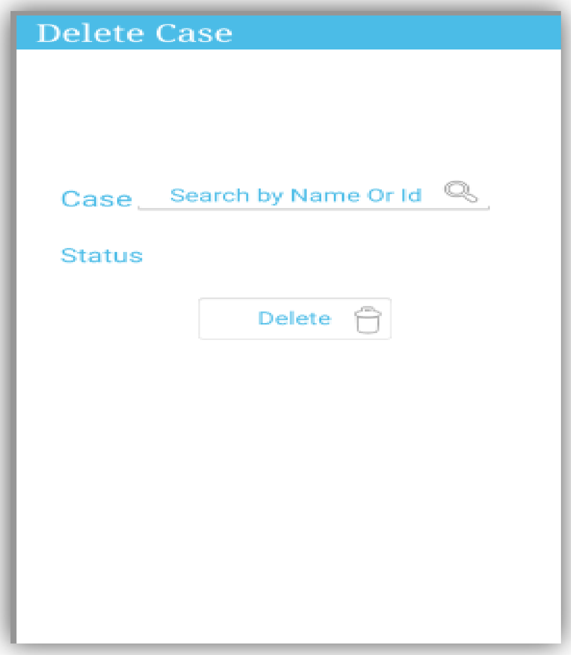


Fig29: Delete Case Screen

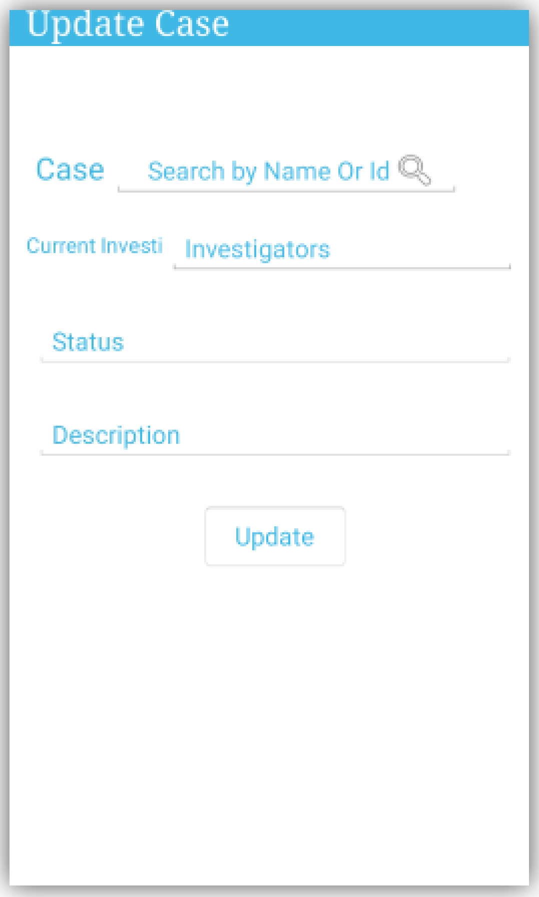


Fig30:Update Case Screen

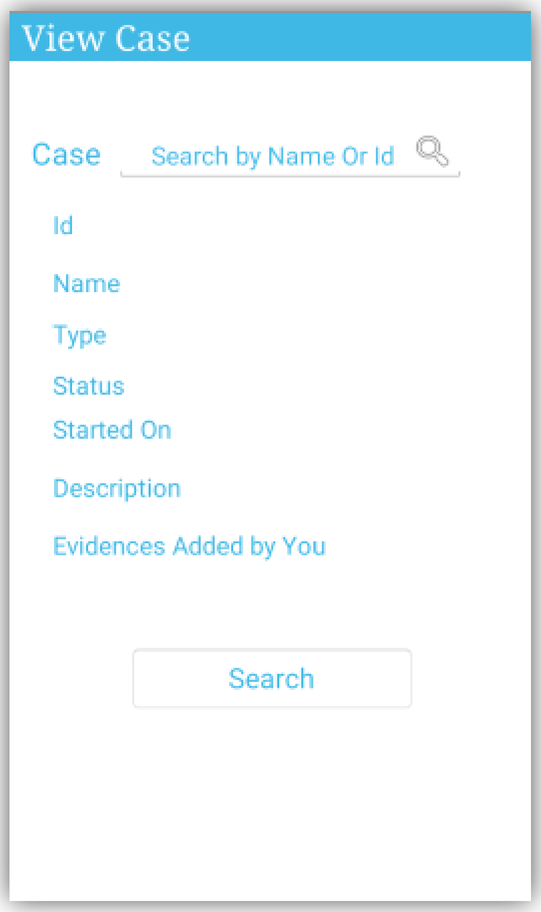


Fig31: Search Case Screen

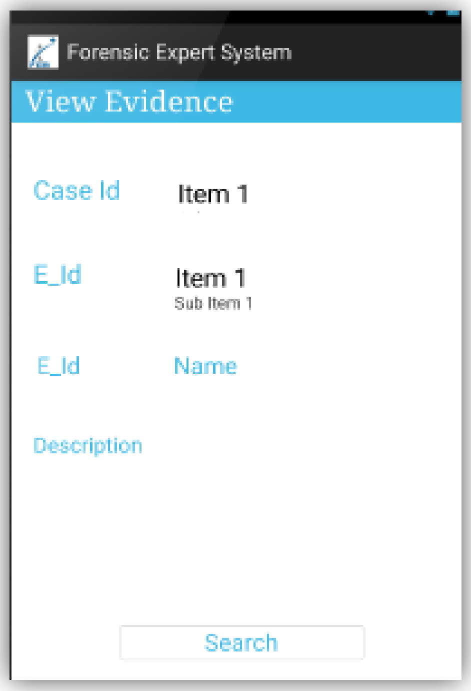


Fig32: View Evidence Screen

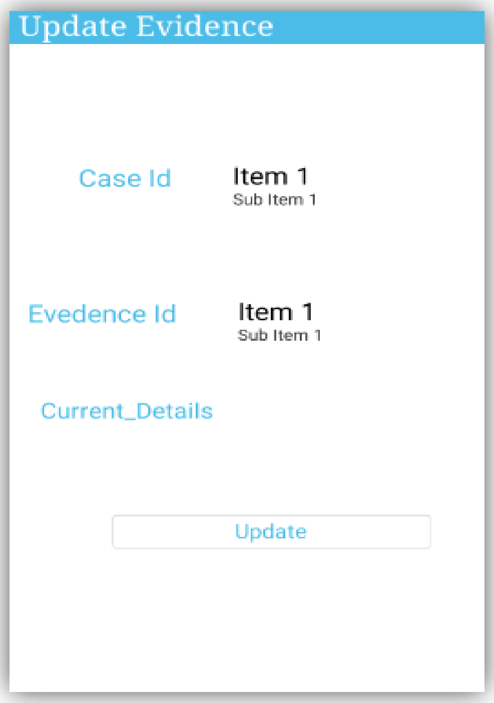
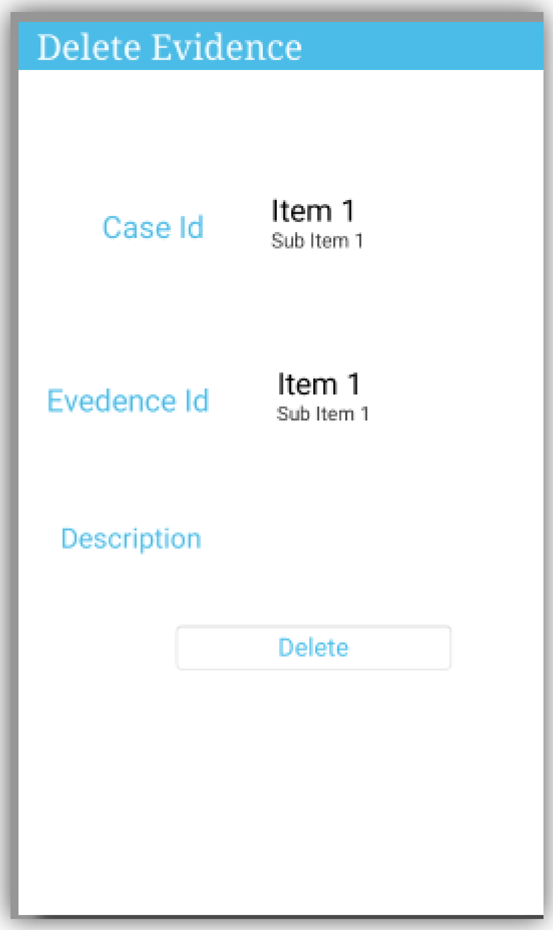


Fig33: Update Evidence Screen



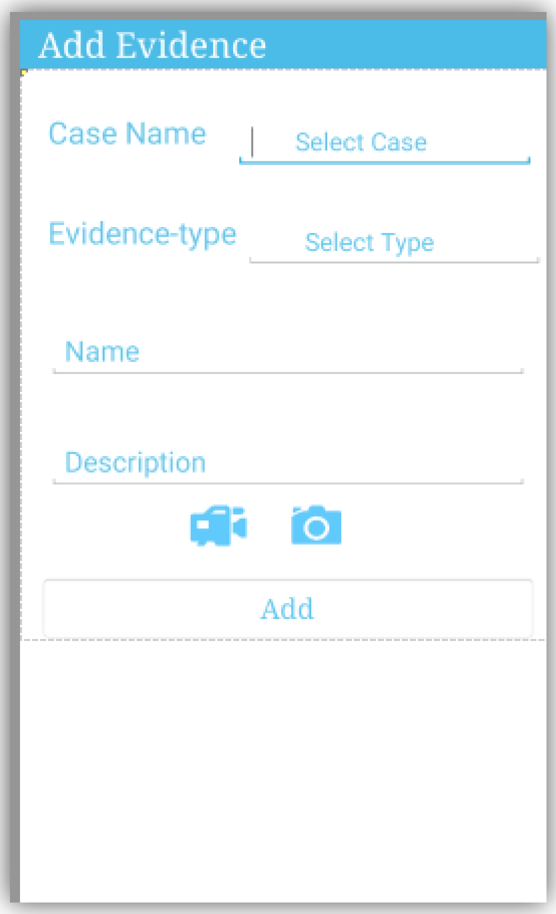


Fig34: Delete Evidence Screen

Fig35: Add Evidence Screen

11.Source code

Activities in the Project

- DBAdapter.java
- Splash.java
- Login.java
- logout.java
- admin.java
- investigator.java
- newuser.java

- deleteuser.java
- searchuser.java
- updateUser.java
- myaccnt.java
- addcase.java
- deletecase.java
- viewcase.java
- modifycase.java
- addevidence.java
- delete_evdnc.java
- upevdnc.java
- changepwd.java
- searchevidence.java

DBAdapter activity code:

```
package com.example.anukoolsrivastav.forensics;
import android.content.Context;
import android.content.ContentValues;
import android.database.sqlite.SQLiteDatabase;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;
import android.graphics.BitmapFactory;
import android.os.NetworkOnMainThreadException;
import android.util.Log;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.DuplicateFormatFlagsException;
import java.util.List;

public class DBAdapter
        //extends SQLiteOpenHelper
{
    public static final String KEY_ROWID = "U_id";
    public static final String KEY_NAME = "Name";
    public static final String KEY_UNAME = "UName";
    public static final String KEY_PASSWORD= "Password";
    public static final String KEY_MOBILENO = "MobileNo";
    public static final String KEY_EMAIL= "Email";
    //Database
    public static final String DATABASE_NAME = "ForensictDB.db";
    //Tabes
    public static final String DATABASE_TABLE = "Cases";
    public static final String TABLE_NAME = "User_img";
    public static final String DATABASE_TABLE1 = "User";
    public static final String DATABASE_TABLE2 = "Victim";
    public static final String DATABASE_TABLE3 = "Evidence";
    public static final int DATABASE_VERSION =1 ;

  private DatabaseHelper DBHelper;
  private SQLiteDatabase db;
  public DBAdapter(Context ctx)
  {
     this.context = ctx;
    DBHelper = new DatabaseHelper(context);
 }
       private static class DatabaseHelper extends SQLiteOpenHelper
                  {
        DatabaseHelper(Context context)
                    {
                        super(context, DATABASE_NAME, null, DATABASE_VERSION);
                    }

        @Override
        public void onCreate(SQLiteDatabase db)
             {

            db.execSQL("CREATE TABLE IF NOT EXISTS User(U_id integer primary key autoincrement,Name text not null,Password text not null,MobileNo text not null,Email text not null)");
            db.execSQL("CREATE TABLE  IF NOT EXISTS Victim(V_id integer primary key autoincrement,Name text not null,Ssn integer not null,MobileNo text not null,Email text not null)");
            db.execSQL("CREATE TABLE  IF NOT EXISTS Cases(C_id integer primary key
autoincrement,name text not null,type text not null,location text not null,status text not null,start
text not null,Addtime text not null,Des text not null)");
            db.execSQL("CREATE TABLE  IF NOT EXISTS Evidence(E_id integer primary key
autoincrement,C_id integer not null,U_id text not null,Type text not null,Detail text not
null,Image text ,time not null,date text not null)");
            db.execSQL("CREATE TABLE  IF NOT EXISTS User_img(U_id integer primary
key,UName text,Image text)");
            db.execSQL("CREATE TABLE  IF NOT EXISTS CaseUid(C_id integer not null,U_id
integer not null)");
                 ContentValues initialValues = new ContentValues();
                 initialValues.put(KEY_ROWID,"1");
                 initialValues.put(KEY_NAME,"admin");
                 initialValues.put(KEY_PASSWORD,"admin");
                 initialValues.put(KEY_MOBILENO,"9872832353");
                 initialValues.put(KEY_EMAIL,"navsingh191@gmail.com");
                 db.insert(DATABASE_TABLE1, null, initialValues);
                 //db.insert()

                  }
        @Override
        public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
            //db.execSQL("DROP TABLE IF EXISTS " + DATABASE_TABLE1);
           // db.execSQL("DROP TABLE IF EXISTS " + DATABASE_C);
            //db.execSQL("DROP TABLE IF EXISTS " + DATABASE_C1);
      //          db.execSQL("INSERT INTO User (Name,Password,MobileNo,Email) VALUES
(Admin,Admin,9872832353,abc)");
           // onCreate(db);

        }
    }
  public DBAdapter open() throws SQLiteException
   {
       db = DBHelper.getWritableDatabase();
      return this;
  }
 public void close()

  {
      DBHelper.close();
  }


 public  void  insertRecord(String name,String pass,String no,String mail)
 {
     //String id1=id;
     String n=name;
     String n2=pass;
     String n3=no;
     String n4=mail;
 ContentValues initialValues = new ContentValues();
   initialValues.put(KEY_NAME,name);
    initialValues.put(KEY_PASSWORD,pass);
     initialValues.put(KEY_MOBILENO,no);
    initialValues.put(KEY_EMAIL,mail);
 db.insert(DATABASE_TABLE1, null, initialValues);


 }
   public  long  insertRecord1(String name,String pass,String no,String mail)
   {
       ContentValues initialValues = new ContentValues();
       initialValues.put(KEY_NAME, name);
       initialValues.put(KEY_PASSWORD, pass);
       initialValues.put(KEY_MOBILENO, no);
       initialValues.put(KEY_EMAIL, mail);
       return db.insert(DATABASE_TABLE1, null, initialValues);
   }


    //Insert Case
    public   long   insertcase(String  name,String  type,String  loc,String  status,String  start,String
time,String dec)
    {
        ContentValues initialValues = new ContentValues();
      //  initialValues.put("U_id",u_id);
        initialValues.put("name",name);
        initialValues.put("type",type);
        initialValues.put("location",loc);
        initialValues.put("status",status);
        initialValues.put("start",start);
        initialValues.put("Addtime",time);
        initialValues.put("Des",dec);
        return db.insert(DATABASE_TABLE, null, initialValues);
    }
public Cursor getRecord12(String rowid)throws SQLiteException
  {
      String  where=(KEY_ROWID + "='"+ rowid+"'")+" OR "+(KEY_NAME + "='"+ rowid+"'"
);
```

```
    Cursor m=db.query(true,DATABASE_TABLE1,new
String[]{KEY_ROWID,KEY_NAME,KEY_PASSWORD,KEY_MOBILENO,KEY_EMAIL},
where,null,null,null,null,null);
    if(m!=null)
      {
        m.moveToFirst();
      }
     return m;
    }
    public String getRecordEmail(String email)throws SQLiteException
    {
        String mail=null;
        Log.d("getRecordEmail","21");
        Cursor m=db.query(true,DATABASE_TABLE1,new
String[]{KEY_ROWID,KEY_NAME,KEY_PASSWORD,KEY_MOBILENO,KEY_EMAIL},
KEY_EMAIL+"='"+email+"'",null,null,null,null,null);
        Log.d("getRecordEmail","22");
        Log.d("Cursor is ",m.toString());
        if(m!=null)
        {
            Log.d("getRecordEmail","23");
            m.moveToFirst();
            Log.d("getRecordEmail","24");
            Log.d("Column Index is ",m.getColumnIndex("Email")+"");
            int index=m.getColumnIndex("Email");
            Log.d("Total Records ",m.getCount()+"");
            if(m.getCount()==0)
            {mail="";}
            else{
            mail=m.getString(index);}
            Log.d("getRecordEmail","25");
            Log.d("getRecordEmail",mail);
        }
        return mail;
    }
    public String getRecordEmailUP(String email,String ddi)throws SQLiteException
    {
        String mail=null;
        String  where=(KEY_ROWID + "!='"+ ddi+"'" )+" AND "+(KEY_NAME + "!='"+ ddi+"'"
)+" AND "+(KEY_EMAIL + "='"+email+"'" );
        Log.d("getRecordEmail","21");
        Cursor m=db.query(true,DATABASE_TABLE1,new
String[]{KEY_ROWID,KEY_NAME,KEY_PASSWORD,KEY_MOBILENO,KEY_EMAIL},
where,null,null,null,null,null);
        Log.d("getRecordEmail","22");
        Log.d("Cursor is ",m.toString());
        if(m!=null)
        {
            Log.d("getRecordEmail","23");
            m.moveToFirst();
            Log.d("getRecordEmail","24");
            Log.d("Column Index is ",m.getColumnIndex("Email")+"");
            int index=m.getColumnIndex("Email");
            Log.d("Total Records ",m.getCount()+"");
            if(m.getCount()==0)
            {mail="";}
            else{
               mail=m.getString(index);}
            Log.d("getRecordEmail","25");
            Log.d("getRecordEmail",mail);
        }
        return mail;
    }
    public Cursor getRecordEmail1(String email)throws SQLiteException
    {
        Cursor                                             m=db.query(true,DATABASE_TABLE1,new
String[]{KEY_ROWID,KEY_NAME,KEY_PASSWORD,KEY_MOBILENO,KEY_EMAIL},
KEY_EMAIL+"='"+email+"'",null,null,null,null,null);
        if(m!=null)
        {
            m.moveToFirst();

               }
        return m;
        }
    public String getRecordMobile(String mobile)throws SQLiteException
    {
        String mobile1=null;
        Cursor                                            m1=db.query(true,DATABASE_TABLE1,new
String[]{KEY_ROWID,KEY_NAME,KEY_PASSWORD,KEY_MOBILENO,KEY_EMAIL},
KEY_MOBILENO+"='"+mobile+"'",null,null,null,null,null);
        if(m1!=null)
        {
            m1.moveToFirst();
            Log.d("getRecordEmail","26");
            Log.d("Column Index is ",m1.getColumnIndex("MobileNo")+"");
            int index=m1.getColumnIndex("MobileNo");
            Log.d("Total Records ",m1.getCount()+"");
            if(m1.getCount()==0)
            {
               mobile1="";
            }
            else{
            mobile1=m1.getString(index);}
            Log.d("getRecordMobile","25");
            Log.d("getRecordMobile",mobile1);
        }
        return mobile1;
    }

//    //delete record by id
    public boolean deleteRecord(String rowid) {
        boolean b=false;
        //  return db.delete("User", "U_id" + "=" + rowid, null) > 0;
        String where=(KEY_ROWID + "='"+ rowid+"'")+" OR "+(KEY_NAME + "='"+
rowid+"'" );
        // Cursor m1 = db.query(true, DATABASE_TABLE1, new String[]{KEY_ROWID,
KEY_NAME, KEY_PASSWORD, KEY_MOBILENO, KEY_EMAIL}, KEY_ROWID + "='"
+ rowid + "'", null, null, null, null, null);
        Cursor m1 = db.query(true, DATABASE_TABLE1, new String[]{KEY_ROWID,
KEY_NAME, KEY_PASSWORD, KEY_MOBILENO, KEY_EMAIL}, where, null, null, null,
null, null);

        if (m1 != null) {
            m1.moveToFirst();
            Log.d("getRecordEmail", "26");
            Log.d("Column Index is ", m1.getColumnIndex("MobileNo") + "");
            //   int index=m1.getColumnIndex("MobileNo");
            Log.d("Total Records ", m1.getCount() + "");
            if (m1.getCount() == 0) {
                // mobile1="";
                return b;
            }
            else {
                //mobile1=m1.getString(index);}
                //  Log.d("getRecordMobile","25");
                //  Log.d("getRecordMobile",mobile1);
                String where2=(KEY_ROWID + "='"+ rowid+"'")+" OR "+(KEY_NAME + "='"+
rowid+"'" );
                 // b= db.delete("User", "U_id" + "=" + rowid, null) > 0;
                b=db.delete("User", where2, null) > 0;
            }
        }
        return  b;
    }   //update
```

```
/* public boolean updateRecord(long rowid)
 {
    return db.update("User", "set Name = Nav","where u_id=1",null) > 0;
 }*/
public int  updateRecord(String id,String Name,String MobileNo,String Email)
 {
    int f=0;
    //  return db.delete("User", "U_id" + "=" + rowid, null) > 0;
    String  where=(KEY_ROWID + "='"+ id+"'")+" OR "+(KEY_NAME + "='"+ id+"'" );
    //  Cursor  m1  =  db.query(true,  DATABASE_TABLE1,  new  String[]{KEY_ROWID,
KEY_NAME, KEY_PASSWORD, KEY_MOBILENO, KEY_EMAIL}, KEY_ROWID + "='"
+ rowid + "'", null, null, null, null, null);
    Cursor  m1  =  db.query(true,  DATABASE_TABLE1,  new  String[]{KEY_ROWID,
KEY_NAME, KEY_PASSWORD, KEY_MOBILENO, KEY_EMAIL}, where, null, null, null,
null, null);
    if (m1 != null) {
       m1.moveToFirst();
       Log.d("getRecordEmail", "26");
       Log.d("Column Index is ", m1.getColumnIndex("MobileNo") + "");
       //   int index=m1.getColumnIndex("MobileNo");
       Log.d("Total Records ", m1.getCount() + "");
       if (m1.getCount() == 0) {
          // mobile1="";
          return f;
       }
       else {
          //mobile1=m1.getString(index);}
          //  Log.d("getRecordMobile","25");
          //  Log.d("getRecordMobile",mobile1);
          String  where2=(KEY_ROWID + "='"+ id+"'")+" OR "+(KEY_NAME + "='"+ id+"'"
);
          // b= db.delete("User", "U_id" + "=" + rowid, null) > 0;
          ContentValues initialValues = new ContentValues();
          initialValues.put("Name",Name);
          initialValues.put("MobileNo",MobileNo);
          initialValues.put("Email",Email);
          //if (c!=null) {
          f= db.update("User", initialValues,where2, null);

        //  b=db.delete("User", where2, null) > 0;
       }
    }
    //Cursor c = getRecord(2);
      return  f;
```

```
       //}
      // else
         //  return  0;

   }
```

LOGIN activity:

```
package com.example.anukoolsrivastav.forensics;
import android.animation.Animator;
import android.animation.AnimatorListenerAdapter;
import android.annotation.TargetApi;
import android.app.Activity;
import android.app.LoaderManager.LoaderCallbacks;
import android.content.ContentResolver;
import android.content.Context;
import android.content.CursorLoader;
import android.content.Intent;
import android.content.Loader;
import android.database.Cursor;
import android.graphics.Typeface;
import android.net.Uri;
import android.net.wifi.WifiManager;
import android.os.AsyncTask;
import android.os.Build.VERSION;
import android.os.Build;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.text.TextUtils;
import android.util.Log;
import android.view.KeyEvent;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnClickListener;
@TargetApi(Build.VERSION_CODES.HONEYCOMB)                              //implements
LoaderCallbacks<Cursor>
public class Login extends Activity {
   private static final String SOAP_ACTION = "http://tempuri.org/login";
   private static final String METHOD_NAME = "login";
   private static final String NAMESPACE = "http://tempuri.org/";
   private static final String URL = "http://anukool.somee.com/Service.asmx";
   /**
```

```
     * A dummy authentication store containing known user names and passwords.
     * TODO: remove after connecting to a real authentication system.
     */
    private EditText e1;
    private EditText e2;
    ProgressBar webservicePG;
    String username, password;
    DBAdapter db = new DBAdapter(this);

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        webservicePG = (ProgressBar) findViewById(R.id.progressBar1);



        TextView txt = (TextView) findViewById(R.id.textView);
        Typeface font = Typeface.createFromAsset(getAssets(), "FontleroyBrownNF.ttf");
        txt.setTypeface(font);
        e1 = (EditText) findViewById(R.id.user);
        e2 = (EditText) findViewById(R.id.pass);
        Button login = (Button) findViewById(R.id.log_in);

        final WifiManager wifiManager = (WifiManager)
this.getSystemService(Context.WIFI_SERVICE);
        // Button pwd = (Button) findViewById(R.id.button);
        login.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View view) {

                localDb(); //Call to local db

                if(wifiManager.isWifiEnabled())
                {
                    new Asynccs().execute();
                    Toast.makeText(getBaseContext(), "Yup", Toast.LENGTH_LONG).show();
                }
                else
                {
                    Toast.makeText(getBaseContext(), "Wi-fi Not available",
Toast.LENGTH_LONG).show();
                }
                  }       });    }
    public void localDb()
    {
                db.open();
                 Cursor m=db.getRecord12("1");
                m.moveToFirst();

                String name=m.getString(m.getColumnIndex("Name"));

                m.close();
                   e1=(EditText)findViewById(R.id.user);
                   e2=(EditText)findViewById(R.id.pass);
                   String userName=e1.getText().toString();
                   String password=e2.getText().toString();
                  // fetch the Password form database for respective user name
                  String storedPassword=db.getSinlgeEntry(userName);
                  String id=db.getid12(userName);
                 // check if the Stored password matches with  Password entered by user
                 if(password.equals(storedPassword))
                       {
                    webservicePG.setVisibility(View.INVISIBLE);
                    if (userName.equals(name))
                       {
                        Intent h = new Intent(Login.this, admin.class);
                        h.putExtra("UserName",name);
                        startActivity(h);
                        Toast.makeText(Login.this, "Congrats Admin: Login Successfull",
Toast.LENGTH_LONG).show();
                        finish();
                       }
                    else
                     if(password.equals(storedPassword))
                    {
                        Toast.makeText(Login.this, "Congrats: Login Successfull",
Toast.LENGTH_LONG).show();
                        Intent home = new Intent(Login.this, investigator.class);
                        //home.putExtras(lName);
                        startActivity(home);
                        finish();
                    }                  }
                else
                {
                    Toast.makeText(Login.this, "User Name or Password does not match",
Toast.LENGTH_LONG).show();
```

```
            }          }

    public class Asynccs extends AsyncTask<String, String, String> {
        @Override
        protected String doInBackground(String... strings) {
            String h;
            try {
              SoapObject Request1 = new SoapObject(NAMESPACE, METHOD_NAME);
                Request1.addProperty("name", e1.getText().toString());
                Request1.addProperty("pwd", e2.getText().toString());
                //texttxt.setText(logintxt.getText().toString());
                SoapSerializationEnvelope envelope = new
SoapSerializationEnvelope(SoapEnvelope.VER11);
                envelope.dotNet = true;
                envelope.setOutputSoapObject(Request1);
                HttpTransportSE androidHttpTransport = new HttpTransportSE(URL);
                try {
                    androidHttpTransport.call(SOAP_ACTION, envelope);
                    final SoapPrimitive result = (SoapPrimitive) envelope.getResponse();
                    Login.this.runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            //String resultData=result.getProperty(0).toString();
                            //test.setText( result.toString());
                            //
Toast.makeText(getBaseContext(),result.toString(),Toast.LENGTH_LONG).show();
                            //  Log.d("hi","login done");
                            if (result.toString().equals("wel")) {
                                Log.d("hi", "login done");
                                Toast.makeText(getBaseContext(), " login successful",
Toast.LENGTH_LONG).show();
                            }

                        }
                    });

                } catch (Exception e) {
                    e.printStackTrace();

                }
            } catch (Exception er) {

                Toast.makeText(getApplicationContext(), "Try Again",
Toast.LENGTH_LONG).show();
```

```
            }

            return null;
        }
    }
```

NewUser Activity:

```
package com.example.anukoolsrivastav.forensics;

import android.app.Activity;
import android.content.ContentValues;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.net.Uri;
import java.util.Date;

/**
 * Created by Anukool Srivastav on 26-09-2014.
 */
public class newuser extends Activity
{
    String err;
    File pic;
    String DB_NAME = DBAdapter.DATABASE_NAME;
    String a,b,c,d,number;
    String imageFileName;
  int img=0;           // Environment.getExternalStorageDirectory() + "/test.db";
    String TABLE_NAME = "User_Image";
      TextView tv0,textView;
      EditText tv11,tv2,tv3,tv4;
    OutputStream imagefile=null;
      Button add,gal1,cam1,con;
    private String selectedImagePath;
    private  static final int PICK_CONTACT=1;
    int a1=1;
    ImageView first;
    private static final int SELECT_PICTURE = 1;
     @Override
  protected void onCreate(Bundle savedInstanceState)
   {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.add_user);
     gal1=(Button)findViewById(R.id.gal);
     cam1=(Button)findViewById(R.id.cam);
     con=(Button)findViewById(R.id.con);
     tv3=(EditText) findViewById(R.id.tv3);
     gal1.setEnabled(true);
     //pic = getDir("User_Images",Context.MODE_PRIVATE);
     first=(ImageView)findViewById(R.id.userimage);
     gal1.setOnClickListener(new View.OnClickListener() {
       @Override
       public void onClick(View v) {
         a1=2;
         Intent intent = new Intent();
         intent.setType("image/*");
         intent.setAction(Intent.ACTION_GET_CONTENT);
         startActivityForResult(
             Intent.createChooser(intent, "Select Picture"),SELECT_PICTURE);
       }
     });
     con.setOnClickListener(new View.OnClickListener() {
       @Override
       public void onClick(View v) {
         Intent intent = new Intent(Intent.ACTION_PICK,
ContactsContract.Contacts.CONTENT_URI);
         intent.setType(ContactsContract.CommonDataKinds.Phone.CONTENT_TYPE);
         a1=3;
         startActivityForResult(intent,PICK_CONTACT );
       }
     });
     cam1.setOnClickListener(new View.OnClickListener() {
       @Override
       public void onClick(View v) {
         a1=1;
         Intent intent=new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
         startActivityForResult(intent,0);
       }
     });

    // EditText etcity=   ((EditText) findViewById(R.id.city));
   // tv11.setFilters(new InputFilter[]{filter(tv11,get)});
```

```
          }

      public void onActivityResult(int requestCode, int resultCode, Intent data) {
         if (a1 == 2)
         {
            if (resultCode == RESULT_OK)
            {
               if (requestCode == SELECT_PICTURE)
               {
                  Uri selectedImageUri = data.getData();
                  selectedImagePath = getPath(selectedImageUri);
                  System.out.println("Image Path : " + selectedImagePath);
                  first.setVisibility(View.VISIBLE);
                  // image1.setImageURI(selectedImageUri);


                  int a1 = 200, b = 200;
                  Bitmap bitmap = BitmapFactory.decodeFile(selectedImagePath);
                  try {
                     pic = getDir("User_Images",Context.MODE_PRIVATE);
                     String                    timeStamp                    =                    new
   SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
                     imageFileName = "JPEG_" + timeStamp+".jpg";
                     imagefile = new FileOutputStream(pic+"/"+imageFileName);
                    // img++;
                     }
                  catch(Exception e)
                  {

                  }
                  bitmap.compress(Bitmap.CompressFormat.JPEG, 80, imagefile);
                  bitmap = Bitmap.createScaledBitmap(bitmap, a1, b, true);
                  first.setImageBitmap(bitmap);
                  //String nmm=first.get

   Toast.makeText(getApplication(),"ok"+imageFileName,Toast.LENGTH_LONG).show();
                  img=1;
               }

            }
         }
         if (a1 == 3) {
            switch (requestCode) {
```

```
        case (PICK_CONTACT):
          if (resultCode == RESULT_OK) {
            // Get the URI that points to the selected contact
            Uri contactUri = data.getData();
            // We only need the NUMBER column, because there will be only one row in the
result
            String[] projection = {
                ContactsContract.CommonDataKinds.Phone.NUMBER};
            Cursor cursor = getContentResolver()
                .query(contactUri, projection, null, null, null);
            cursor.moveToFirst();

            // Retrieve the phone number from the NUMBER column
            int column =
cursor.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER);
            number = cursor.getString(column);
            Toast.makeText(this, "contact info : " + number,
Toast.LENGTH_LONG).show();
            //String[] ary = number.toString().split("");
            //Toast.makeText(this, "contact info : " + ary[0],
Toast.LENGTH_LONG).show();
            tv3.setText("");
            char[] stringArray;
            //convert string into array using toCharArray() method of string class
            stringArray = number.toCharArray();
            String as = "";
            if (stringArray[0] == '0') {
              for (int i = 1; i < stringArray.length; i++) {
                // tv3.append(stringArray[i]);
                as += stringArray[i];
              }
              //  }
              Toast.makeText(this, "contact info : " + as, Toast.LENGTH_LONG).show();
              tv3.setText(as);
              // Do something with the phone number...
            } else if (stringArray[0] == '9' && stringArray[1] == '1') {
              for (int i = 2; i < stringArray.length; i++) {
                // tv3.append(stringArray[i]);
                as += stringArray[i];
              }
              //  }
              Toast.makeText(this, "contact info : " + as, Toast.LENGTH_LONG).show();
              tv3.setText(as);
```

```
                    } else if (stringArray[0] == '+' && stringArray[1] == '9' && stringArray[2] ==
'1') {
                        for (int i = 3; i < stringArray.length; i++) {
                            // tv3.append(stringArray[i]);
                            as += stringArray[i];
                        }
                        //  }
                        Toast.makeText(this, "contact info : " + as, Toast.LENGTH_LONG).show();
                        tv3.setText(as);
                    } else {

                        Toast.makeText(this, "contact info : " + number,
Toast.LENGTH_LONG).show();
                        tv3.setText(number);
                    }

                }       }     }
            if (a1 == 1)
            {
                super.onActivityResult(requestCode, resultCode, data);
                if(resultCode==RESULT_OK)
                {

                    int a1 = 200, b = 200;
                    //Bitmap bm=(Bitmap) data.getExtras().get("data");
                    //      photoImage.setImageBitmap(bm);
                    Uri selectedImageUri = data.getData();
                    selectedImagePath = getPath(selectedImageUri);
                    Toast.makeText(this, "Image saved successfully
"+resultCode+requestCode+selectedImagePath, Toast.LENGTH_LONG).show();
                    System.out.println("Image Path : " + selectedImagePath);
                    //int a1 = 200, b = 200;
                    Bitmap bitmap = BitmapFactory.decodeFile(selectedImagePath);
                    try {
                        pic = getDir("User_Images",Context.MODE_PRIVATE);
                        String timeStamp = new
SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
                        imageFileName = "JPEG_" + timeStamp+".jpg";
                        imagefile = new FileOutputStream(pic+"/"+imageFileName);
                        Toast.makeText(this, "Image saved successfully "+pic+"/"+imageFileName,
Toast.LENGTH_LONG).show();

                        // img++;
                    }
                    catch(Exception e)
                    {

                    }
                    bitmap.compress(Bitmap.CompressFormat.JPEG, 40, imagefile);
                   // bitmap.compress(Bitmap.CompressFormat.JPEG, 40, imagefile);
                    bitmap = Bitmap.createScaledBitmap(bitmap, a1, b, true);
                    first.setImageBitmap(bitmap);
                    img=1;
                }
        }    }
      public String getPath(Uri uri) {
        String[] projection = { MediaStore.Images.Media.DATA };
        Cursor cursor = managedQuery(uri, projection, null, null, null);
        int column_index = cursor
                .getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        return cursor.getString(column_index);
    }
    public boolean onCreateOptionsMenu(Menu menu)
    {
        // Inflate the menu; this adds items to the action bar if it is present.
        //final TextView tv=(TextView)findViewById(R.id.tv1);
        getMenuInflater().inflate(R.menu.admin, menu);
        menu.add(0, Menu.FIRST,  Menu.NONE,"My Account").setIcon(R.drawable.user);
        SubMenu sub1=menu.addSubMenu(0,Menu.FIRST+1,Menu.NONE,"Change Password");
        SubMenu sub3=menu.addSubMenu(0,Menu.FIRST+2,Menu.NONE,"Logout");
        int positionOfMenuItem0 = 0; // or whatever...
        MenuItem item = menu.getItem(positionOfMenuItem0);
        SpannableString s = new SpannableString("My Account");
        s.setSpan(new ForegroundColorSpan(Color.argb(255, 67, 192, 251)), 0, s.length(), 0);
        item.setTitle(s);
        int positionOfMenuItem1 = 1; // or whatever...
        MenuItem item1 = menu.getItem(positionOfMenuItem1);
        SpannableString s1 = new SpannableString("Change Password");
        s1.setSpan(new ForegroundColorSpan(Color.argb(255,67,192,251)), 0, s1.length(), 0);
        item1.setTitle(s1);
        int positionOfMenuItem2 = 2; // or whatever...
        MenuItem item2 = menu.getItem(positionOfMenuItem2);
        SpannableString s2 = new SpannableString("LogOut");
        s2.setSpan(new ForegroundColorSpan(Color.argb(255,67,192,251)), 0, s2.length(), 0);
        item2.setTitle(s2);
        //tv.setText("Six Clicked");
        return true;
    }
```

Add Case Activity:

```
package com.example.anukoolsrivastav.forensics;

import android.app.DatePickerDialog;
import android.app.Dialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import org.ksoap2.SoapEnvelope;
import org.ksoap2.serialization.SoapObject;
import org.ksoap2.serialization.SoapPrimitive;
import org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;

/**
 * Created by Anukool Srivastav on 24-09-2014.
 */
public class addcase extends ActionBarActivity implements OnClickListener {
    ArrayList<String> items=new ArrayList<String>();
   // ArrayList<String> items1=new ArrayList<String>();
   private static final String SOAP_ACTION = "http://tempuri.org/myCases";
    private static final String METHOD_NAME = "myCases";
    private static final String NAMESPACE = "http://tempuri.org/";
    private static final String URL = "http://anukool.somee.com/Service.asmx";
    Spinner id,invstr1;
     private ImageButton ib;
     private Calendar cal;
     private int day;
     private int month;
     private int year;
     private TextView tv1;
    String[] toArr,toArr1,uniques;
    String a="",b="";
    EditText status,loc,dec,tv11,name;
    ArrayAdapter<String> dataAdapter;
    List<String> labels;
    AutoCompleteTextView actv;
    MultiAutoCompleteTextView actv1;
    Button btn;
    TextView ctp;
    String ctype,cstype;
    String id1;

    TextView txtdate1;

    DBAdapter db=new DBAdapter(this);
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        final WifiManager wifiManager = (WifiManager) this.getSystemService(Context.WIFI_SERVICE);
        super.onCreate(savedInstanceState);
        setContentView(R.layout.add_case);
        ib = (ImageButton) findViewById(R.id.imgbtn);
        cal = Calendar.getInstance();
        day = cal.get(Calendar.DAY_OF_MONTH);
        month = cal.get(Calendar.MONTH);
        ctp = (TextView) findViewById(R.id.ctp);
        year = cal.get(Calendar.YEAR);
        tv1 = (TextView) findViewById(R.id.textView);
        btn = (Button) findViewById(R.id.button);
        ib.setOnClickListener(this);
        items.add("Actual / grievous bodily harm, wounding & other assaults");
        items.add("Armed robbery");
        items.add("Arson");
        items.add("Burglary, robbery & theft");
        items.add("Road traffic accidents");

        final ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, R.layout.spinner, items);
        actv = (AutoCompleteTextView) findViewById(R.id.auto2);
        actv.setAdapter(adapter);
        db.open();
        labels = db.getTableValues12();
        dataAdapter = new ArrayAdapter<String>(this, R.layout.spinner, labels);
        actv1 = (MultiAutoCompleteTextView) findViewById(R.id.auto1);
        actv1.setTokenizer(new MultiAutoCompleteTextView.CommaTokenizer());
        actv1.setAdapter(dataAdapter);
        name=(EditText) findViewById(R.id.name);
        actv1.setOnClickListener(new OnClickListener()
        {
            @Override
            public void onClick(View v) {
                actv1.addTextChangedListener(new TextWatcher() {
                    @Override
                    public void beforeTextChanged(CharSequence s, int start, int count, int after) {
                    }
                    @Override
                    public void onTextChanged(CharSequence s, int start, int before, int count)
                    {
                        a="";
                        b="";
                    }
                    @Override
                    public void afterTextChanged(Editable s) {

                    }               });           }       });


        //Case Type
         actv.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                actv.addTextChangedListener(new TextWatcher() {

                    @Override
                    public void onTextChanged(CharSequence s, int start, int before, int count)
                    {
                        //  actv.setText("");
                        ctype="";
                        ctp.setText("");
                        ctp.setVisibility(View.GONE);
                        //ctp.setEnabled(false);
                    }


        actv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View arg1, int pos,
                        long id)
            {
              ctype =adapter.getItem(pos);
             Toast.makeText(getApplicationContext(), " Selected " + ctype,
    Toast.LENGTH_LONG).show();
                if(ctype.equals("Others"))
                {
                    ctp.setVisibility(View.VISIBLE);
                    ctp.setEnabled(true);
                }
```

```
                else
                {
                    ctp.setVisibility(View.GONE);
                    ctp.setText("");
                }
                        }
        });
      //Add Case Into Database
                btn.setOnClickListener(new OnClickListener()
                {
            @Override
            public void onClick(View v)

            {
              // String[] uniques;
                //Storing Time Hodden from user
                db.open();
                Calendar c1 = Calendar.getInstance();
                SimpleDateFormat sdf1 = new SimpleDateFormat("h:m:s a");
                String strdate1 = sdf1.format(c1.getTime());
                txtdate1 = (TextView) findViewById(R.id.textView1);
                txtdate1.setText(strdate1);
               toArr=actv1.getText().toString().split(",");
                if(toArr!=null) {
                    for (int i = 0; i < toArr.length; i++) {
                        a += db.getRecordcase(toArr[i]) + ",";
                    }
                    //Toast.makeText(getApplicationContext(),          "          "          +          a,
Toast.LENGTH_LONG).show();
                    toArr1 = a.split(",");
                    // String b="";
                    Set<String> set = new HashSet<String>();
                    Collections.addAll(set, toArr1);
                     uniques = set.toArray(new String[0]);
                    //println(uniques);
                    for (int p1 = 0; p1 < uniques.length; p1++) {
                        b += uniques[p1] + ",";
                        Toast.makeText(getApplicationContext(),      "univalues"      +      uniques[p1],
Toast.LENGTH_LONG).show();
                    }
                  //        Toast.makeText(getApplicationContext(),    "repet    tu    baad    "    +    b,
Toast.LENGTH_LONG).show();
                }
                  Long cui;
```

```
          Long id = db.insertcase(name.getText().toString().trim(), ctype,
loc.getText().toString().trim(), status.getText().toString().trim(), tv1.getText().toString().trim(),
txtdate1.getText().toString().trim(), dec.getText().toString().trim());




          Toast.makeText(getApplicationContext(), "Record Added",
Toast.LENGTH_LONG).show();
          id1 = db.getcaseid(name.getText().toString());
          int cid=  Integer.parseInt(id1);
          Toast.makeText(getApplicationContext(), "Case Id is " + id1,
Toast.LENGTH_LONG).show();
          for (int p11 = 0; p11 < uniques.length; p11++)
          {
            // b += uniques[p1] + ",";
             cui=db.insertcaseusers(cid,Integer.parseInt(uniques[p11]));
             Toast.makeText(getApplication(),"Users
"+uniques[p11].toString(),Toast.LENGTH_LONG).show();
          }
// Web service code
          if(wifiManager.isWifiEnabled())
          {
             new Asynccs().execute();
                    }
          else
          {
             Toast.makeText(getBaseContext(), "Wi-fi Not available",
Toast.LENGTH_LONG).show();
          }

          b="";
          a="";
        }
         db.close();
      }
    });
  }

  public class Asynccs extends AsyncTask<String, String, String> {


    @Override
    protected String doInBackground(String... strings) {
            try {

                SoapObject Request1 = new SoapObject(NAMESPACE, METHOD_NAME);
                Request1.addProperty("C_id", id1.toString());
                Request1.addProperty("name", name.toString());
                Request1.addProperty("type", ctype.toString());
                Request1.addProperty("loc", loc.toString());
                Request1.addProperty("status",status.toString());
                Request1.addProperty("sDate",tv1.toString());
                Request1.addProperty("sTime",txtdate1.toString());
                Request1.addProperty("desc",dec.toString());



                //texttxt.setText(logintxt.getText().toString());

                SoapSerializationEnvelope envelope = new
SoapSerializationEnvelope(SoapEnvelope.VER11);
                envelope.dotNet = true;
                envelope.setOutputSoapObject(Request1);

                HttpTransportSE androidHttpTransport = new HttpTransportSE(URL);
                try {
                    androidHttpTransport.call(SOAP_ACTION, envelope);
                    final SoapPrimitive result = (SoapPrimitive) envelope.getResponse();
                    addcase.this.runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            //String resultData=result.getProperty(0).toString();
                            //test.setText( result.toString());
                            //
Toast.makeText(getBaseContext(),result.toString(),Toast.LENGTH_LONG).show();
                            //  Log.d("hi","login done");
                            if (result.toString().equals("added")) {
                                Log.d("hi", "Case Added Successfully");
                                Toast.makeText(getBaseContext(), "Case Added Successfully",
Toast.LENGTH_LONG).show();
                            }

                        }                });

                } catch (Exception e) {
                    e.printStackTrace();
                }
            } catch (Exception er) {

                Toast.makeText(getApplicationContext(), "Try Again", Toast.LENGTH_LONG).show();
            }

            return null;
        }
    }



    @Override
    public void onClick(View v) {
        showDialog(0);
    }

    @Override
    @Deprecated
    protected Dialog onCreateDialog(int id) {
        DatePickerDialog dialog = new DatePickerDialog(this, datePickerListener,year,month,day);
        dialog.getDatePicker().setMaxDate(new Date().getTime());
        return dialog;
       // return new DatePickerDialog(this, datePickerListener, year, month, day);
    }

    private DatePickerDialog.OnDateSetListener datePickerListener = new DatePickerDialog.OnDateSetListener()
    {

        public void onDateSet(DatePicker view, int selectedYear,
                              int selectedMonth, int selectedDay) {

            tv1.setText(selectedDay + " / " + (selectedMonth + 1) + " / "
                    + selectedYear);
        }
};
```

Add Evidence Activity:

```
package com.example.anukoolsrivastav.forensics;
```

```
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;

/**
 * Created by Anukool Srivastav on 28-09-2014.
 */
public class addevidence extends Activity {
    ArrayList<String> items=new ArrayList<String>();
    Spinner id;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.add_evidence);
        items.add("1");
        items.add("2");
        id =(Spinner)findViewById(R.id.spinner);
        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,R.layout.spinner, items);
        id.setAdapter(adapter);
       // Button btn = (Button) findViewById(R.id.cam);
       /* btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
               selectImage();

            }
        });*/
    }
   /* ImageView v= (ImageView) findViewById(R.id.imageView);

    private void selectImage() {
        final CharSequence[] options = {"Take Photo", "Choose from Gallery", "Cancel"};
        AlertDialog.Builder builder = new AlertDialog.Builder(addevidence.this);
        builder.setTitle("Add Photo!");
        builder.setItems(options, new DialogInterface.OnClickListener() {
```

```
      @Override

      public void onClick(DialogInterface dialog, int item) {

        if (options[item].equals("Take Photo")) {
          Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
          File f = new File(android.os.Environment.getExternalStorageDirectory(),
"temp.jpg");
          intent.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(f));
          startActivityForResult(intent, 1);
        } else if (options[item].equals("Choose from Gallery")) {
          Intent intent = new Intent(Intent.ACTION_PICK,
android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
          startActivityForResult(intent, 2);
        } else if (options[item].equals("Cancel")) {
          dialog.dismiss();
        }
      }    });
    builder.show();
  }
    @Override
    protected void onActivityResult ( int requestCode, int resultCode, Intent data){

      super.onActivityResult(requestCode, resultCode, data);
      if (resultCode == RESULT_OK) {
        if (requestCode == 1) {
          File f = new File(Environment.getExternalStorageDirectory().toString());
          for (File temp : f.listFiles()) {
            if (temp.getName().equals("temp.jpg")) {
              f = temp;
              break;
            }
          }
          try {
            Bitmap bitmap;
            BitmapFactory.Options bitmapOptions = new BitmapFactory.Options();
            bitmap = BitmapFactory.decodeFile(f.getAbsolutePath(),
                bitmapOptions);
            v.setImageBitmap(bitmap);
            String path = android.os.Environment
                .getExternalStorageDirectory()
                + File.separator
                + "Phoenix" + File.separator + "default";
            f.delete();
```

```
                    OutputStream outFile = null;

                    File file = new File(path, String.valueOf(System.currentTimeMillis()) + ".jpg");
                    try {
                        outFile = new FileOutputStream(file);
                        bitmap.compress(Bitmap.CompressFormat.JPEG, 85, outFile);
                        outFile.flush();
                        outFile.close();

                    }

                } else if (requestCode == 2) {

                    Uri selectedImage = data.getData();
                    String[] filePath = {MediaStore.Images.Media.DATA};
                    Cursor c = getContentResolver().query(selectedImage, filePath, null, null, null);
                    c.moveToFirst();

                    int columnIndex = c.getColumnIndex(filePath[0]);
                    String picturePath = c.getString(columnIndex);
                    c.close();
                    Bitmap thumbnail = (BitmapFactory.decodeFile(picturePath));
                    Log.w("path of image from gallery......******************.........", picturePath +
"");
                    v.setImageBitmap(thumbnail);

                }           }
        }*/
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        //final TextView tv=(TextView)findViewById(R.id.tv1);
        getMenuInflater().inflate(R.menu.admin, menu);
        menu.add(0, Menu.FIRST,  Menu.NONE,"My Account").setIcon(R.drawable.user);
        SubMenu sub1=menu.addSubMenu(0,Menu.FIRST+1,Menu.NONE,"Change Password");
        SubMenu sub3=menu.addSubMenu(0,Menu.FIRST+2,Menu.NONE,"Logout");
        int positionOfMenuItem0 = 0; // or whatever...
        MenuItem item = menu.getItem(positionOfMenuItem0);
        SpannableString s = new SpannableString("My Account");
        s.setSpan(new ForegroundColorSpan(Color.argb(255, 67, 192, 251)), 0, s.length(), 0);
        item.setTitle(s);
        int positionOfMenuItem1 = 1; // or whatever...
        MenuItem item1 = menu.getItem(positionOfMenuItem1);
        SpannableString s1 = new SpannableString("Change Password");
        s1.setSpan(new ForegroundColorSpan(Color.argb(255,67,192,251)), 0, s1.length(), 0);
        item1.setTitle(s1);
        int positionOfMenuItem2 = 2; // or whatever...
        MenuItem item2 = menu.getItem(positionOfMenuItem2);
        SpannableString s2 = new SpannableString("LogOut");
        s2.setSpan(new ForegroundColorSpan(Color.argb(255,67,192,251)), 0, s2.length(), 0);
        item2.setTitle(s2);
        //tv.setText("Six Clicked");
        return true;
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if(item.getItemId()==1)
        {
            Intent cases8 = new Intent(addevidence.this,myaccnt.class);
            startActivity(cases8);
        }
        if(item.getItemId()==3)
        {
            Intent cases8 = new Intent(addevidence.this,Login.class);
            cases8.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK
|Intent.FLAG_ACTIVITY_CLEAR_TASK );
            startActivity(cases8);
            //finish();
        }
        if(item.getItemId()==2)
        {
            Intent cases8 = new Intent(addevidence.this,change.class);
            startActivity(cases8);
            //finish();
        }
        return true;

    }
    }
```

My Account :

```
package com.example.anukoolsrivastav.forensics;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.text.SpannableString;
import android.text.style.ForegroundColorSpan;
import android.view.Menu;
import android.view.MenuItem;
import android.view.SubMenu;
import android.widget.TextView;

/**
 * Created by Anukool Srivastav on 26-09-2014.
 */
public class myaccnt extends Activity {
    String a;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.my_account);
        Bundle getName = getIntent().getExtras();
        a=getName.getString("user");
        TextView tv;
        tv=(TextView)findViewById(R.id.editText8);
        tv.setText(a);
    }
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        //final TextView tv=(TextView)findViewById(R.id.tv1);
        getMenuInflater().inflate(R.menu.admin, menu);
        menu.add(0, Menu.FIRST,  Menu.NONE,"My Account").setIcon(R.drawable.user);
        SubMenu sub1=menu.addSubMenu(0,Menu.FIRST+1,Menu.NONE,"Change Password");
        SubMenu sub3=menu.addSubMenu(0,Menu.FIRST+2,Menu.NONE,"Logout");
        int positionOfMenuItem0 = 0; // or whatever...
        MenuItem item = menu.getItem(positionOfMenuItem0);
        SpannableString s = new SpannableString("My Account");
        s.setSpan(new ForegroundColorSpan(Color.argb(255, 67, 192, 251)), 0, s.length(), 0);
        item.setTitle(s);
        int positionOfMenuItem1 = 1; // or whatever...
        MenuItem item1 = menu.getItem(positionOfMenuItem1);
        SpannableString s1 = new SpannableString("Change Password");
        s1.setSpan(new ForegroundColorSpan(Color.argb(255,67,192,251)), 0, s1.length(), 0);
        item1.setTitle(s1);
        int positionOfMenuItem2 = 2; // or whatever...
        MenuItem item2 = menu.getItem(positionOfMenuItem2);
        SpannableString s2 = new SpannableString("LogOut");
        s2.setSpan(new ForegroundColorSpan(Color.argb(255,67,192,251)), 0, s2.length(), 0);
        item2.setTitle(s2);
        //tv.setText("Six Clicked");
        return true;
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if(item.getItemId()==1)
        {
            Intent cases8 = new Intent(myaccnt.this,myaccnt.class);
            startActivity(cases8);
        }
        if(item.getItemId()==3)
        {
            Intent cases8 = new Intent(myaccnt.this,Login.class);
            cases8.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK
|Intent.FLAG_ACTIVITY_CLEAR_TASK );
            startActivity(cases8);
            //finish();
        }
        if(item.getItemId()==2)
        {
            Intent cases8 = new Intent(myaccnt.this,change.class);
            startActivity(cases8);
           // finish();
        }
        return true;

    }
}
```

\

Search Evidence:

```
package com.example.anukoolsrivastav.forensics;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.text.SpannableString;
import android.text.style.ForegroundColorSpan;
import android.view.Menu;
import android.view.MenuItem;
import android.view.SubMenu;
import android.widget.ArrayAdapter;
import android.widget.Spinner;

import java.util.ArrayList;
public class searchevdnc extends Activity {
    ArrayList<String> items=new ArrayList<String>();
    ArrayList<String> items1=new ArrayList<String>();
    Spinner id,id2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.search_evidence);
        items.add("1");
        items.add("2");
        id =(Spinner)findViewById(R.id.spinner);
        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,R.layout.spinner, items);
        id.setAdapter(adapter);
        items1.add("1");
        items1.add("2");
        id2 =(Spinner)findViewById(R.id.spinner2);
        ArrayAdapter<String> adapter1 = new ArrayAdapter<String>(this,R.layout.spinner,
items1);
        id2.setAdapter(adapter1);
    }
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        //final TextView tv=(TextView)findViewById(R.id.tv1);
        getMenuInflater().inflate(R.menu.admin, menu);
        menu.add(0, Menu.FIRST,  Menu.NONE,"My Account").setIcon(R.drawable.user);
        SubMenu sub1=menu.addSubMenu(0,Menu.FIRST+1,Menu.NONE,"Change Password");
        SubMenu sub3=menu.addSubMenu(0,Menu.FIRST+2,Menu.NONE,"Logout");
        int positionOfMenuItem0 = 0; // or whatever...
        MenuItem item = menu.getItem(positionOfMenuItem0);
        SpannableString s = new SpannableString("My Account");
        s.setSpan(new ForegroundColorSpan(Color.argb(255, 67, 192, 251)), 0, s.length(), 0);
        item.setTitle(s);
        int positionOfMenuItem1 = 1; // or whatever...
        MenuItem item1 = menu.getItem(positionOfMenuItem1);
        SpannableString s1 = new SpannableString("Change Password");
        s1.setSpan(new ForegroundColorSpan(Color.argb(255,67,192,251)), 0, s1.length(), 0);
        item1.setTitle(s1);
        int positionOfMenuItem2 = 2; // or whatever...
        MenuItem item2 = menu.getItem(positionOfMenuItem2);
        SpannableString s2 = new SpannableString("LogOut");
        s2.setSpan(new ForegroundColorSpan(Color.argb(255,67,192,251)), 0, s2.length(), 0);
        item2.setTitle(s2);
        //tv.setText("Six Clicked");
        return true;
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if(item.getItemId()==1)
        {
            Intent cases8 = new Intent(searchevdnc.this,myaccnt.class);
            startActivity(cases8);
        }
        if(item.getItemId()==3)
        {
            Intent cases8 = new Intent(searchevdnc.this,Login.class);
            cases8.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK
|Intent.FLAG_ACTIVITY_CLEAR_TASK );
            startActivity(cases8);
          //  finish();
        }
        if(item.getItemId()==2)
        {
            Intent cases8 = new Intent(searchevdnc.this,change.class);
            startActivity(cases8);
            //finish();
        }
        return true;

    }
}
```

12.Bibliography

Below the list of websites that we have gone through for the development of the application:

Internet:

a) http://androidsourcecode.blogspot.in/2010/10/android-reading-inbox-sms.html
b) www.vogella.com/tutorials/**AndroidSQLite**/article.htm
c) www.codeproject.com/.../how-to-**get-data-from-database-in-android**
d) http://stackoverflow.com/questions/17794551/fetch-data-from-existing-sqlite-database

Various Books Followed:

a) Android Programming Guide for Beginners
b) Android Application development by O'Really Publication
c) Android Recipes Apress by Dave Smith publication

Attendance-:

Date/Month	August	September	October	November	December
1		P	p	P	
2		P	Holiday	Sunday	
3		P	Holiday	P	
4	P	P	Holiday	A	
5	P	P	Holiday	A	
6	P	P	Holiday	Holiday	
7	Sunday	Sunday	Holiday	P	
8	P	P	Holiday	P	
9	P	P	P	Sunday	
10	P	P	P	P	
11	P	P	P	P	
12	P	P	P(Sunday)	P	
13	P	P	P	A	
14	P	Sunday	P	P	
15	Holiday	P	P	A	
16	P	P	P	Sunday	
17	Sunday	P	P	P	
18	P	P	P	P	
19	P	P	Sunday	P	
20	P	P	P	P	
21	P	P	P	P	
22	P	P	P	P	
23	P	P	P	P	
24	Sunday	P	P	P	
25	P	P	P	P	
26	P	P	sunday	P	
27	P	P	P	P	
28	P	P	P		
29	P	P	P		
30	P	P	P		
31	Sunday	P	P		

MONTHLY GRADING

(By external internship in-charge from organization)

Name of the student: Navjot Singh Registration Number : 11302225

Project Title: Forensic expert system

Name of Organization & Address: VenturePact

Month: **September, 2014**

Parameter	**Grade(A+/A/B+/B/C+/C/D/E)**
Punctuality	A+
Regularity of Work	A
Improvement in Learning	A
Self-motivation/ Dedication/Initiative	B
Technical Competency	A
Discipline & Sincerity	B+
Problem Solving Capability	A

Date 15/10/2014 **Authorized Signatory** __________________

Name Sandeep Verma **Designation** Software Developer

MONTHLY GRADING

(By external internship in-charge from organization)

Name of the student: Navjot Singh Registration Number : 11302225

Project Title: Forensic expert system

Name of Organization & Address: VenturePact

Month: **October, 2014**

Parameter	**Grade(A+/A/B+/B/C+/C/D/E)**
Punctuality	A+
Regularity of Work	A
Progress in work since last appraisal	A
Improvement in Learning	A
Self-motivation/ Dedication/Initiative	B
Technical Competency	B+
Discipline & Sincerity	B+
Problem Solving Capability	A

Date 21/11/2014 **Authorized Signatory** ___________________

Name Sandeep Verma **Designation** Software Developer

MONTHLY GRADING

(By external internship in-charge from organization)

Name of the student: Navjot Singh Registration Number : 11302225

Project Title: Forensic expert system

Name of Organization & Address: VenturePact

Month: **November, 2014**

Parameter	**Grade(A+/A/B+/B/C+/C/D/E)**
Punctuality	A+
Regularity of Work	A
Progress in work since last appraisal	A
Improvement in Learning	A
Self-motivation/ Dedication/Initiative	B
Technical Competency	B+
Discipline & Sincerity	A
Problem Solving Capability	B+

Date 27/11/2014 **Authorized Signatory** __________________

Name Sandeep Verma **Designation** Software Developer