MINI PROJECT
(2020-21)

Crime Investigation System

MID-TERM REPORT

Institute of Engineering & Technology

Submitted by-

1) **Prachi Bansal (181500459)**
2) **Mani Bansal (181500364)**

Supervised By: -
Ms. Harvinder Kaur
Technical Trainer
Department of Computer Engineering & Applications

ACKNOWLEDGMENT

The project work in this report is an outcome of continuous work over a period and drew intellectual support from various sources. We would like to articulate our profound gratitude to all those people who extended their wholehearted co- operation and have helped us in completing this project successfully. We are thankful to our mentor **Harvinder Kaur** for teaching and assisting us in making the project successful. We would also like to thank other fellow mates for guiding and encouraging us throughout the duration of the project. We would also like to thank teaching staff for their constant encouragement, support and guidance which helped us in successfully completing the project work.

Prachi Bansal **Mani Bansal**

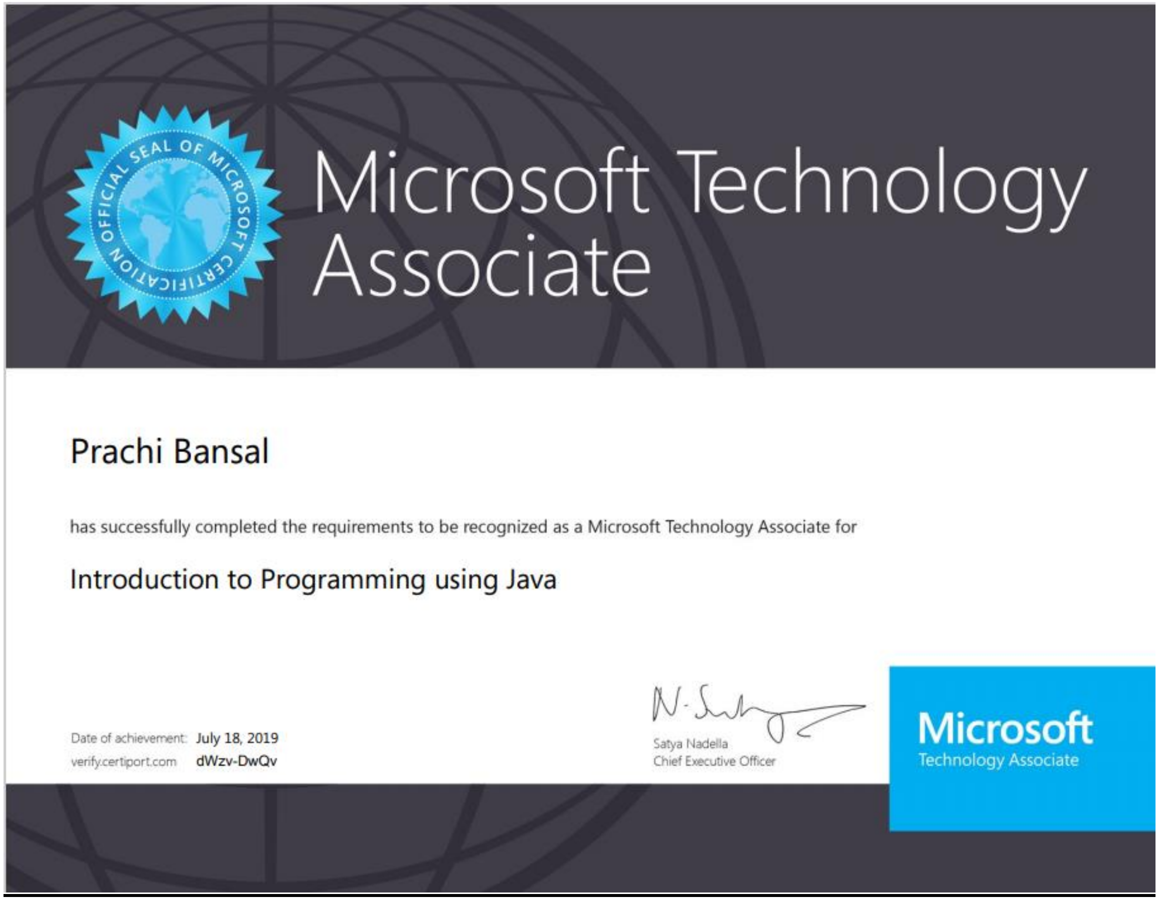
ABSTRACT

The proposed system applies to all Police stations across the country and specifically looks into the subject of Crime Records Management. It is well understood that Crime Prevention, Detection and Conviction of criminals depend on a highly responsive backbone of Information Management. The efficiency of the police function and the effectiveness with which it tackles crime depend on what quality of information it can derive from its existing records and how fast it can have access to it.

It is proposed to centralize Information Management in Crime for the purposes of fast and efficient sharing of critical information across all Police Stations across the territory. Initially, the system will be implemented across Cities and Towns and later on, be interlinked so that a Police detective can access information across all records in the state thus helping speedy and successful completion to cases. The System would also be used to generate information for pro-active and preventive measures for fighting crime.

The project has been planned to be having the view of distributed architecture, with centralized storage of the database. The application for the storage of the data has been planned. The standards of security and data protective mechanism have been given a big choice for proper usage. The application takes care of different modules and their associated reports, which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff.

Certificate



OFFICIAL SEAL OF MICROSOFT CERTIFICATION

Microsoft Technology Associate

Prachi Bansal

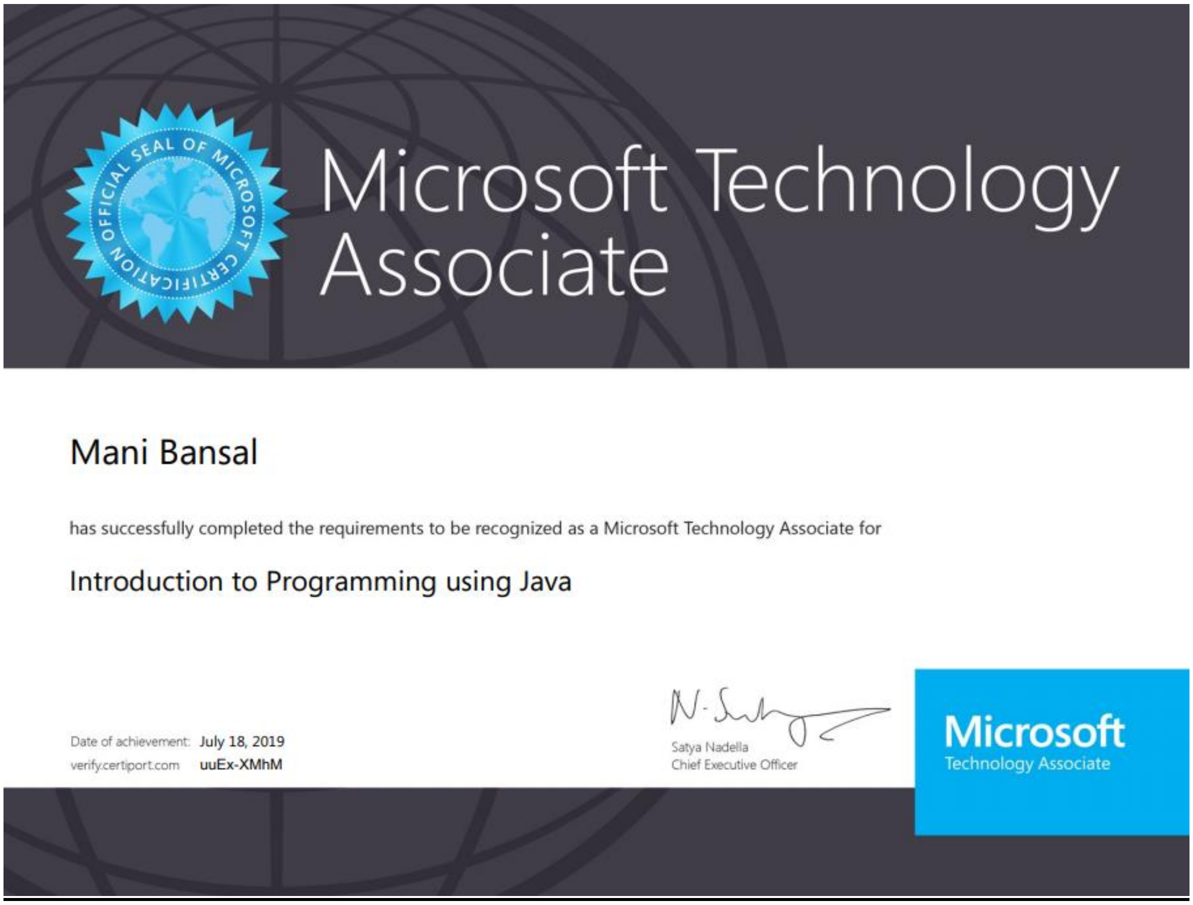
has successfully completed the requirements to be recognized as a Microsoft Technology Associate for

Introduction to Programming using Java

Date of achievement: July 18, 2019
verify.certiport.com dWzv-DwQv

Satya Nadella
Chief Executive Officer

Microsoft
Technology Associate



OFFICIAL SEAL OF MICROSOFT CERTIFICATION

Microsoft Technology Associate

Mani Bansal

has successfully completed the requirements to be recognized as a Microsoft Technology Associate for

Introduction to Programming using Java

Date of achievement: July 18, 2019
verify.certiport.com uuEx-XMhM

Satya Nadella
Chief Executive Officer

Microsoft
Technology Associate

CONTENTS

Introduction

1.1. Introduction to Project

Overview

This Crime Investigation Management System is being used to report all the crimes. Whenever a crime is being suspected or any unwanted activity is suspected than the system will record the activity and saved it as database for further assistance. The system also keep track of all the criminals from their past to present in order to maintained a good environment and so that in future if the criminal do any of the wrong activity than police can use the database which is being already stored in the servers for assisting that criminal activity.

Why new system?

- The system at any point of time can provide the details of the police station and the employees.
- The system at any point of time can provide the details of victims and the registered FIR’s
- The system at any point of time can provide the details of evidence and their sequence
- The system at any point of time can provide the details of existing charge sheets and their statuses.

2.1. Analysis Model

The model that is basically being followed is the WATER FALL MODEL, which states that the phases are organized in a linear order. First of all the feasibility study is done. Once that part is over the requirement analysis and project planning begins. If system exists one and modification and addition of new module is needed, analysis of present system can be used as basic model.

The design starts after the requirement analysis is complete and the coding begins after the design is complete. Once the programming is completed, the testing is done. In this model the sequence of activities performed in a software development project are: -

- Requirement Analysis
- Project Planning
- System design
- Detail design
- Coding
- Unit testing
- System integration & testing

Here the linear ordering of these activities is critical. End of the phase and the output of one phase is the input of other phase. The output of each phase is to be consistent with the overall requirement of the system. Some of the qualities of spiral model are also incorporated like after the people concerned with the project review completion of each of the phase the work done.

WATER FALL MODEL was being chosen because all requirements were known beforehand and the objective of our software development is the computerization/automation of an already existing manual working system.

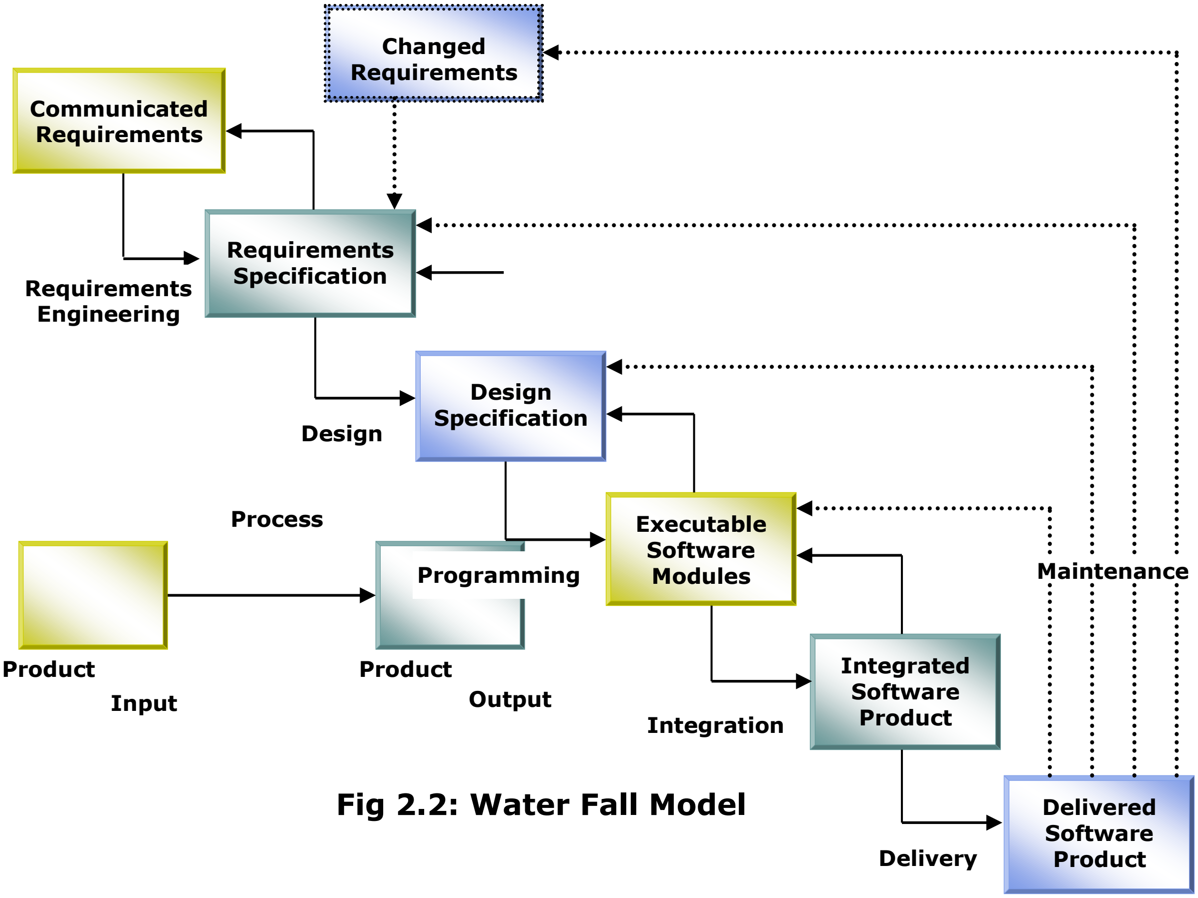


Fig 2.2: Water Fall Model

2.2 Existing System:

The existing system contains the about all the police stations that are registered as per the jurisdiction of the system. It also gets integrated with the employees who are working in these stations along with their designation.

2.4. Proposed System

The system after careful analysis has been identified to be presented with the following modules:

- **Police stations registration module:** This module maintains the information about all the police stations that are registered as per the jurisdiction of the system. It also gets integrated with the employees who are working in these stations along with their designation.
- **Victims FIR registration module:** This module maintains the information related to the first investigation report of the crime sequences that have taken place. The Fir registers all that a data that is necessary for the investigation to take place in proper length. It identifies the crime category and the crime nature.
- **Investigating evidence registration module:** This module makes a collection of information related to all the evidences that become categorically important under the normal sequence of the investigation, this module dynamically concentrates upon the changes that take place while the system of investigation is under process.

3. Software Requirement Specification

3.1. Overview

Purpose: The main purpose for preparing this document is to give a general insight into the analysis and requirements of the existing system or situation and for determining the operating characteristics of the system.

Scope of the Development Project:

Database Tier: The concentration is applied by adopting the Oracle 9i Enterprise versions. SQL is taken as the standard query language. The overall business rules are designed by using the power of PL/SQL components like stored procedures stored functions and database triggers.

User Tier: The use interface is developed is a browses specific environment to have distributed architecture. The components are designed using HTML standards and Java server pages power the dynamic of the page design.

Developer Responsibilities Overview:

The developer is responsible for:

- Developing the system, which meets the SRS and solving all the requirements of the system?
- Demonstrating the system and installing the system at client's location after the acceptance testing is successful.
- Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.
- Conducting any user training that might be needed for using the system.
- Maintaining the system for a period of one year after installation.

3.2. Hardware Requirements:

- Processor: Intel dual core or above
- Memory: 1GB
- Hard Drive: 50GB
- RAM: 1GB RAM or above

3.3. Software Requirements:

- Operating System: Windows 7 or higher
- Technology : Eclipse
- Database: MySQL
- Antivirus software

3.4. Performance Requirements:

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

4. SYSTEM DESIGN

4.1 Introduction

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

4.2 Data Flow Diagrams

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD'S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

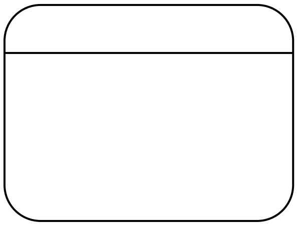
Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design.

A DFD is also known as a "bubble Chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

4.2.1 Dfd Symbols:

In the DFD, there are four symbols

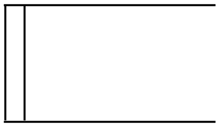
1. A square defines a source(originator) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data



Process that transforms data flow.

Source or Destination of data

Data flow



Data Store

4.2.2 Constructing a DFD:

Several rules of thumb are used in drawing DFD'S:

Process should be named and numbered for an easy reference. Each name should be representative of the process.

The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.

When a process is exploded into lower level details, they are numbered.

The names of data stores and destinations are written in capital letters.

Process and dataflow names have the first letter of each work capitalized.

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out.

Missing interfaces redundancies and like is then accounted for often through interviews.

4.2.3 Silent Feature of DFD's

1. The DFD shows flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
2. The DFD does not indicate the time factor involved in any process whether the dataflow take place daily, weekly, monthly or yearly.
3. The sequence of events is not brought out on the DFD.

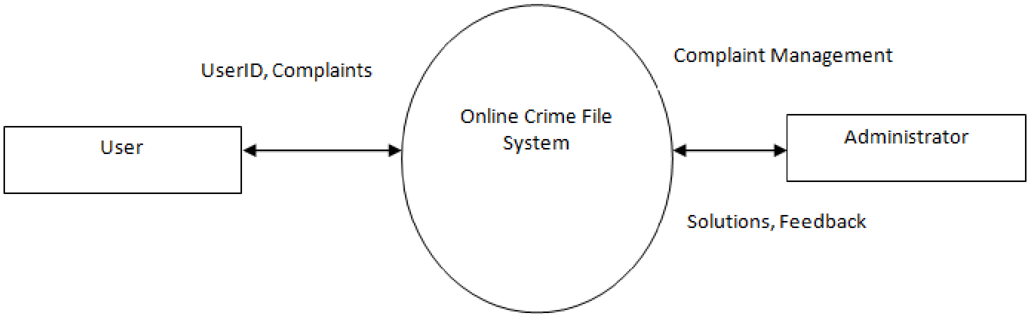
4.2.4 Data Flow:

1) A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The later is usually indicated however by two separate arrows since these happen at different type.
2) A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.
3) A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow produce some other data flow returns the original data into the beginning process.
4) A Data flow to a data store means update (delete or change).

5) A data Flow from a data store means retrieve or use. A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

Data Flow Diagram for our System

Context flow diagram



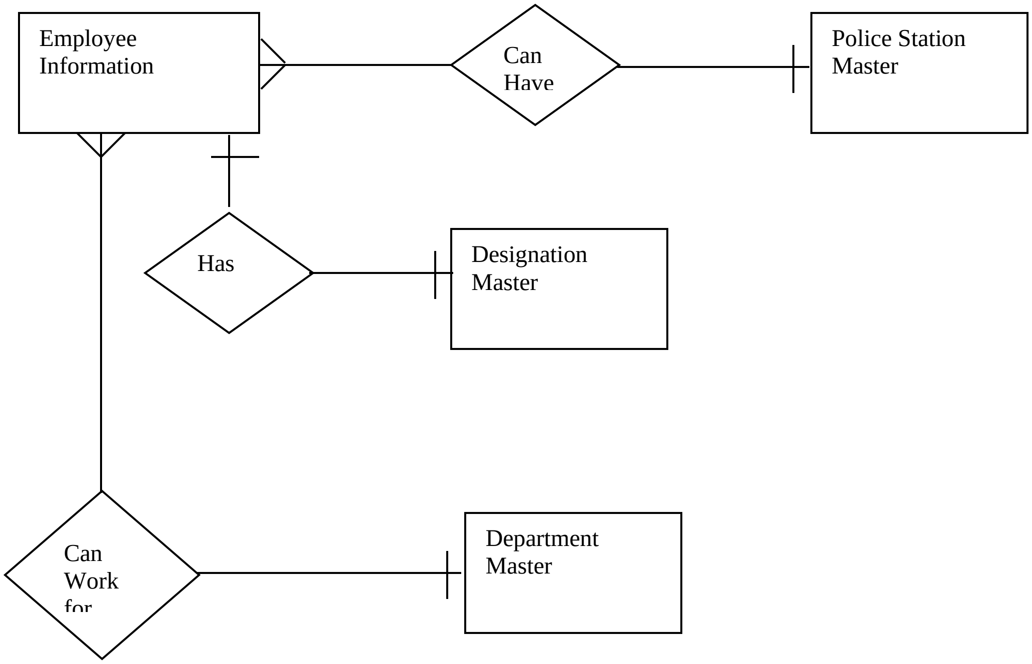
ER-Diagrams

- The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the date modeling activity the attributes of each data object noted is the ERD can be described resign a data object descriptions.
- The set of primary components that are identified by the ERD are

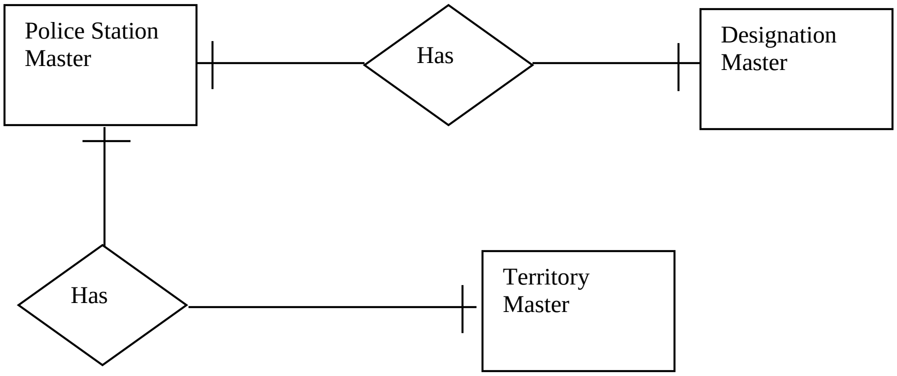
  ◆ Data object ◆ Relationships

  ◆ Attributes ◆ Various types of indicators.
- The primary purpose of the ERD is to represent data objects and their relationships.

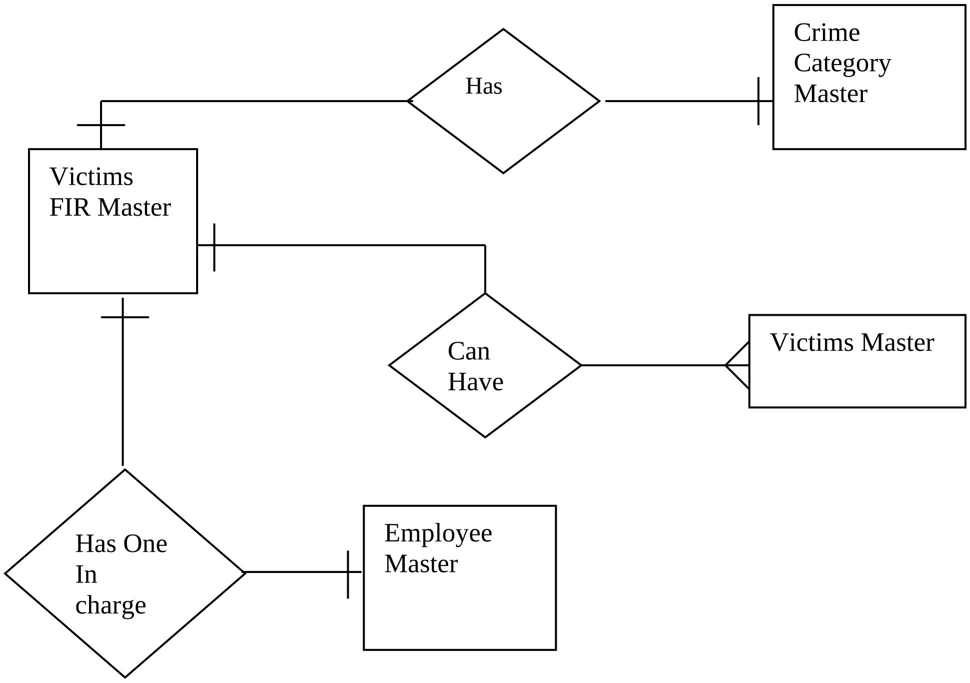
Employee Information



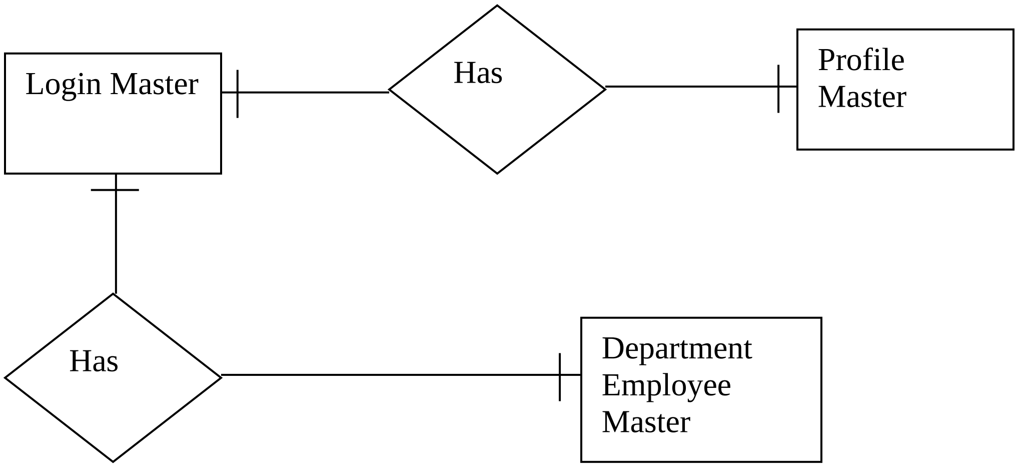
Police Station Master



Victims FIR Master



Login Master



TECHNOLOGIES USED

1. Introduction

1.1 Overview

Java is a general-purpose, concurrent, class-based, object-oriented computer programming language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to byte code (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2012, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 10 million users.Java was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

Java can be used to write applications and applets. A Java application is similar to any other high-level language program: It can only be compiled and then run on the same machine. An applet is compiled on one machine, stored on a server in binary, and can be sent to another machine over the Internet to be interpreted by a Java-aware browser. Java comes with a large library of ready-made classes and objects. The key difference between Java 1.0 and 1.1 was in this library. Similarly, Java 2.0 has a very much larger library for handling user interfaces (Swing by name) but only small changes to the core of the language.

1.2 Object-oriented Programming

Java supports object-oriented programming techniques that are based on a hierarchy of classes and well-defined and cooperating objects .

Classes: A class is a structure that defines the data and the methods to work on that data. When you write programs in Java, all program data is wrapped in a class, whether it is a class you write or a class you use from the Java API libraries. Classes in the Java API libraries define a set of objects that share a common structure and behavior.

Objects: An instance is a synonym for object. A newly created instance has data members and methods as defined by the class for that instance.

Well-Defined Boundaries and Cooperation: Class definitions must allow objects to cooperate during execution.

Inheritance and Polymorphism: One object-oriented concept that helps objects work together is inheritance. Inheritance defines relationships among classes in an object-oriented language. The relationship is one of parent to child where the child or extending class inherits all the attributes (methods and data) of the parent class. In Java, all classes descend from java.lang.Object and inherit its methods. Figure 1 shows the class hierarchy as it descends from java.lang.Objectfor the classes in the user interface example above. The java.lang.Objectmethods are also shown because they are inherited and implemented by all of its subclasses, which is every class in the Java API libraries. java.lang.Objectdefines the core set of behaviors that all classes have in common.

1.3 The Basic GUI (graphical user interface) Application

There are two basic types of GUI program in Java :stand-alone applications and (online)applets. An applet is a program that runs in a rectangular area on a Web page. Applets are generally small programs, meant to do fairly simple things, although there is nothing to stop them from being very complex. A GUI program offers a much richer type of user interface, where the user uses a mouse and keyboard to interact with GUI components such as windows, menus, buttons, check boxes, text input boxes, scroll bars, and so on.

JFrame and JPanel: In a Java GUI program, each GUI component in the interface isrepresented by an object in the program. One of the most fundamental types of component is the window. Windows have many behaviors. They can be opened and closed. They can be resized.They have “titles” that are displayed in the title bar above the window. And most important, they can contain other GUI components such as buttons and menus. Java, of course, has a built-in class to represent windows. There are actually several different types of window, but the most common type is represented by the JFrame class (which is included in the package javax.swing).

A **JFrame** is an independent window that can, for example, act as the main window of an application. One of the most important things to understand is that a JFrame object comes with many of the behaviors of windows already programmed in. In particular, we has one Jframe “InterfaceForm” that contents all the components, which enables the user to work on our system easily and draw any shape in Panel and the ability to be opened and closed.

JPanel is another of the fundamental classes in Swing . The basic JPanel is, again, just a blank rectangle. There are two ways to make a useful JPanel :

The first is to add other components to the panel.

The second is to draw something in the panel. Panel, which is used as a drawing surface.

Components and Layout: Another way of using a JPanel is as a container to hold other components. In our project, we used Eclipse IDE to create all components in JFrame and JPanel.

Events and Listeners: The structure of containers and components sets up the physical appearance of a GUI, but it doesn’t say anything about how the GUI behaves. That is, what can the user do to the GUI and how will it respond? GUIs are largely event-driven; that is, the program waits for events that are generated by the user’s actions (or by some other cause). When an event occurs, the program responds by executing an event-handling method. In order to program the behavior of a GUI, you have to write event-handling methods to respond to the events that you are interested in. The most common technique for handling events in Java is to use event listeners. A listener is an object that includes one or more event-handling methods. When an event is detected by another object, such as a button or menu, the listener object is notified and it responds by running the appropriate event-handling method. An event is detected or generated by an object. Another object, the listener, has the responsibility of responding to the event. The event itself is actually represented by a third object, which carries information about the type of event, when it occurred, and so on. This division of responsibilities makes it easier to organize large programs. As an example, consider the Undo or Redo button in our sample program. When the user clicks the button, an event is generated.

2.Graphics and Painting

Everything you see on a computer screen has to be drawn there, even the text. The (online) Java API includes a range of classes and methods that are devoted to drawing. In this section, I'll look at some of the most basic of these that helping us to achieve our project. The physical structure of a GUI is built of components. The term component refers to a visual element in a GUI, including buttons, menus, text-input boxes, scroll bars, check boxes, and so on. In Java, GUI components are represented by objects belonging to subclasses of the class java.awt.Component. In a graphical system, a windowing toolkit is usually responsible for providing a framework to make it relatively painless for a graphical user interface (GUI) to render the right bits to the screen at the right time. Both the AWT (abstract windowing toolkit) and Swing provide such a framework. But the APIs that implement it are not well understood by some developers -- a problem that has led to programs not performing as well as they could. In order to use graphics in Java programs, there are a number of libraries we need to import. For the sake of what will be covered in these notes, you need the following statements:

```
import javax.swing.JFrame;
import javax.swing.JPanel;
import java.awt.Graphics;
import java.awt.geom.*
import java.awt.Color;
import javax.swing.JLabel;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import javax.swing.ImageIcon;
import java.awt.BorderLayout;
import java.awt.event.*;
import java.awt.image.BufferedImage;
import java.awt.image.MemoryImageSource;
import java.awt.image.PixelGrabber;
import java.io.File;
import java.io.IOException;
import javax.imageio.*;
```

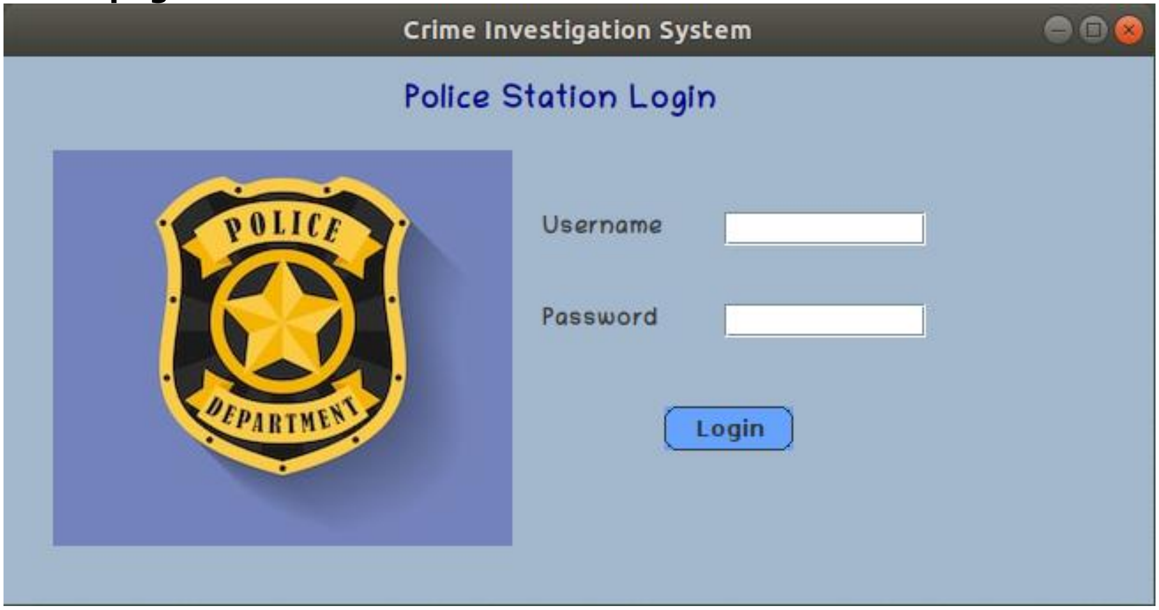
MySQL Database

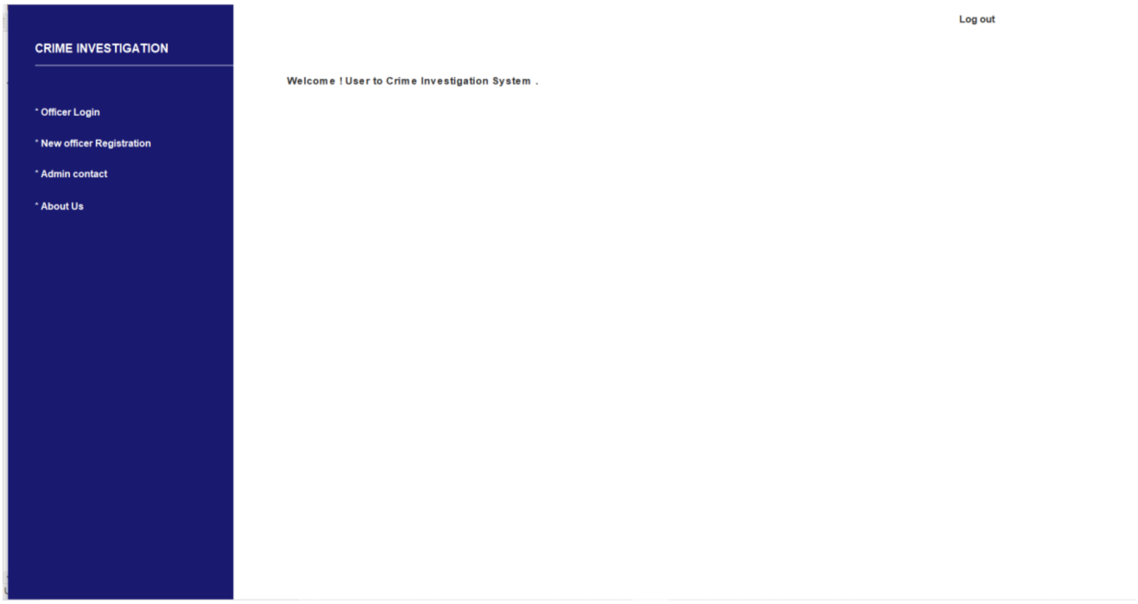
MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons −

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

Output Screen

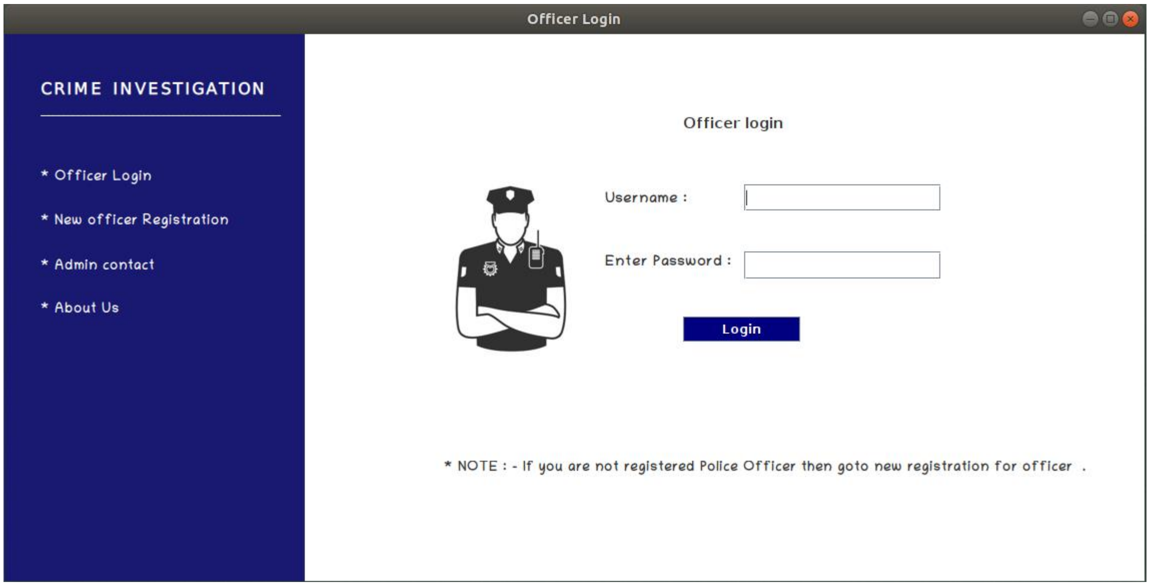
Home page Screen:





Welcome ! User to Crime Investigation System .

Login Page:



Officer Login

CRIME INVESTIGATION

- * Officer Login
- * New officer Registration
- * Admin contact
- * About Us

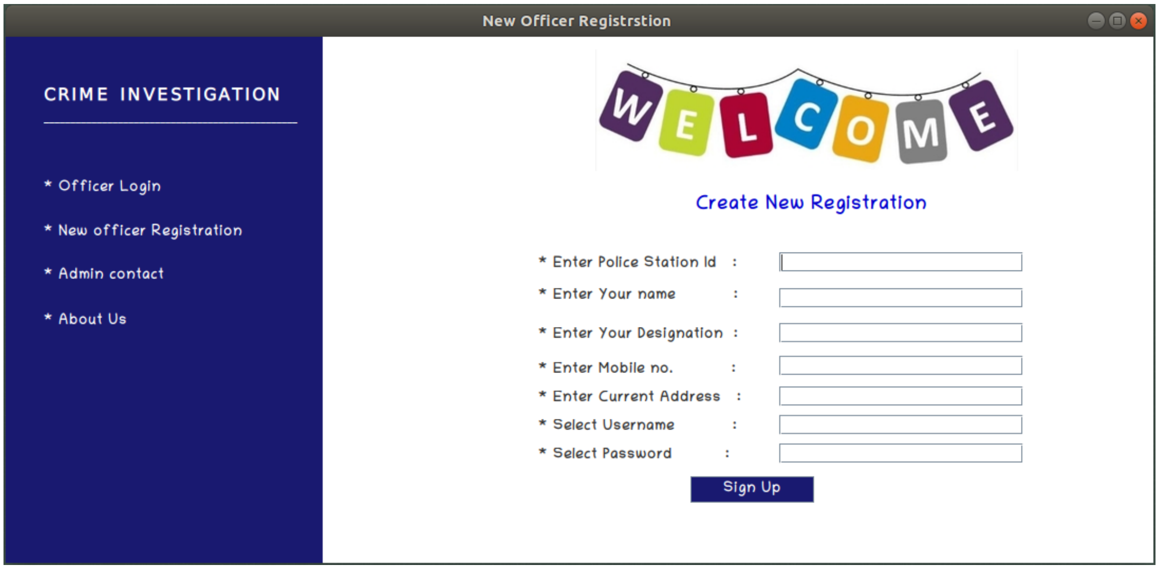
Officer login

Username :

Enter Password :

* NOTE : - If you are not registered Police Officer then goto new registration for officer .

Registration page:



New Officer Registrstion

CRIME INVESTIGATION

- * Officer Login
- * New officer Registration
- * Admin contact
- * About Us

Create New Registration

* Enter Police Station Id :

* Enter Your name :

* Enter Your Designation :

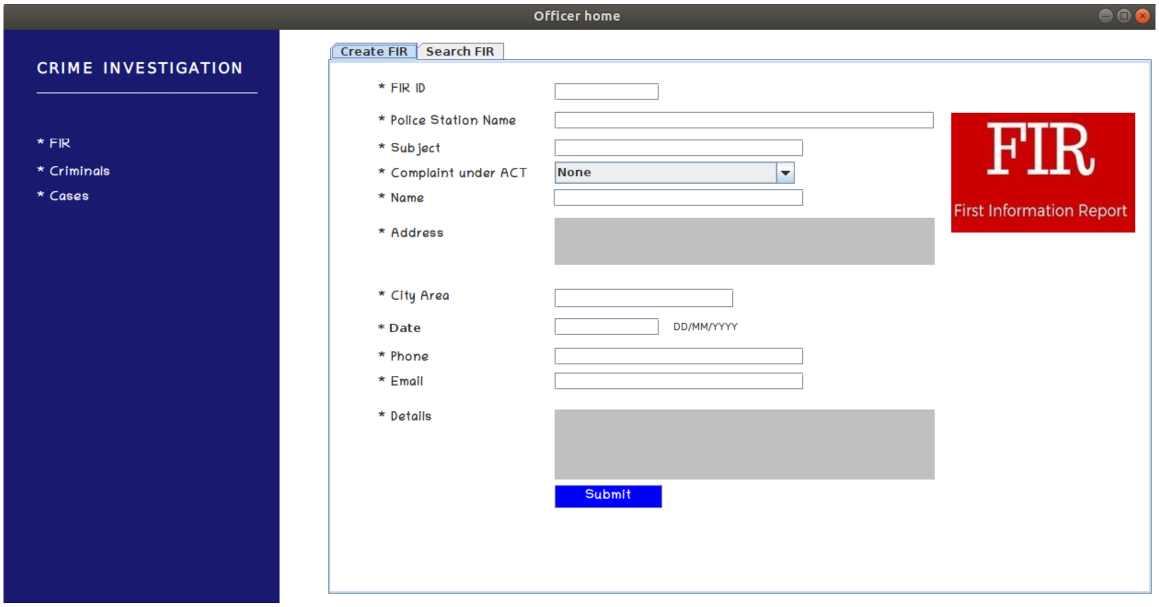
* Enter Mobile no. :

* Enter Current Address :

* Select Username :

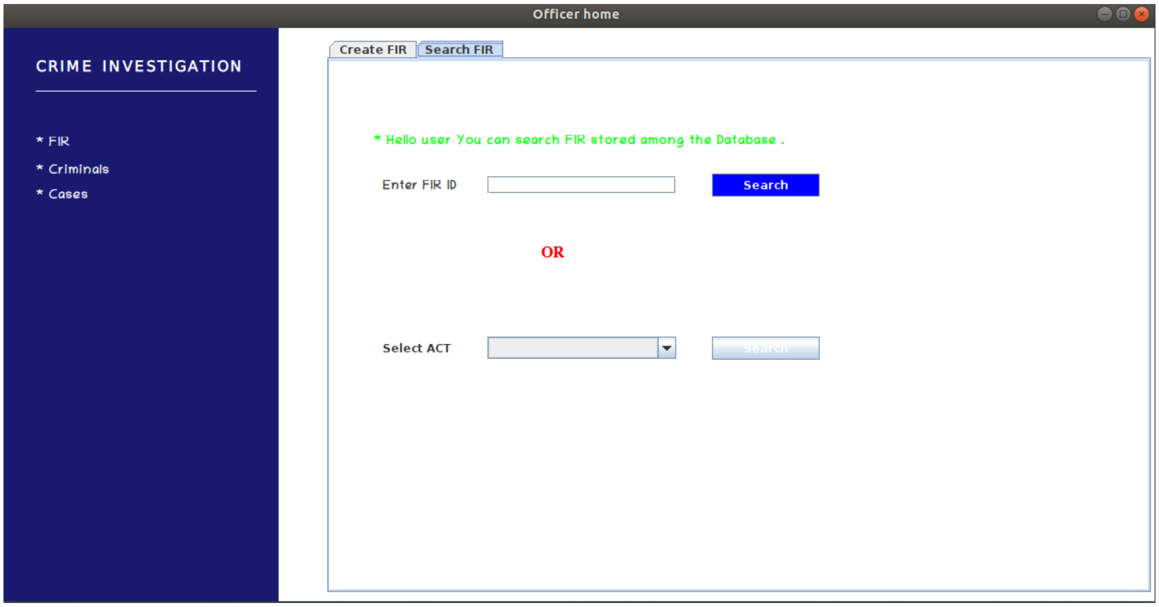
* Select Password :

Create FIR:



Complaint under ACT: None

Search FIR:



* Hello user You can search FIR stored among the Database .

TESTING

6. Testing

1. The process of executing a system with the intent of finding an error.
2. Testing is defined as the process in which defects are identified, isolated, subjected for rectification and ensured that product is defect free in order to produce the quality product and hence customer satisfaction.
3. Quality is defined as justification of the requirements
4. Defect is nothing but deviation from the requirements
5. Defect is nothing but bug.
6. Testing --- The presence of bugs
7. Testing can demonstrate the presence of bugs, but not their absence
8. Debugging and Testing is not the same thing!
9. Testing is a systematic attempt to break a program or the AUT

Testing Methodologies:

- Black box Testing: is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application.
Usually Test Engineers are involved in the black box testing.
- White box Testing: is the testing process in which tester can perform testing on an application with having internal structural knowledge.
Usually The Developers are involved in white box testing.

- Gray Box Testing: is the process in which the combination of black box and white box tonics are used.

6.1 STLC (Software Testing Life Cycle)

Test Planning:

- Test Plan is defined as a strategic document which describes the procedure how to perform various testing on the total application in the most efficient way.
- This document involves the scope of testing,
- Objective of testing.
- Areas that need to be tested.
- Areas that should not be tested.
- Scheduling Resource Planning.

Types of Testing:

- Regression Testing: is one of the best and important testing. Regression testing is the process in which the functionality, which is already tested before, is once again tested whenever some new change is added in order to check whether the existing functionality remains same.
- Re-Testing: is the process in which testing is performed on some functionality which is already tested before to make sure that the defects are reproducible and to rule out the environments issues if at all any defects are there.
- Static Testing: is the testing, which is performed on an application when it is not been executed.ex: GUI, Document Testing
- Dynamic Testing: is the testing which is performed on an application when it is being executed.ex: Functional testing.
- Alpha Testing: it is a type of user acceptance testing, which is conducted on an application when it is just before released to the customer.
- Beta-Testing: it is a type of UAT that is conducted on an application when it is released to the customer, when deployed in to the real time environment and being accessed by the real time users.
- Installation Testing: it is the process of testing in which the tester try to install or try to deploy the module into the corresponding environment by following the guidelines produced in the deployment document and check whether the installation is successful or not.

Conclusion

This software provides facility for reporting online crimes, complaints, missing persons, show most wanted person details mailing as well as chatting. This software is developed with scalability in mind. Additional modules can be easily added when necessary. The software is developed with modular approach. All modules in the system have been tested with valid data and invalid data and everything work successfully. Thus the system has fulfilled all the objectives identified and is able to replace the existing system.

The project has been completed successfully with the maximum satisfaction of the organization. The constraints are met and overcome successfully. The system is designed as like it was decided in the design phase. The project gives good idea on developing a full-fledged application satisfying the user requirements.

The system is very flexible and versatile. This software has a user-friendly screen that enables the user to use without any inconvenience. Validation checks induced have greatly reduced errors. Provisions have been made to upgrade the software. The application has been tested with live data and has provided a successful result. Hence the software has proved to work efficiently.

Scope for Future Enhancement

- In future system will provide mail facility to editors. Editors can send mail to other stations editor.
- In future system will allow user to register complains online.
- In future system will provide facility to send message.
- The method of video conferencing can be added to make the project livelier.
- Users can view the progress of their complaint online.
- By the future technology user can view the case details and progress of the complaints on their mobile phones

BIBLIOGRAPHY

- **FOR MySQL**

  www.w3schools.com

- **FOR JavaSwing**

  https://www.javatpoint.com/java-swing

  https://en.wikipedia.org/wiki/Swing_(Java)

  https://www.guru99.com/java-swing-gui.html