Institute of Arts and Sciences (Chiniot Campus)

Govt College University Faisalabad

Software Requirements Specification

(SRS Document)

For

Police Station Management System

By

Zainab Javed 225350
Pakeeza Ghulam Muhammad 225336

Instructor
Dr. Waqar Hussain
Bachelor in Computer Science (2022-2026)

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Declaration

We hereby declare that the project work entitled “Police Station Management System” submitted as a semester project for Object Oriented Programing(CSI-401), is our own work, based on personal study, research, and sources used in its preparation, whether they be books, articles, reports, and lecture notes.
We also certify that this project work is submitted in the partial fulfillment of the requirements to pass the OOP course and has not previously been submitted for assessment in any academic capacity and not copied/plagiarized in part/whole work of other persons. In case of any violation and negligence, we shall be fully responsible to face any kind of consequences in terms of grades.

Member	**Student Name**	**Sign**
1	*Zainab*	
2	*Pakeeza*	

2. Introduction

2.1 Purpose

The purpose of this document is to describe the Police Station Management System. This document contains the functional, behavioral and non-functional requirements of the project and it also contains the guide lines for system engineers and designers to start working the project. The main purpose of this project is to maintain easy circulation system using computers in the police stations.

2.2 Intended Audience and Reading Suggestions

- *Police men*
- *Officers*

2.3 Project Scope

The project is developed mainly for the use of the Police Station authority. This project will work as a complete user interface. This system can be implemented under various conditions. We can add new information and when we require making reusability, it is possible as there is flexibility in all the modules. The language used for developing this project is Object Oriented Programming in C++ which is very efficient in storing and retrieving the information as needed.

2.4 References

- *Books*
  - *C++ How to Program 8th edition*
  - *Object Oriented Programming 4th edition*

3. Overall Description

3.1 Product Perspective

Police Station Management System is a replacement for the existing system which was using a number of books for storing the information about Crime records, Cases, Criminals, FIRs etc. The new system would be more efficient and easy for managing and manipulating.

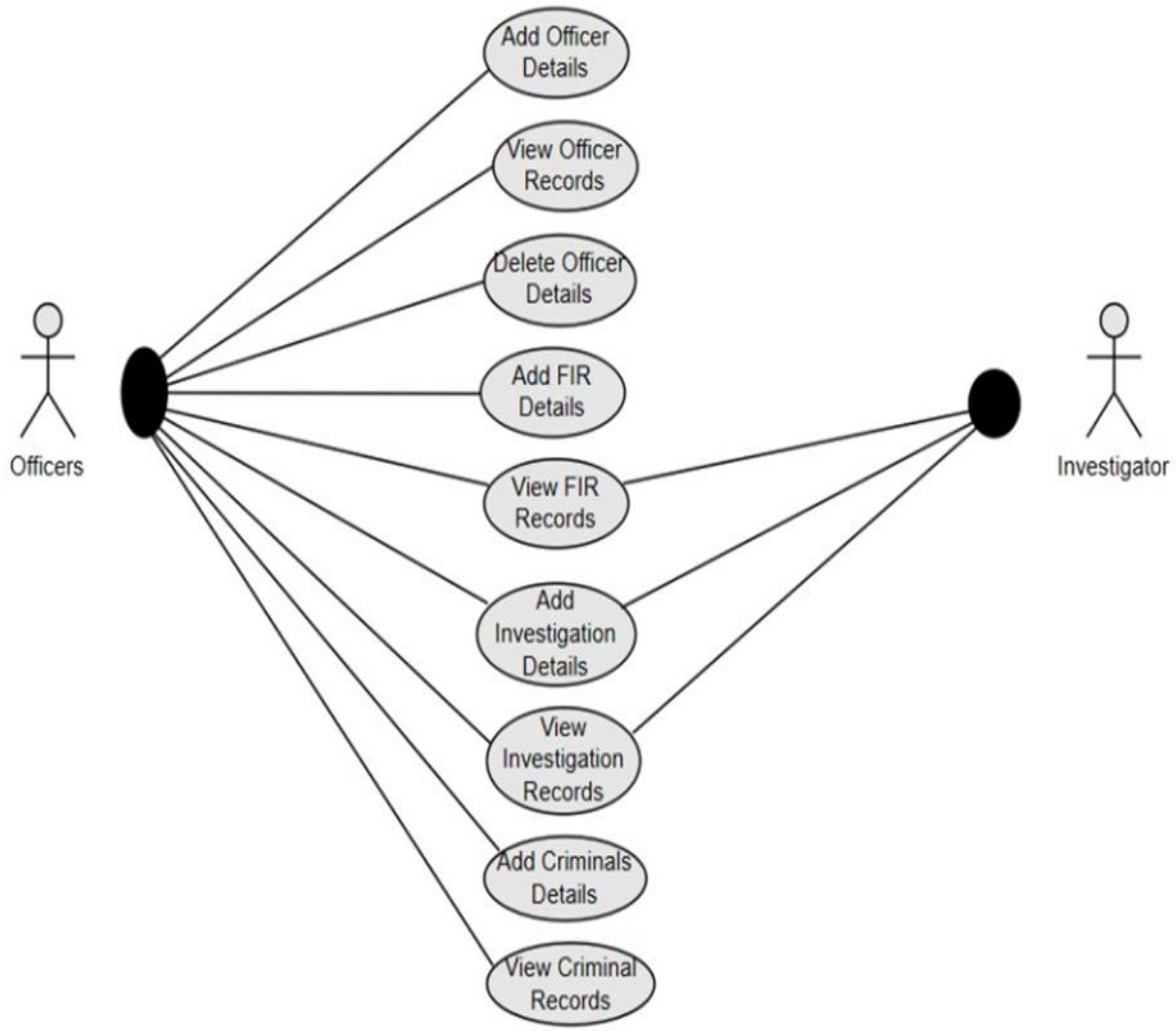
3.2 Product Features

The features that are available to the persons manage the system are:

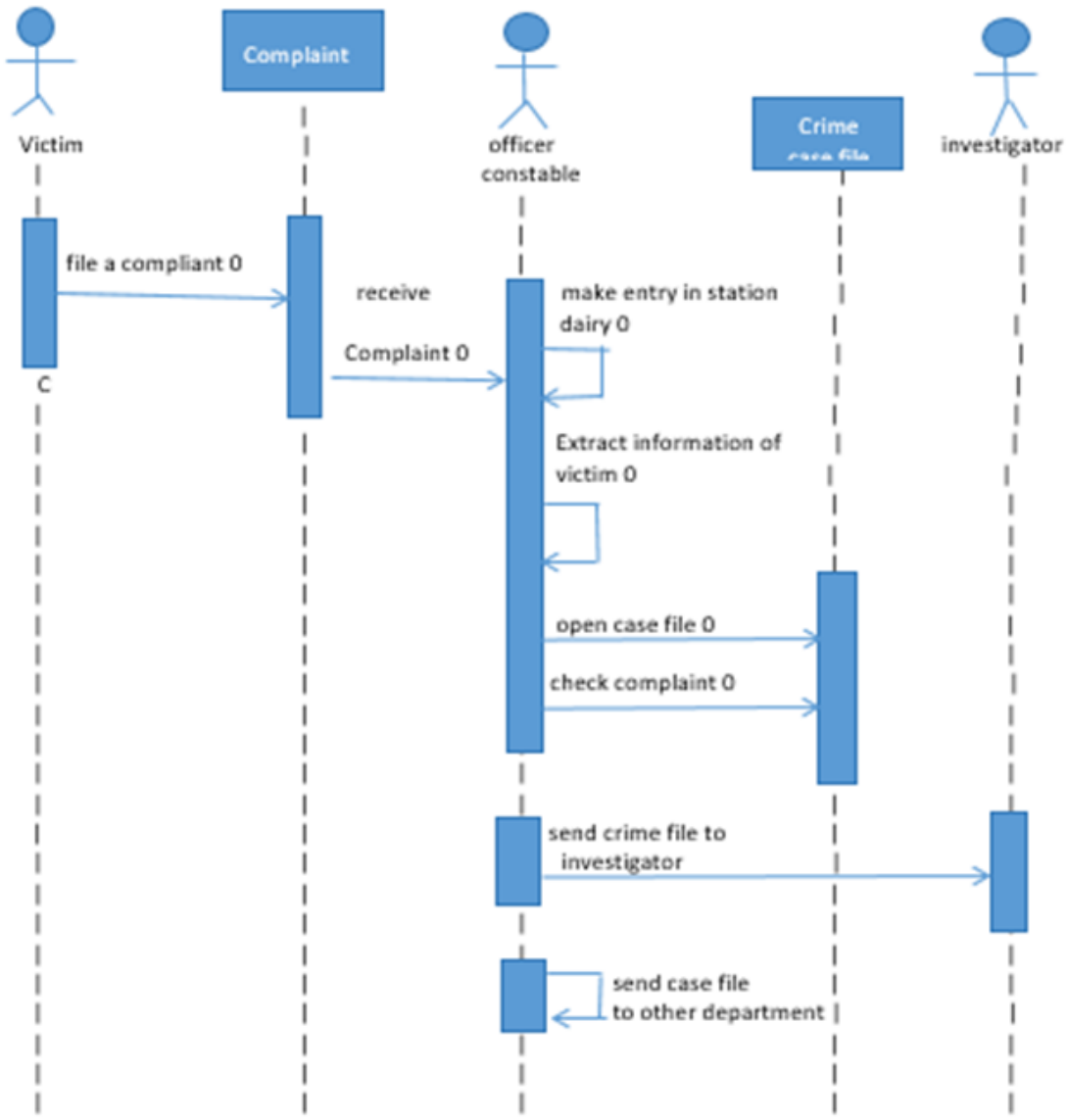
- *Can add information to the database.*
- *Can modify the information.*
- *Can remove any unwanted information.*
- *Can check the information in the added records and access all the records.*

3.3 User Classes and Characteristics

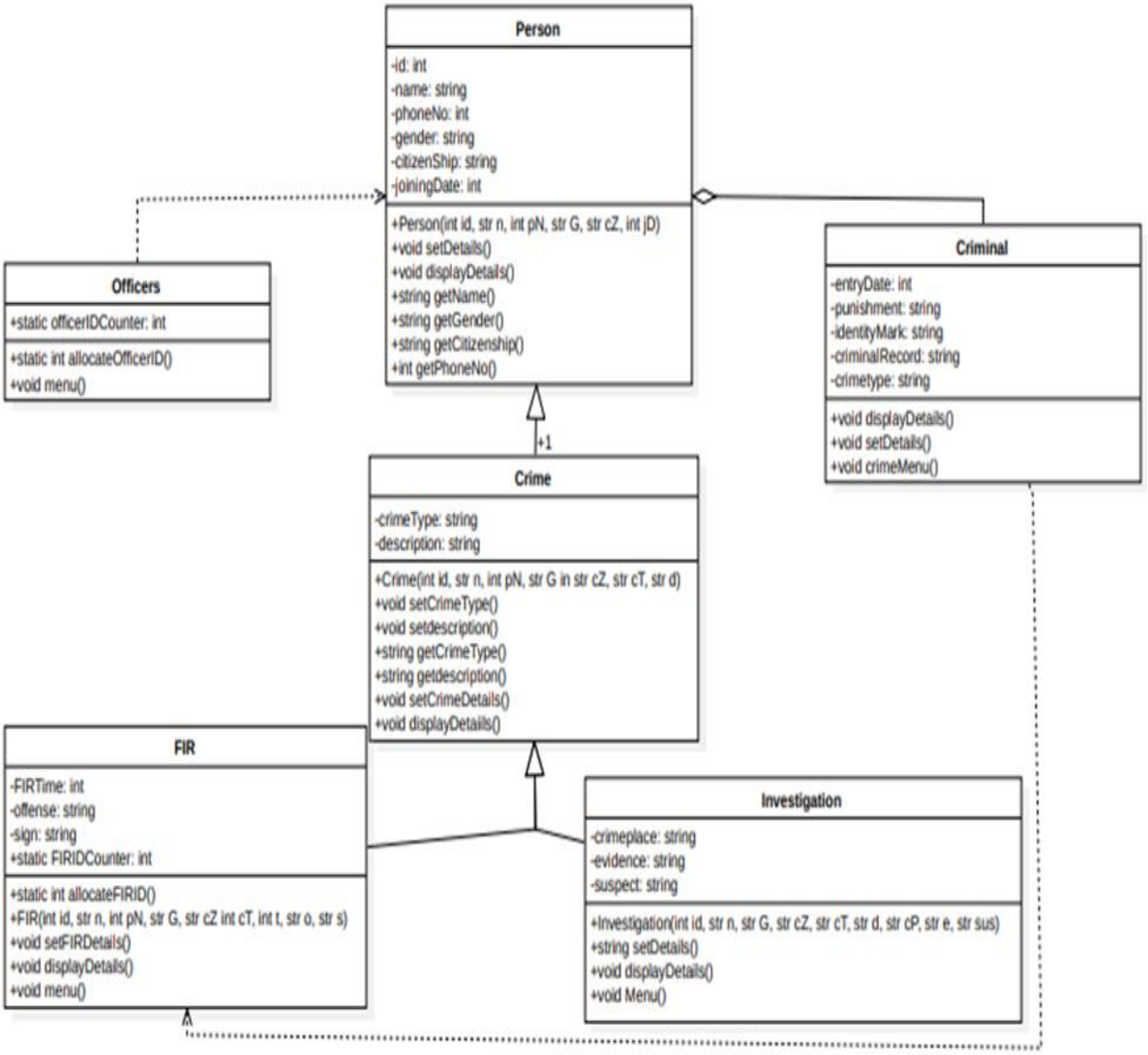
Use Case Diagram



Sequence Diagram



Class Diagram



3.4 Operating Environment

The system will be compatible operating systems windows.

3.5 Design and Implementation Constraints

- *The system will be developed using C++ Programming Language.*
- *The database will be used simple txt files.*
- *Compatibility with latest version of C++.*

3.6 User Documentation

Comprehensive user documentation will be provided, including user manuals and guides for Officers.

3.6.1 Concepts

Object-Oriented Programming (OOP) Principles:

• Class

Classes Used:

- **Person:** *Persons with an ID, name and personal info etc.*
- **Officers:** *Represents a officers with a ID, name, post etc*.
- **Crime:** *Crime represent a crime type and description.*
- **FIR:** *FIR represent complain in details.*
- **Investigation:** *Represent inquiry of crime in details.*
- **Criminal:** *Represent a criminal with an ID, name etc.*
- **Explanation:** *A class is a user-defined data type that we can use in our program, and it works as an object constructor, or a "blueprint" for creating objects.*

• Encapsulation

- ***Private Members***: *Various data members in classes are marked as private, encapsulating the internal details of each class.*

- ***Explanation***: *Encapsulation is the bundling of data and the methods (functions) that operate on the data into a single unit, i.e., a class. This helps in controlling access to the data by providing access specifiers (public, private, protected).*

• Inheritance

- *Crime class inherits from person class.*
- *FIR and Investigation inherits from Crime class.*
- *Crime use private inheritance.*

- ***Explanation****: Inheritance is the mechanism by which one class can inherit properties and behavior from another class promoting code reusability.*

• Composition

- *Criminal class own person class.*
- ***Explanation***: *Composition refers to the design principle where a class is composed of or contains objects of other classes, enabling the creation of more complex and reusable structures.*

• Friend Class

- *Officer's class is a friend of Person class.*
- ***Explanation:*** *Because of the friend relationship, the Officer class can access all private data members of the Person class.*

• Polymorphism

- *All classes use same signature define for different behaviors.*

Software Engineering Principles:

DRY

- *Avoiding duplicating code by encapsulating common functionalities within classes and methods.*

Code Reusability

- *Inheritance for Reusability: Inheritance is used to reuse the functionality.*

System Design

- *The Police Station Management System is a secure and user-friendly platform designed to streamline law enforcement operations.*

Readability

- *Descriptive Identifiers: Variable and function names are chosen descriptively, contributing to code readability.*

Maintainability

- *Structured Code: The code is organized into classes and functions enhancing readability and maintainability.*

Other Concepts

- **File Handling**

  In program we using file handling to persistently store and retrieve data, separating concerns by saving data in individual files etc.

- **Exception Handling**

  We used error-handling techniques such as file existence checks and appropriate user input validation. For increased dependability, confirm the existence of the file before opening it, respond properly to successful actions, and manage invalid user input.

- **Vectors**

  Person class, there's a member variable phoneNumbers declared as vector<int> phoneNumbers. Vectors are used for storing and managing dynamic arrays of data. Vectors are used in several places, such as storing phoneNumbers.

3.7 Assumptions and Dependencies

The assumptions are:
The coding should be error free. The system should be user friendly so that the users can easily access data which have more storage capacity and provide fast access to database. Search facilities and provide quick search results.
The dependencies are:
The specific hardware and software due to which the product will run on the basis of listing requirements and specification, the system will be developed and run. The end-users should have knowledge about the system. Updates are to be made correctly and data entered without any mistakes.

4. System Feature

4.1 System Feature 1

The system we have implemented appears to be a Police Station Management System with features related to Officers, Crime (FIR), Criminals, and Investigations. Here are the key features of our system:

4.1.1 Description and Priority

The Police Station Management System is a software application designed to manage and organize information related to police officers, cases (FIR), investigations, and criminals. It includes modules for adding new officers, recording FIR details, managing case investigations, and maintaining criminal records. The system allows users to view and search for specific information.
Priority: *Medium*

4.1.2 Officers Module:

. ***Add New Officer:***

- *Collects information about a new officer, including name, phone number, joining date, gender, and citizenship.*
- *Generates a unique ID for each officer.*

View Officers' Data:

- *Allows the user to view details of officers by searching for their ID.*

Delete Officer Data:

- *Enables the removal of an officer's data based on their ID.*

4.1.3 Cases Module:

FIR Management:

- *Adds new FIR information, including details about the crime, suspect, and evidence.*
- *Generates a unique ID for each FIR.*
- *Allows viewing details of FIRs based on their ID.*

Investigation Management:

- *Manages information related to case investigations, including details about the crime, evidence, suspect, etc.*
- *Generates a unique ID for each investigation.*
- *Allows viewing details of investigations based on their ID.*

4.1.4 Criminals Module:

Criminals Information:

- *Adds new information about criminals, including their personal details, entry date, punishment, identity marks, criminal records, and crime type.*
- *Generates a unique ID for each criminal.*
- *Allows viewing details of criminals based on their ID.*

4.2 Stimulus/Response Sequences

4.2.1 Add a New Police Officer

- *User selects the "Add Officer" option.*
- *System prompts user to enter officer details (ID, name, phone number, etc.).*
- *User inputs officer details.*
- *System validates and adds the new officer to the database.*
- *System responds with a success message or an error message if validation fails.*

4.2.2 Record a New FIR

- *User selects the "Add FIR" option.*
- *System prompts user to enter FIR details (complainant name, description, etc.).*
- *User inputs FIR details.*
- *System validates and records the new FIR.*
- *System responds with a success message or an error message if validation fails*.

4.2.3 View Crime Details

- *User searches for a specific crime using crime ID.*
- *System retrieves and displays crime details.*
- *System provides options to view associated FIRs, Investigation and other relevant information.*

4.2.4 Update Criminal Records

- *User selects the "Criminal Records" option.*
- *System displays a list of criminals.*
- *User selects a criminal to update or adds a new criminal record.*
- *System allows the user to modify or add details.*
- *System responds with a success message or an error message if validation fails.*

4.3 Functional Requirements

REQ-1 Data Persistence:

- *All entered data (officers, FIRs, investigations, criminals) should be stored persistently in files.*
- *The system should be able to read from these files to display information when needed.*

REQ-2 Error Handling:

- *The system must handle invalid inputs gracefully and provide informative error messages.*
- *Users should be guided to correct their inputs in case of errors during data entry or retrieval.*

REQ-3 Menu:

- *Users should be able to navigate through different sections of the application using a menu.*
- *The menu options should be clear and concise, providing a user-friendly interface.*

REQ-4 Security:

- *User authentication data should be securely stored, and the system should protect against unauthorized access.*
- *File access permissions should be set appropriately to ensure data integrity and confidentiality.*

REQ-5 Termination:

- *Users should have a clear option to terminate the application or log out securely.*

REQ-6 User Interface Design:

- *The user interface should be intuitive and well-designed, making it easy for users to interact with the system.*
- *Proper formatting and alignment of information should enhance readability.*

REQ-7 Dynamic ID Allocation:

- *The system should dynamically allocate unique IDs for officers, FIRs, investigations, and criminals to avoid conflicts and ensure uniqueness.*

5. External Interface Requirements

5.1 User Interfaces

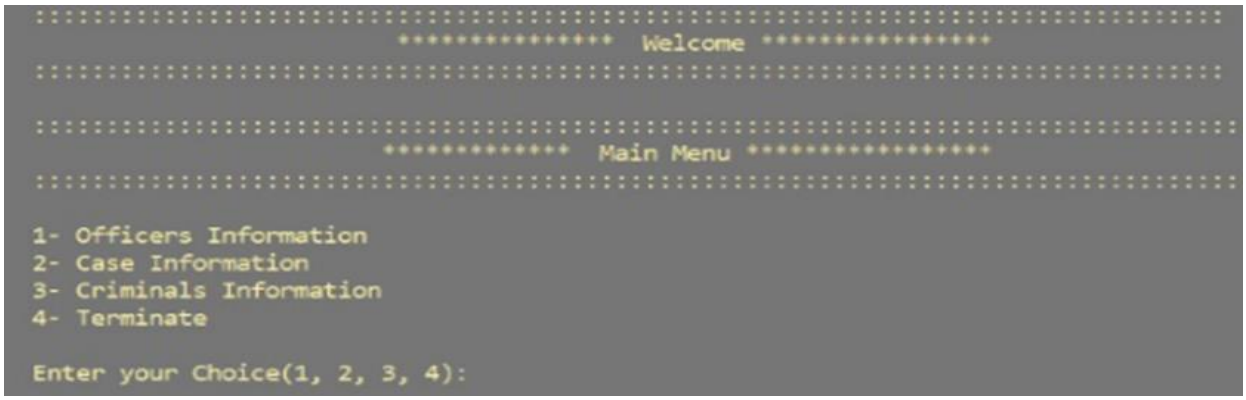
The system will have a console-based interface suitable for interaction within a command-line environment on Windows.

5.1.1 Getting Start

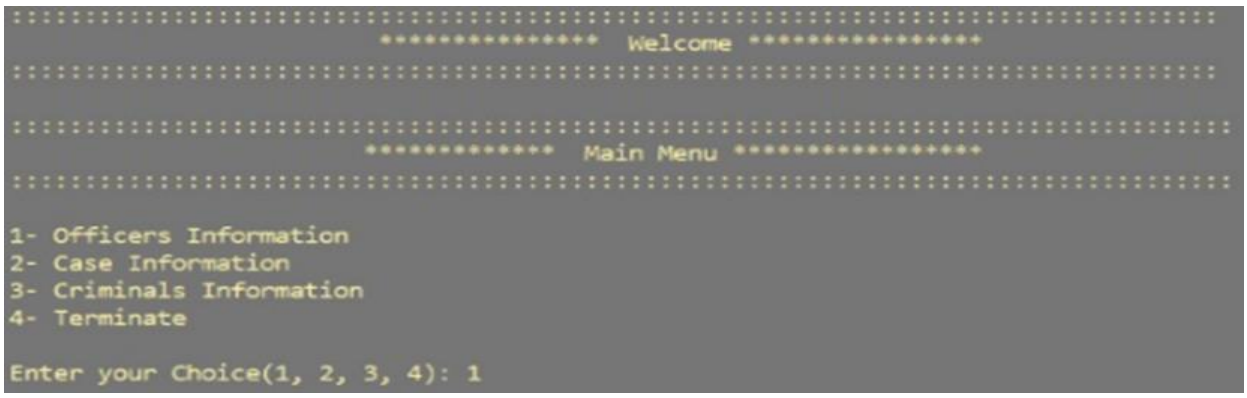
First of all, when you use the program, you will be greeted with a welcome screen displaying the project title and a brief description. Press "Enter" to proceed to the main menu.

5.1.2 Main menu

In this menu you have option depending upon your authority.



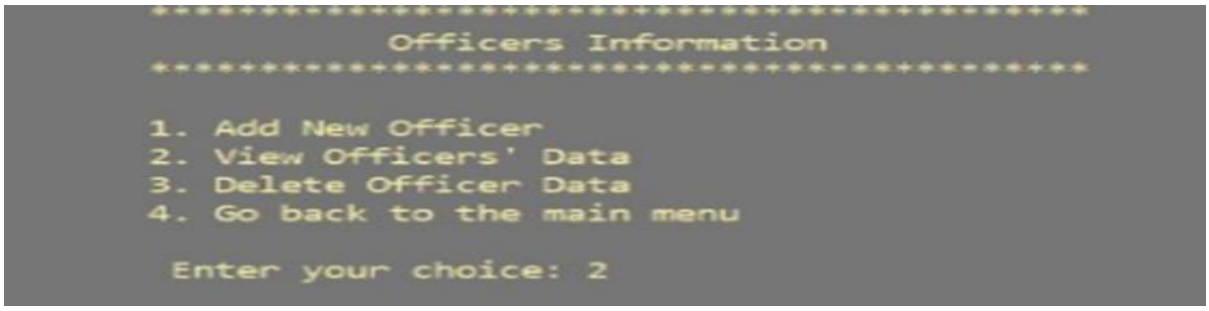
5.1.3 Enter 1



5.1.4 Officers Information

5.1.5 Add New Officer

5.1.6 Enter 2

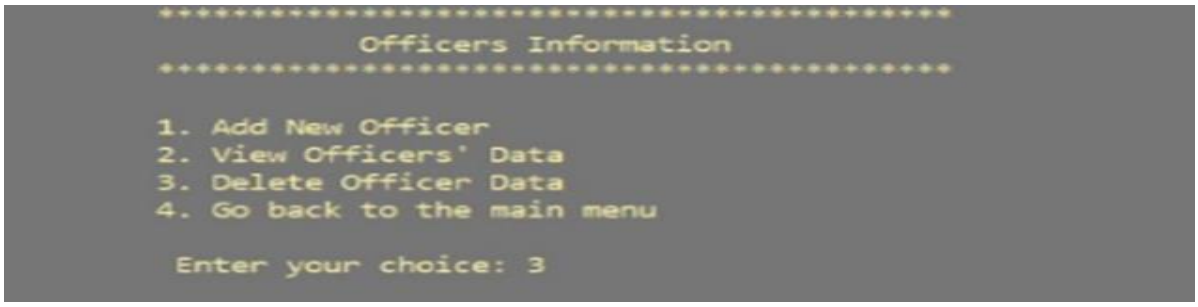


5.1.7 View Officer Data

5.1.8 Enter ID

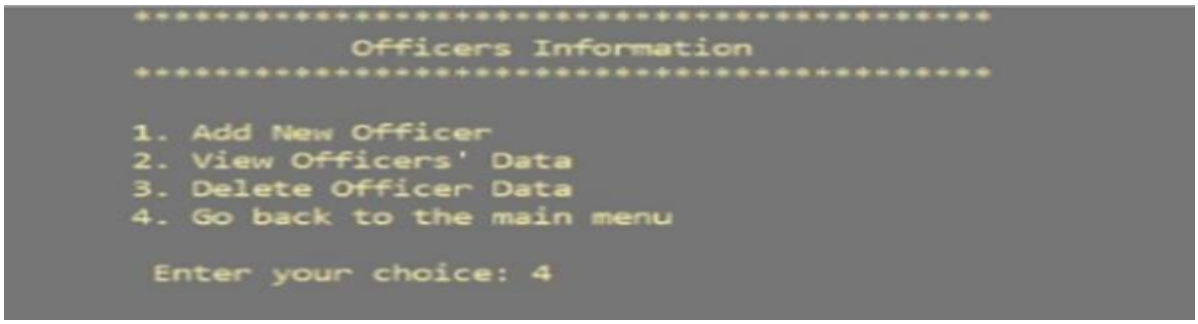
5.1.9 View record

5.1.10 Enter 3

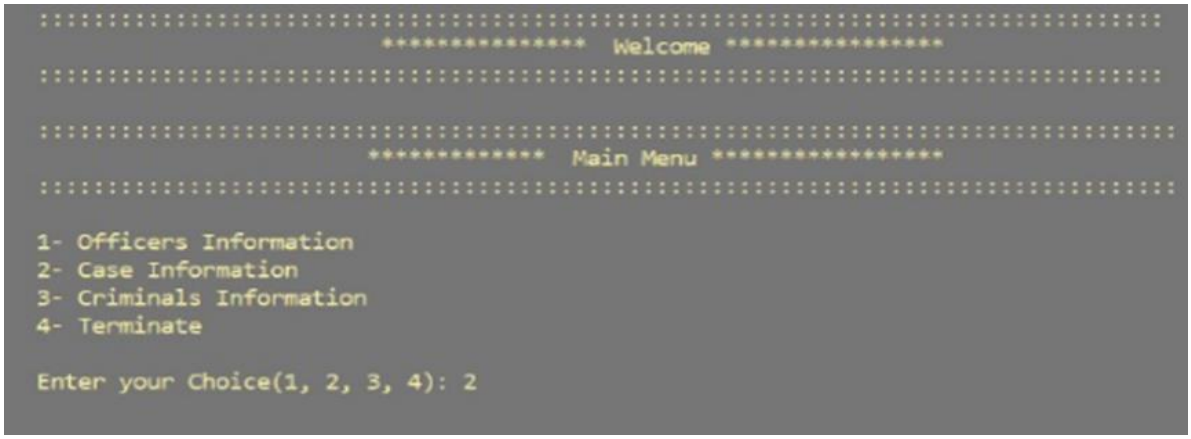


5.1.11 Enter ID which you want to delete

5.1.12 back to menu



5.1.12 Enter 2



5.1.14 Enter 1

5.1.15 Add FIR Data

5.1.16 Enter 2 for view Data

5.1.17 View Record Data

5.1.18 Enter 3 to back

5.1.19 Enter 2 for investigation

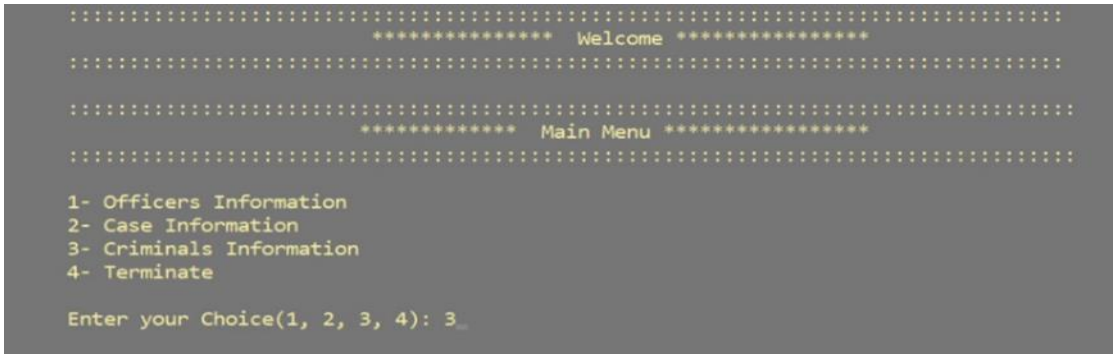
5.1.20 Enter 1 for add Info

5.1.21 Add Info

5.1.22 Enter 2 for view Data

5.1.23 Enter 3 for back to menu

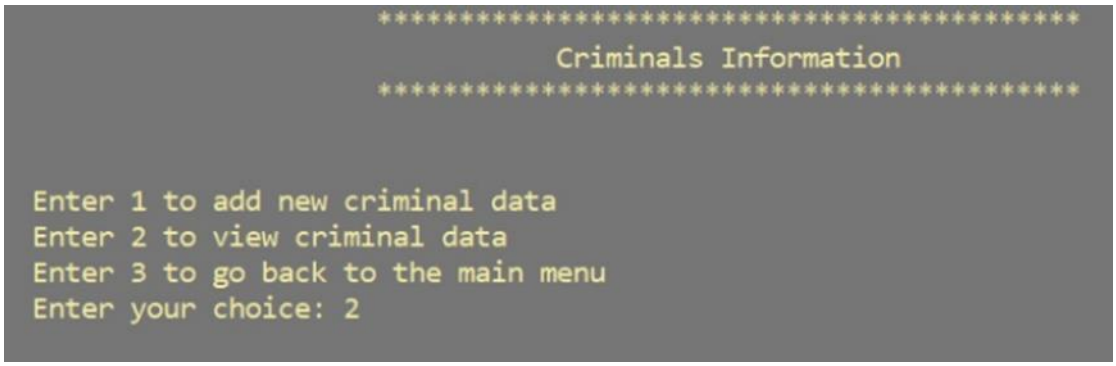
5.1.24 Enter 3 for Criminal Info



5.1.25 Enter 1 for add info

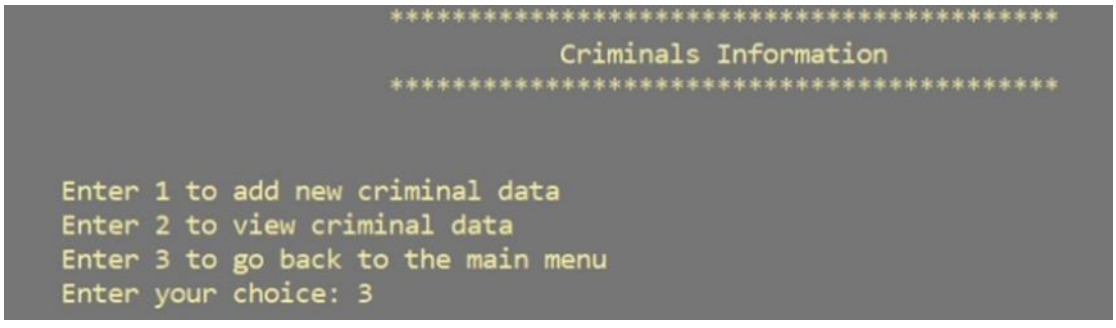
5.1.26 Add Data

5.1.27 Enter 2 for view

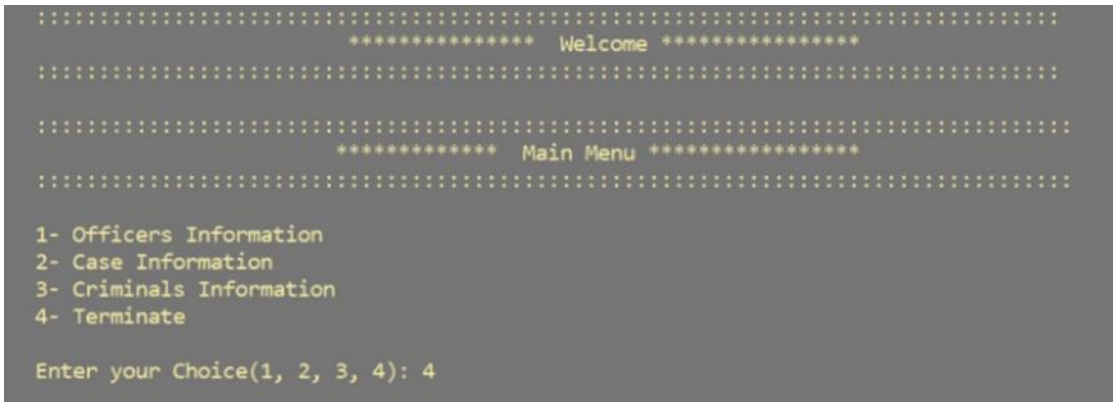


5.1.28 View Record

5.1.29 Enter 3 for back to menu



5.1.30 Enter 4 to Terminate



5.2 Hardware Interfaces

The system will require computing hardware compatible with the Windows operating system.

5.3 Software Interfaces

Software will depend on the security features provided by the operating system and the language C++.The system will have a direct connection with the database .The data's are stored in the database and retrieved as per requirements.

5.4 Communications Interfaces

The application does not require external communication interfaces, as it operates within a standalone console environment.

6. Other Nonfunctional Requirements

6.1 Performance Requirements

The performance of the system should be fast and accurate. Response of the system to an operation should be within reasonable time. The system should be able to handle large amount of data. The system should also capable for updating the data such as Officers, Complaints, Criminal records, FIR, Investigation etc.

6.2 Safety Requirements

The system should ensure the security of user data stored locally on the machine.

6.3 Security Requirements

The application should implement measures to prevent unauthorized access to police station data and results stored on the local machine.

6.4 Software Quality Attributes

***Reliability**: The system should function without errors, providing a stable experience for Officers.*
***Usability**: The console-based interface should be designed for ease of use, with clear prompts and instructions for users interacting via the command line.*