**A**

**Project Report**

**on**

**Criminal Record Management System**

****

**B.Sc (H) Computer Science, Semester IV, 2022**

**Department of Computer Science**

**Aryabhatta College**

**University of Delhi**

Submitted by: Supervisor:

**Raman(20059570019) Dr. Deepak Sharma**

**Nitika Kumari(20059570006)**

**Contents**

**Acknowledgement**

**Certificate**

[**Problem Statement**](#_heading=h.tyjcwt) **5**

[**Process Model**](#_heading=h.rsl52cbq6k7) **6**

[1.1 Overall Description](#_heading=h.1t3h5sf) 7

[1.1.2 User Characteristics](#_heading=h.17dp8vu) 8

[1.1.3 General Constraints](#_heading=h.3rdcrjn) 8

[1.1.4 Assumptions and Dependencies](#_heading=h.26in1rg) 8

[1.2.1 User Interfaces](#_heading=h.lnxbz9) 9

[1.2.2 Hardware Interfaces](#_heading=h.35nkun2) 9

[1.2.3 Software Interfaces](#_heading=h.1ksv4uv) 9

[1.3 Functional Requirements](#_heading=h.44sinio) 10

[1.3.1 FR 1](#_heading=h.2jxsxqh) 10

[1.3.2 FR 2](#_heading=h.z337ya) 10

[1.3.3 FR 3](#_heading=h.u2gro1wkxto5) 10

[1.3.4 FR 4](#_heading=h.1y810tw) 11

[1.3.5 FR 5](#_heading=h.4i7ojhp) 11

[1.3.6 FR 6](#_heading=h.2xcytpi) 11

[1.3.7 FR 7](#_heading=h.1ci93xb) 11

[1.3.8 FR 8](#_heading=h.28xj11p4xf5q) 11

[1.3.9 FR 9](#_heading=h.2bn6wsx) 12

[1.3.10 FR 10](#_heading=h.qsh70q) 12

[1.3.11 FR 11](#_heading=h.3as4poj) 12

[1.3.12 FR 12](#_heading=h.1pxezwc) 12

[1.3.13 FR 13](#_heading=h.6ero5av1s45s) 12

[1.4 Performance Requirement](#_heading=h.3fwokq0) 13

[1.5 Design Constraint](#_heading=h.1v1yuxt) 13

[1.6 Data Flow Diagram](#_heading=h.uttn7d46k5n7) 14

[1.6.1 Context Level DFD](#_heading=h.13pch6wbar37) 14

[1.6.1 1 Level DFD](#_heading=h.1mrcu09) 14

[1.6.1 2 Level DFD](#_heading=h.2lwamvv) 15

[1.7 Data Dictionary](#_heading=h.4k668n3) 17

[1.8 Use case Diagram](#_heading=h.2zbgiuw) 20

[1.9 Sequence Diagram](#_heading=h.e57i1zbvzki2) 21

[**2. Estimations**](#_heading=h.3cqmetx) **22**

[2.1 Function Points](#_heading=h.1rvwp1q) 22

[2.2 Efforts](#_heading=h.4bvk7pj) 24

[**3. Scheduling**](#_heading=h.jm2az4hm8po5) **25**

[**4.1 Risk Table**](#_heading=h.2u19ai4zspp9) **26**

[**5. Design**](#_heading=h.3q5sasy) **26**

[5.1 Screen Design](#_heading=h.25b2l0r) 27

[5.2 ER Diagram](#_heading=h.ib544euet6i0) 35

[5.3 Database Design](#_heading=h.soj5cxeibui) 36

[**6. Coding**](#_heading=h.ax7swmcdnpgy) **37**

[**7. Testing**](#_heading=h.7yyds7py7xon) **93**

[7.1 Flowgraph](#_heading=h.xvir7l) 93

[7.2 DD Flow Graph](#_heading=h.1baon6m) 95

[7.3 Cyclomatic Complexity](#_heading=h.2afmg28) 95

[7.4 Test Cases](#_heading=h.39kk8xu) 96

[**8. Future Scope**](#_heading=h.1opuj5n) **104**

[**9. References**](#_heading=h.48pi1tg) **104**

**Acknowledgement**

In performing our assignment, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this assignment gives us much Pleasure. We would like to show our gratitude to our professor Dr. Deepak Sharma who introduced us to the Methodology of work, and for giving us a good guideline for assignment throughout numerous consultations. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in writing this assignment.

Many people, especially our classmates and team members, have made valuable comment suggestions on this proposal which gave us inspiration to improve our assignment. We thank all the people for their help directly and indirectly to complete our assignment.

**Raman Dr. Deepak Sharma**

**(20059570019)**

**Nitika Kumari**

**(20059570006 )**

**`**

**Certificate**

This is to certify that **(Raman)** and **(Nitika Kumari)** successfully carried out the completion of the project entitled **“(Criminal Record Management System)”** under my supervision. The Project has been submitted as per the requirement of the Lab based on **Software Engineering** of B.Sc. (H) Computer Science, IV Semester.

**Raman Sharma Dr. Deepak Sharma**

**(20059570019) (**Supervisor)

**Nitika Kumari**

**(20059570006 )**

# 

# 

# 

# 

# 

# 

# 

# **Problem Statement**

In the existing system manual methods are used for maintaining criminal records and filing FIR which is not an efficient way. There are chances of losing data. This application will solve these problems and provide a database for storing data.

The aim of the proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces manual work. The existing system in police stations has several disadvantages and many more difficulties to work well. The proposed system tries to eliminate or reduce these difficulties up to some extent. The proposed system helps the police station employees to work user friendly.

Crime File System is a system used to report, access crimes, file FIR, maintain criminal and officers records. This project is mainly useful for police stations. This system will help to manage all the activities in a police station using computers. Currently all the work is done manually, by computerizing all the activities inside a police station can be managed easily and effectively.

# 

# 

# 

# **Process Model**

According to the nature of the project and requirements. The Prototype Model is the most suitable model. In this type of model, a set of general objectives for software is defined but does not identify detailed requirements for functions and features or the developer may be unsure of the efficiency of the software, or the form that human-machine interaction should take. In a Prototyping Evolutionary Process Model, initially it begins with communication with stakeholders, the overall objectives for the software is defined, identify whatever requirements are known, and outline areas where further definition is mandatory. A prototyping iteration is planned quickly, and a quick design focusing on the representation of different aspects of the software that will be visible to end users will be made. The quick design leads to the construction of a prototype. The prototype is deployed and evaluated by stakeholders, who provide feedback that is used to further refine requirements of the software.

We have chosen this model because of the following features:

* Iteration occurs as the prototype is tuned to satisfy the needs of various stakeholders, and will lead to improved design and increased efficiency of the software.
* It will enable us to better understand the needs of the end users and what is to be done to make it better after each iteration.
* For the final software, existing program fragments can be used or apply tools to the existing prototype that will enable the working program to be generated quickly.

1. Software Requirement Specification

## **1.1** **Overall Description**

We aim to design, Crime file system. We will manage all such activities (like registration of the complaint, adding new crime, saving details of officers and criminals, updating information, search for particular viewing of the respective FIR, criminal,officer) that will save time, manpower.

This system will provide better prospective for the enhancement of organization regarding quality and transparency.

**1.1.1 Product Functions**

The main functions we focus on during the making of this software are :

1. **Login Page**

Pre-registered users should be able to login on the platform and have their details verified accordingly.

1. **Users(admin and officer) should be able to easily add criminal, officer, FIR and crime details.**

Admin should easily be able to add a new crime, new officer to the database so that the new officer can login and they can create a new FIR and be able to add a new criminal to the database.

1. **Users(admin and officer) should be able to easily modify FIR, officer, crime and criminal details.**

Admin should easily be able to modify crime, and officer details to the database. And officer should be able to easily change FIR details or criminal details.

1. **Users(admin and officer) should be able to easily search FIR, officer, crime and criminal details.**

Admin should easily be able to search any crime, and officer from the database. And officer should be able to easily search FIR details or criminal details so that he/she can refer to it or change it.

### 

### 

### **1.1.2 User Characteristics**

The proposed system tries to provide proper security and reduces manual work. This software is basically designed to add a new officer, crime, criminal, and create a new FIR. And can both modify and search these details easily from the database.

The three main types of users we focus on are-

1. User that is an admin who can add crime details and officer details to the database.
2. Our second user is police officers residing in a police station and their work includes writing a FIR/complaint and a new criminal to the system.

The following is hypothetical scenarios of individuals that might use the criminal record management and profit from it in different ways:

Dev Singh is a 32 year old inspector. He used to work in the east Delhi zone in the police department. And now he is transferring to the new south Delhi zone police station. Where he’ll give his details to the admin. And the admin adds his details to the database. Dev Singh login to the system and then use it to fill-in the complaint of a victim. And use it to search criminals, FIR.

### **1.1.3** **General Constraints**

1. The software requires a device with GUI and a keyboard(may or may not be simulated).
2. The ability of the software to interface with a database like sql must be implemented.

### **1.1.4** **Assumptions and Dependencies**

1. All Users(officers) must be pre-registered to the database by the admin. And admin will provide an officer-id to officers through which he/she can login to the system.
2. So we depend on the admin to enter new crime and officer details to the database and provide an officer-id to officers.

**1.2 External Interface Requirements**

### **1.2.1** **User Interfaces**

1. A user-friendly color scheme should be chosen.
2. The application should have its logo present on each screen once a logo has been designed.
3. Access to the internet would be necessary.
4. The UI should have well defined constraints to ensure that the software displays correctly on the screens of all compatible devices.
5. The GUI should have continuity, all screens should have the same design and layouts should be consistent.

### **1.2.2** **Hardware Interfaces**

The software may have to operate on some existing or predetermined hardware, thus imposing restrictions on the design. In order to use this system, customers or the users are required to have a device capable of connecting to the database with python and tkinter installed on it and the device should have a compatible operating system.

### **1.2.3** **Software Interfaces**

In order to use this GUI application, the officers/admin or users are required to have python and tkinter installed to their computer.

We have to use programming languages Python for backend and tkinter for designing a GUI application that is professional looking and can run on any device.

We require the XAMPP for establishing a local host to run over python idle, mysql to use it as a database and Microsoft Visual Studio IDE for programming.

## **1.3** **Functional Requirements**

### **1.3.1** **FR 1**

**Description:** Log-in process

**Input:** Input details like admin-id or officer-id and password.

**Processing:** Verifying id and password from database.

**Output:** Admin menu or Officer menu opens according to the login which the user opted for.

### **1.3.2** **FR 2**

**Description:** Admin menu

**Input:** Admin can input values like ‘add new crime’ or ‘add new officer’ or ‘modify crime’ or ’‘modify officer’.

**Processing:** Process admin choice.

**Output:** According to the option admin choose it takes the admin to that window admin choose.

### 

### 

### **1.3.3** **FR 3**

**Description:** Add new Crime

**Input:** Admin inputs values like crime name, crime type, IPC section, crime description.

**Processing:** Input values are saved in the database and Crime-id is generated.

**Output:** Generated Crime-id is displayed.

### **1.3.4** **FR 4**

**Description:** Add new Officer

**Input:** Admin inputs values like Officer name, mobile number, address, position, dob, gender, zone.

**Processing:**  Input values are saved in the database and Officer-id is generated.

**Output:** Generated Officer-id is displayed.

### **1.3.5** **FR 5**

**Description:** Modifying existing Crime details

**Input:** Admin input Crime-id of the crime he/she wants to edit and input the new updated details.

**Processing:** New input values are updated in the database.

**Output:** Values are updated.

### **1.3.6** **FR 6**

**Description:** Modifying existing Officer details

**Input:** Admin input Officer-id of the officer he/she wants to edit and input the new updated details.

**Processing:** New input values are updated in the database.

**Output:** Values are updated.

### **1.3.7** **FR 7**

**Description:** Officer menu

**Input:** Officers can input values like ‘add new FIR’ or ‘add new criminal’ or ‘search officer’ or ‘search criminal’ or ‘modify FIR’ or ‘modify criminal details’.

**Processing:** Process officer choice.

**Output:** According to the option officer choose it takes the officer to that window officer choose.

### 

### **1.3.8** **FR 8**

**Description:** Add new FIR

**Input:** Officer inputs values like investigation officer-id, date of FIR, offence type, time of offence, place of offence, date of offence, Complainant (name, mobile number, nationality, gender, dob, address), Accused (name, mobile number, address, gender).

**Processing:** Input values are saved in the database and FIR-id is generated.

**Output:** Generated FIR-id is displayed.

### **1.3.9** **FR 9**

**Description:** Add new Criminal.

**Input:** Officer inputs values like Criminal (name, crime-id, last-seen, gender, birthplace, in prison, address, photo, skin colour, height, weight, dob).

**Processing:** Input values are saved in the database and Criminal-id is generated.

**Output:** Generated Criminal-id is displayed.

### **1.3.10** **FR 10**

**Description:** Search FIR details.

**Input:** Officer input FIR-id of the FIR he/she wants to search.

**Processing:** Values of the input FIR-id are fetched from the database.

**Output:** Values related to given FIR-id are displayed.

### **1.3.11** **FR 11**

**Description:** Search Criminal details.

**Input:** Officer input Criminal-id of the Criminal he/she wants to search.

**Processing:** Values of the input Criminal-id are fetched from the database.

**Output:** Values related to given Criminal-id are displayed.

### **1.3.12** **FR 12**

**Description:** Modifying existing FIR details

**Input:** Officer input FIR-id of the FIR he/she wants to edit and input the new updated details.

**Processing:** New input values are updated in the database.

**Output:** Values are updated.

### 

### 

### **1.3.13** **FR 13**

**Description:** Modifying existing Criminal details

**Input:** Officer input Criminal-id of the Criminal he/she wants to edit and input the new updated details.

**Processing:** New input values are updated in the database.

**Output:** Values are updated.

## **1.4** **Performance Requirement**

Response time- The system will give responses within 1 second after checking the patient information and other information.

Capacity- The system must support 1000 people at a time.

Safety requirements- In case of any damage to the databases there must be a recovery method to restore the databases.

Accessibility- Software must be easily accessible to the end users at any point of time. Users must find functionality easy in order to better interact, also the web application should be easily navigable.

Backup- Software must provide a backup to the end users, in case there is some problem with the end user's current accessing device.

## **1.5** **Design Constraint**

* Database is password protected.
* System is only accessible within the criminal’s gui only.
* System is wirelessly networked with an encryption

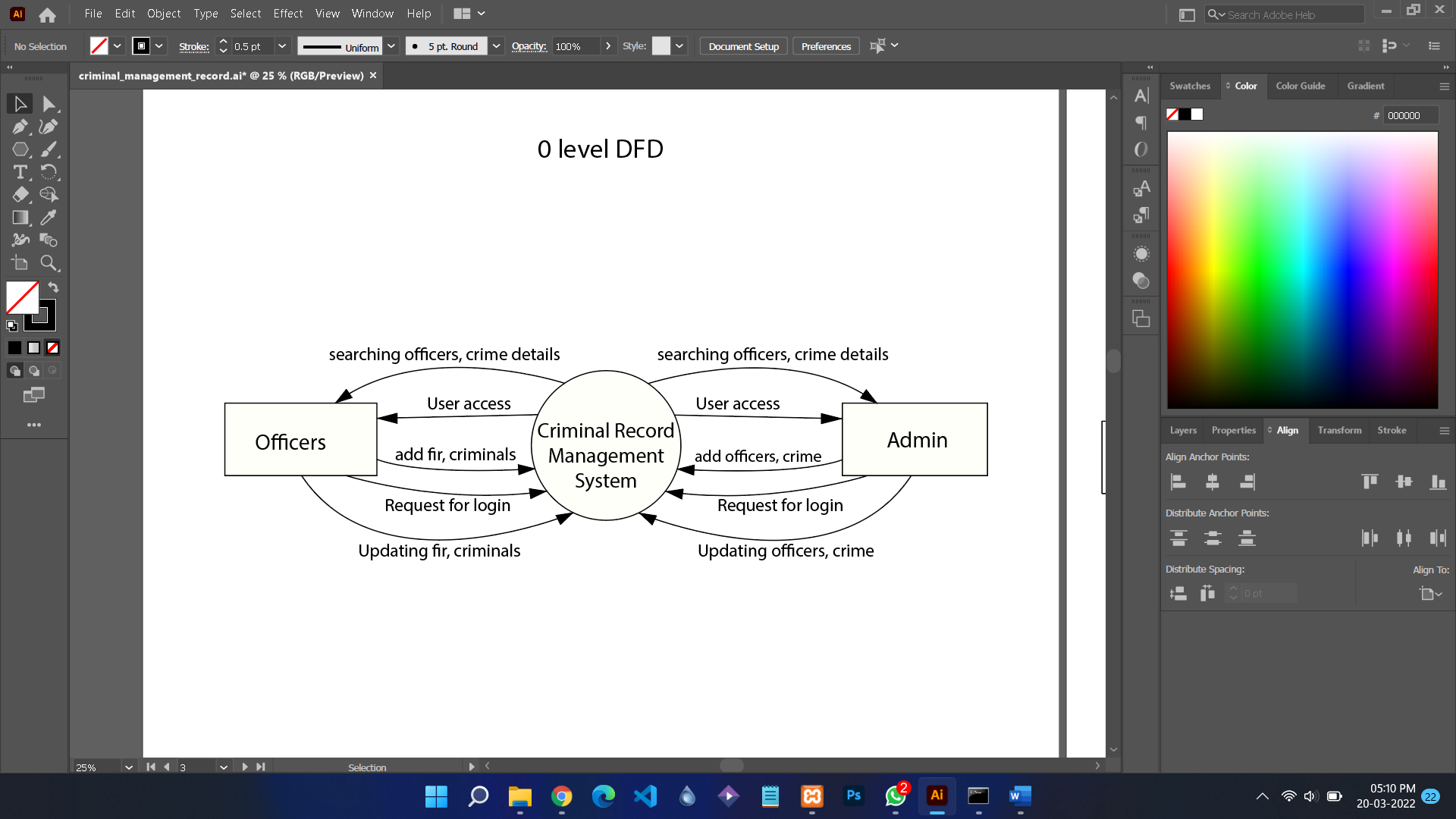
## 

## 

## 

## **1.6** **Data Flow Diagram**

## **1.6.1 Context Level DFD**

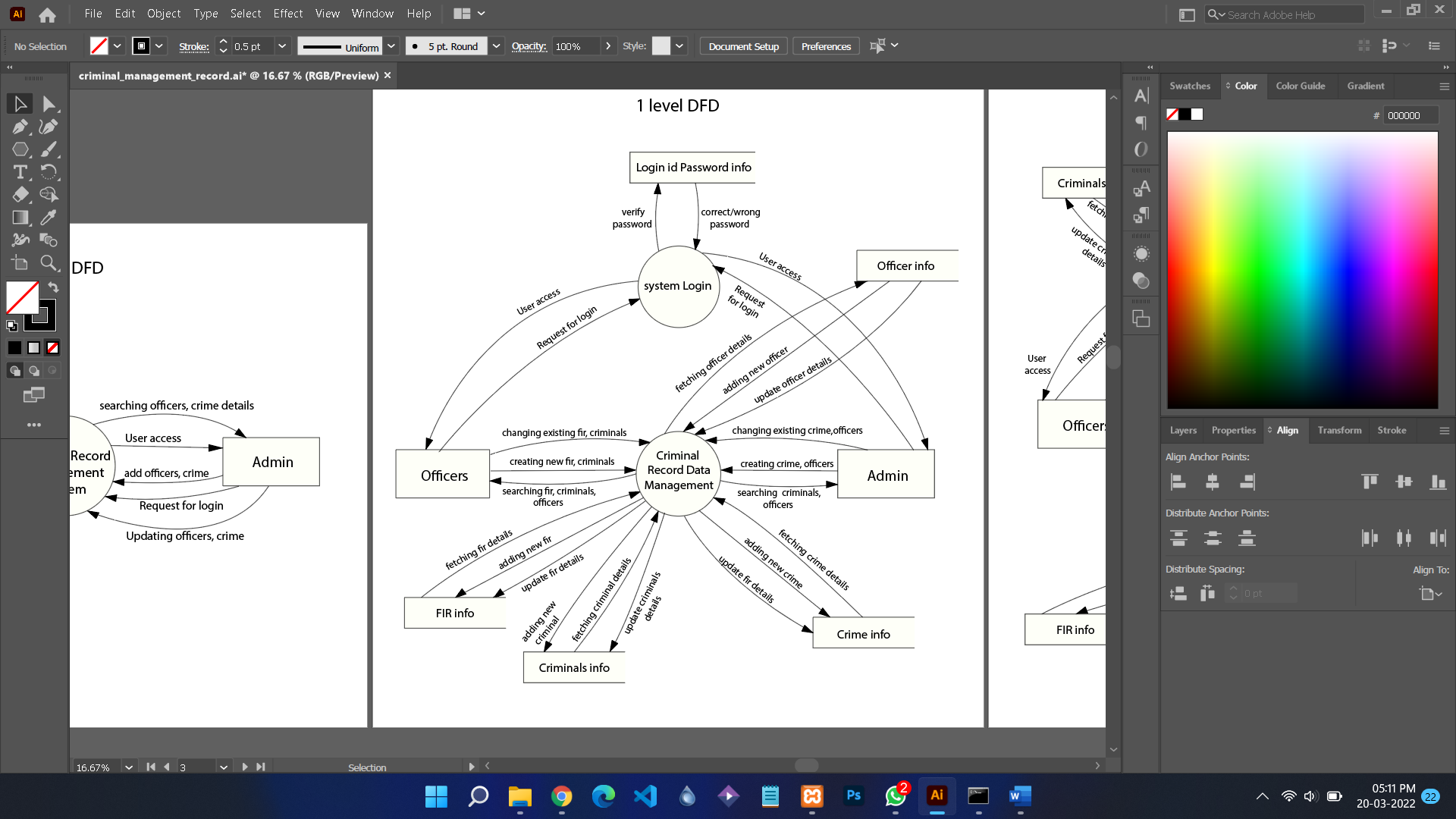


## 

## 

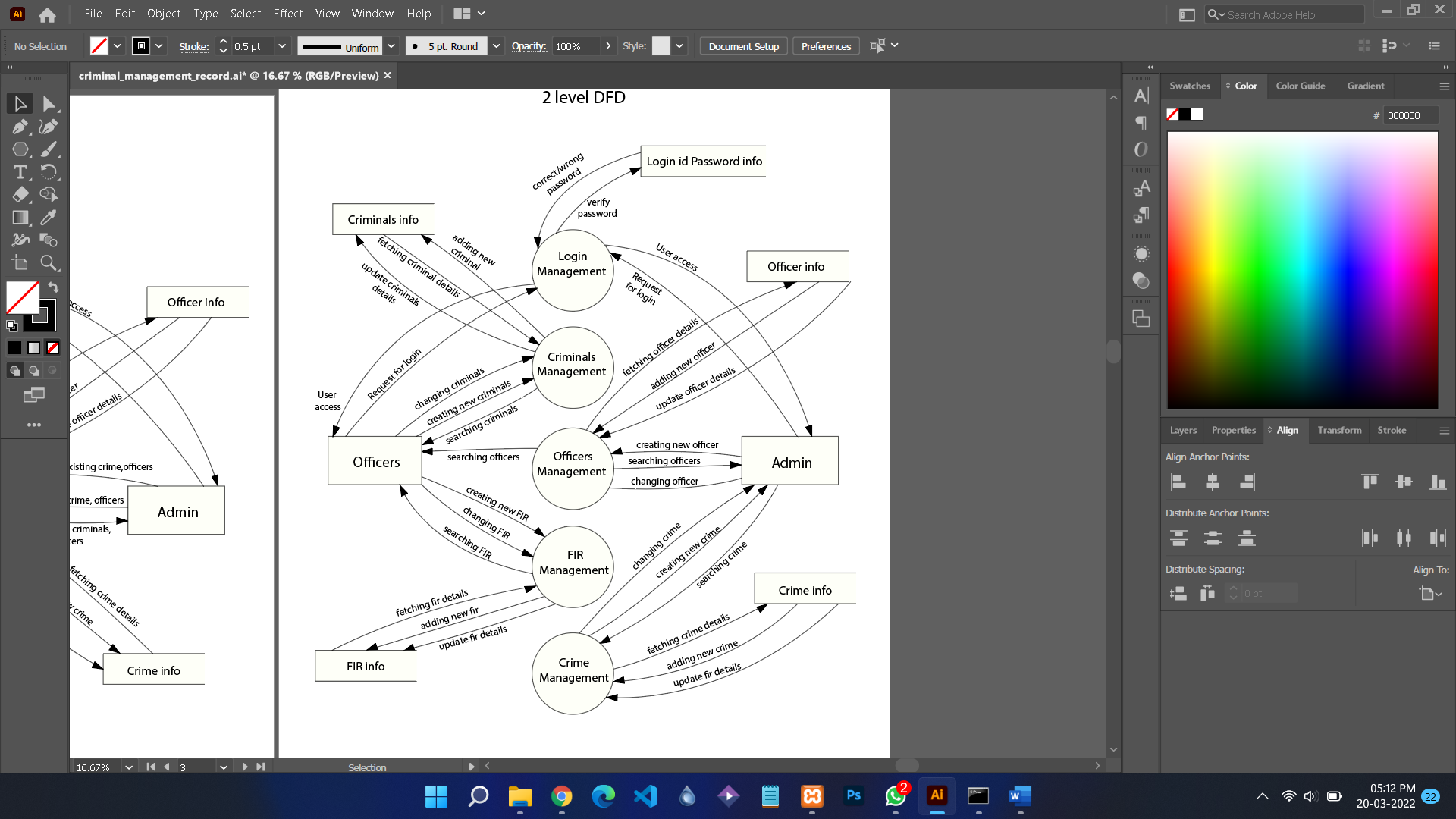
## 

## **1.6.1 1 Level DFD**



## 

## **1.6.1 2 Level DFD**



## 

## 

## 

## **1.7** **Data Dictionary**

**Login:**

| **Fields** | **Data Type** | **Size** | **Constraints** | **Description** |
| --- | --- | --- | --- | --- |
| Username | varchar | 15 | Primary key | Unique Admin or Officer id |
| Password | varchar | 12 | - | Password of admin or id |

**Crime:**

| **Fields** | **Data Type** | **Size** | **Constraints** | **Description** |
| --- | --- | --- | --- | --- |
| Crime\_id | char | 8 | Primary key | Unique Crime id |
| Crime\_name | varchar | 50 | Unique key | Name of the crime |
| Crime\_type | varchar | 40 | - | Type of the crime |
| IPC\_section | char | 20 | - | Ipc section of the crime |
| Crime\_description | text | - | Unique key | Description of the crime |

**Officer:**

| **Fields** | **Data Type** | **Size** | **Constraints** | **Description** |
| --- | --- | --- | --- | --- |
| Officer\_id | char | 8 | Primary key | Unique Officer id |
| Officer\_name | varchar | 30 | - | Full name of the officer |
| Officer\_mobile\_number | bigint | 10 | Unique key | Mobile number of the officer |
| Officer\_address | varchar | 70 | - | Permanent Address of the officer |
| Officer\_position | char | 15 | - | Position of the officer |
| Officer\_dob | date | - | - | Date of Birth of the officer |
| Officer\_gender | char | 6 | - | Gender of the officer |
| Officer\_zone | char | 20 | - | Zone at which the officer is working |

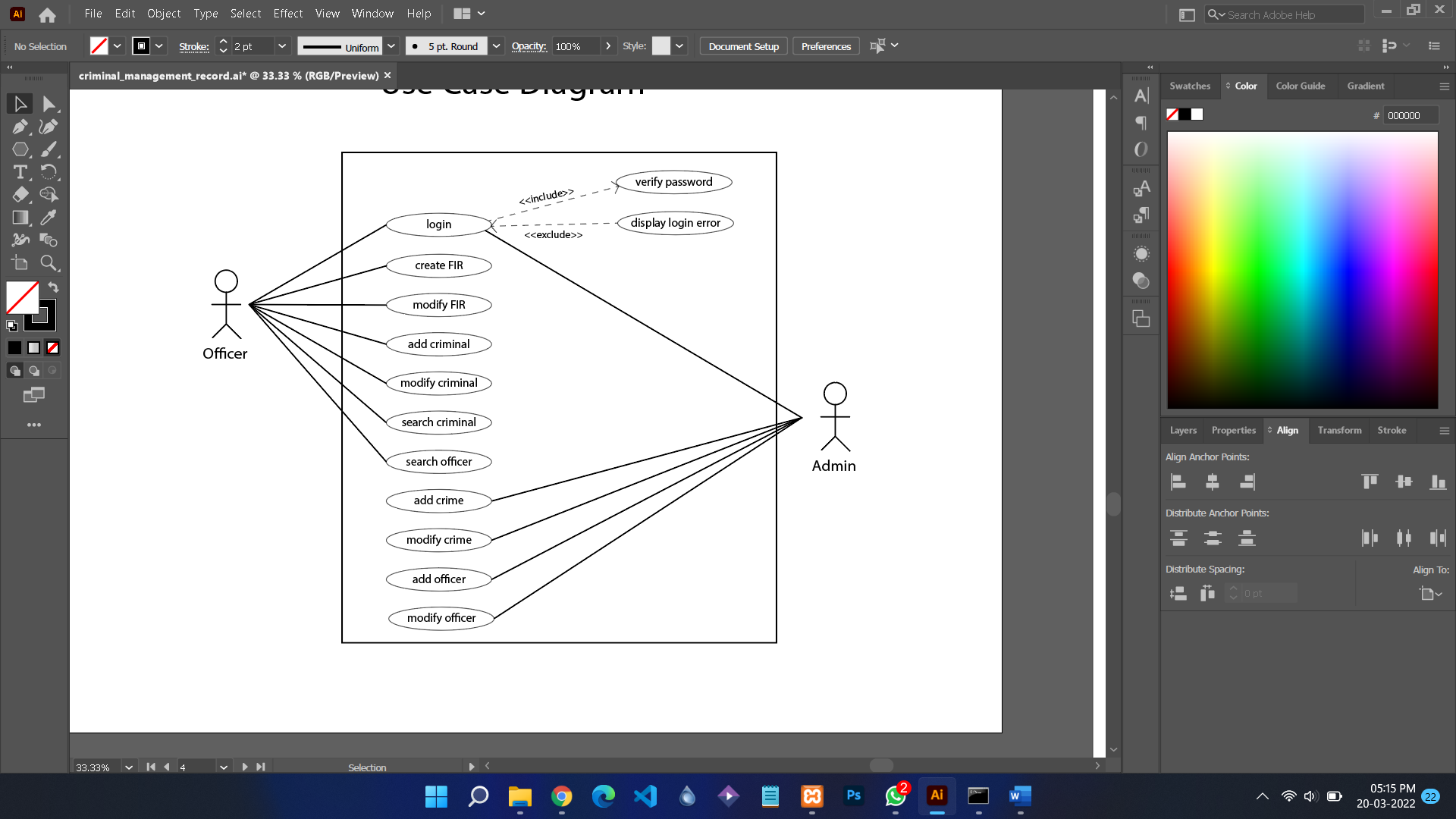
**FIR:**

| **Fields** | **Data Type** | **Size** | **Constraints** | **Description** |
| --- | --- | --- | --- | --- |
| FIR\_id | char | 10 | Primary key | Unique FIR id |
| Investigation\_officer\_id | char | 8 | Foreign key | Officer id of the investigation officer |
| Offence\_type | varchar | 40 | - | Type of offence |
| Date\_of\_FIR | date | - | - | Date on when the FIR is registered |
| Time\_of\_FIR | time | - | - | Time at when the FIR is registered |
| FIR\_description | text | - | - | Description of FIR |
| Place\_of\_offence | varchar | 70 | - | Place at where the offence took place |
| Date\_of\_offence | date | - | - | Date on when the offence took place |
| Time\_of\_offence | time | - | - | Time at when the offence took place |
| Complainant\_name | varchar | 30 | - | Complainant’s full name |
| Complainant\_mobile\_number | bigint | 10 | Unique key | Complainant’s mobile number |
| Complainant\_address | varchar | 70 | - | Complainant’s permanent address |
| Complainant\_dob | date | - | - | Complainant’s Date of Birth |
| Complainant\_gender | char | 6 | - | Complainant’s gender |
| Complainant\_nationality | char | 16 | - | Complainant ‘s nationality |
| Accused\_name | varchar | 30 | - | Full Name of the accused |
| Accused\_mobile\_number | bigint | 10 | Unique key | Mobile number of the accused |
| Accused\_address | varchar | 70 | - | Permanent address of the accused |
| Accused\_gender | char | 6 | - | Gender of the accused |
| Accused\_nationality | char | 16 | - | Nationality of the accused |

**Criminal:**

| **Fields** | **Data Type** | **Size** | **Constraints** | **Description** |
| --- | --- | --- | --- | --- |
| Criminal\_id | char | 9 | Primary key | Unique FIR id |
| Criminal\_name | varchar | 30 | - | Full Name of the criminal |
| Criminal\_crime\_id | char | 8 | Foreign key | Id of the crime that criminal did |
| Criminal\_last\_seen | varchar | 70 | - | Last place and time this criminal is seen |
| Criminal\_height\_in\_cm | decimal | (5,2) | - | Height of the criminal |
| Criminal\_weight\_in\_kg | decimal | (5,2) | - | Weight of the criminal |
| Criminal\_skin\_colour | char | 8 | - | Skin colour of the criminal |
| Criminal\_dob | date | - | - | Date of Birth of the criminal |
| Criminal\_birthplace | varchar | 50 | - | Birthplace of the criminal |
| Criminal\_gender | char | 6 | - | Gender of the criminal |
| In\_prision | char | 3 | - | Yes if the criminal is in prison right now else No |
| Photo | blob | - | - | Photo of the criminal |

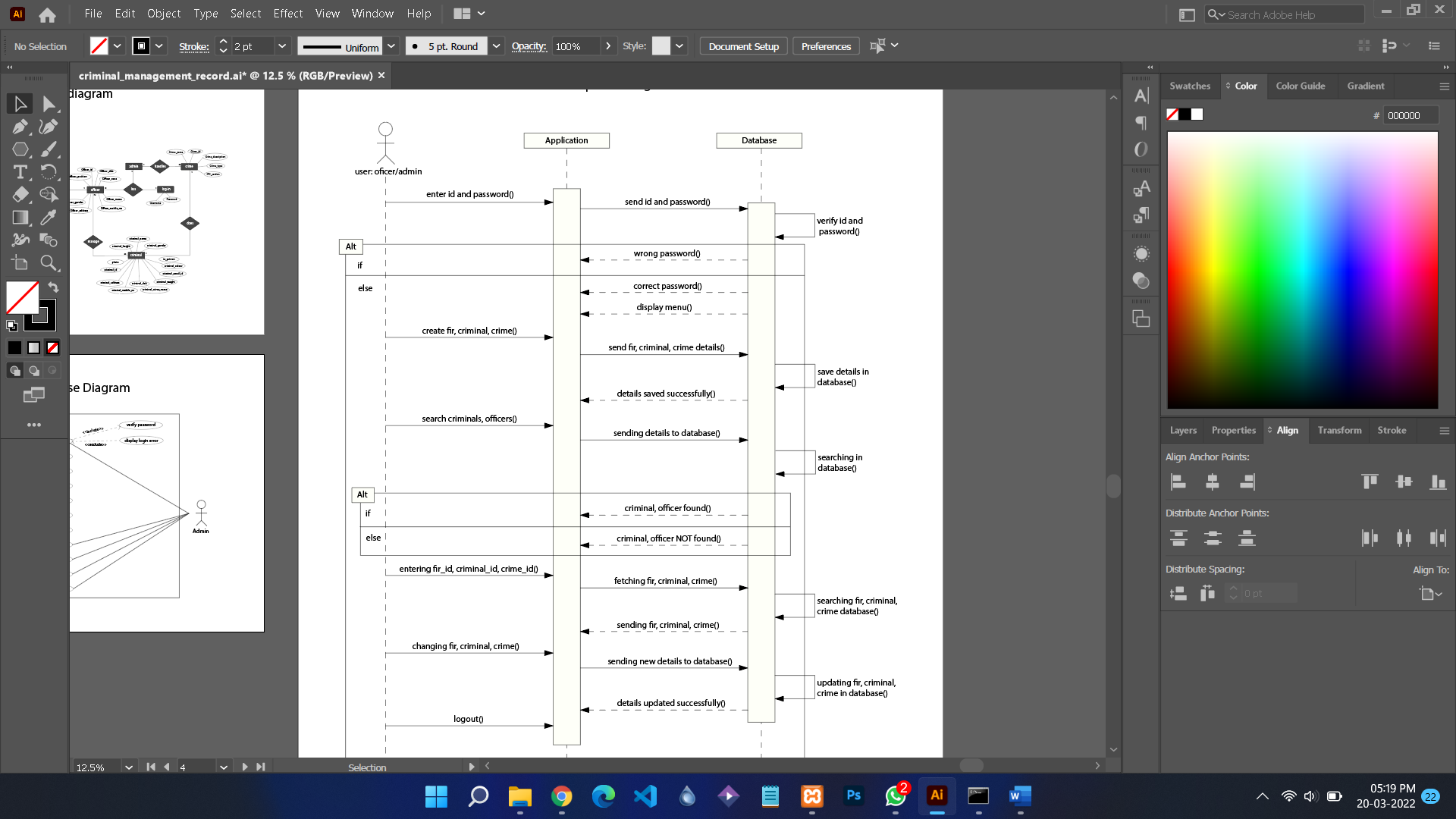
## **1.8** **Use case Diagram**



## 

## 

## **1.9** **Sequence Diagram**



# 

# 

# **2.** **Estimations**

## **2.1** **Function Points**

Given that themeasurement parameters have average complexity.

Computing the Count Total, i.e. Unadjusted Function Point Count (UFP)

| **Measurement**  **parameters** | **Count** | **Weighing factor (Average)** | **FP Count = Count x Weighing factor** |
| --- | --- | --- | --- |
| Number of User Inputs | 3 | 4 | 12 |
| Number of User Outputs | 1 | 5 | 5 |
| Number of User Inquires | 1 | 4 | 4 |
| Number of Files | 2 | 10 | 20 |
| Number of External Interfaces | 1 | 7 | 7 |

| **Count total**  **Or**  **Unadjusted Function Point(UFP)** | 45 |
| --- | --- |

The **value adjustment factor (VAF)** (also called **Complexity adjustment factor**) is calculated as follows:

**VAF = [0.65 + 0.01 \* Σ(Fi)]**

*(In the formula to calculate the Functional Point the value of Σ(Fi) (i=1-14) is calculated based on the responses to 14 questions. Each of these questions is answered (given a rating) using a scale that ranges from 0 (not important or applicable) to 5 (absolutely essential).*

*The VAF can thus vary from 0.65 - 1.35.)*

| ***Rating*** | ***0*** | ***1*** | ***2*** | ***3*** | ***4*** | ***5*** |
| --- | --- | --- | --- | --- | --- | --- |
| ***Degree of Influence*** | *Not present, or no influence*  *i.e. not important or applicable* | *Incidental influence (Simple)* | *Moderate influence*  *(Relatively Simple)* | *Average influence*  *(Average)* | *Significant influence* | *Strong influence throughout*  *absolutely essential* |

| **S.no** | **Questions** | **Value (Fi)** |
| --- | --- | --- |
| 1 | Does the system require reliable backup and recovery? |  |
| 2 | Are specialised data communications required to transfer information to or  from the application? |  |
| 3 | Are there distributed processing functions? |  |
| 4 | Is performance critical? |  |
| 5 | Will the system run in an existing, heavily utilised operational environment? |  |
| 6 | Does the system require online data entry? |  |
| 7 | Does the online data entry require the input transaction to be built over multiple screens or operations? |  |
| 8 | Are the ILFs updated online? |  |
| 9 | Are the inputs, outputs, files, or inquiries complex? |  |
| 10 | Is the internal processing complex? |  |
| 11 | Is the code designed to be reusable? |  |
| 12 | Are conversion and installation included in the design? |  |
| 13 | Is the system designed for multiple installations in different organisations? |  |
| 14 | Is the application designed to facilitate change and ease of use by the user? |  |
| **Σ(Fi)** | |  |

Given, Complexity adjustment factor = Value Adjustment Factor (VAF)

= [0.65 + 0.01 \* Σ(Fi)] = [0.65 + 0.01 \* 36]

= 1.01

Adjusted FP Count = Unadjusted FP Count 1.01× VAF

= 45 x 1.01

Hence, Adjusted FP Count or FP estimated = 45.45

## **2.2** **Efforts**

The average productivity for this kind of system = 13.5 FP/PM

Total Ffforts = FP(calculated) / average productivity

=45.45/13.5

=3.366

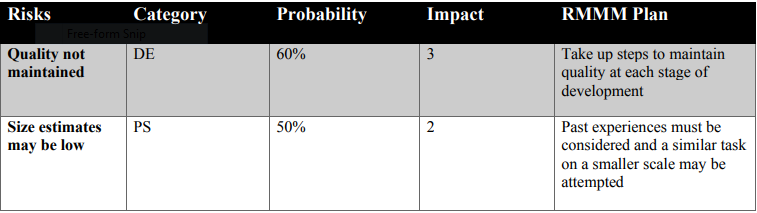
# 

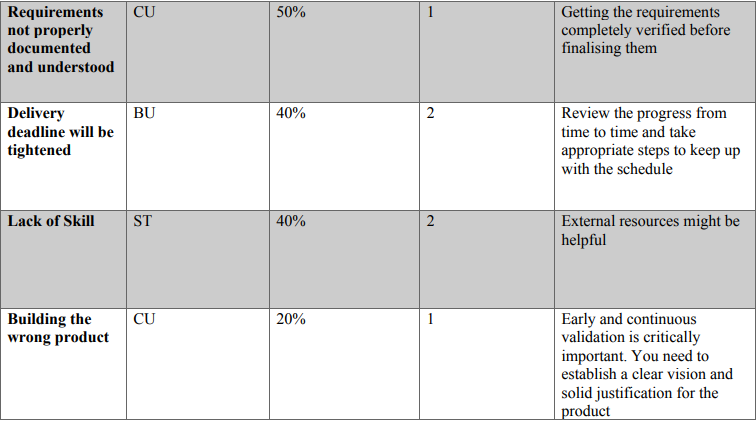
# **3.** **Scheduling**

| **WEEKS**  **WORK TASKS** | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 | Week 7 | Week 8 | Week 9 | Week 10 | Week 11 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Problem Statement |  |  |  |  |  |  |  |  |  |  |  |
| Process Model |  |  |  |  |  |  |  |  |  |  |  |
| SRS |  |  |  |  |  |  |  |  |  |  |  |
| DFD |  |  |  |  |  |  |  |  |  |  |  |
| Data Dictionary |  |  |  |  |  |  |  |  |  |  |  |
| Functional Points |  |  |  |  |  |  |  |  |  |  |  |
| Effort Estimation |  |  |  |  |  |  |  |  |  |  |  |
| Risk Analysis |  |  |  |  |  |  |  |  |  |  |  |
| System Design |  |  |  |  |  |  |  |  |  |  |  |
| Coding |  |  |  |  |  |  |  |  |  |  |  |
| Testing |  |  |  |  |  |  |  |  |  |  |  |

**4.** **Risk Management**

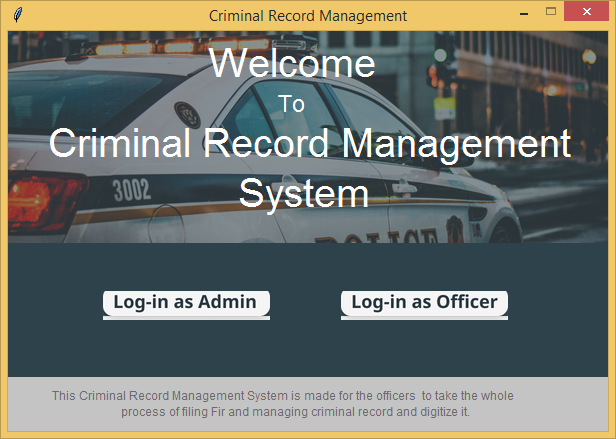
# **4.1 Risk Table**

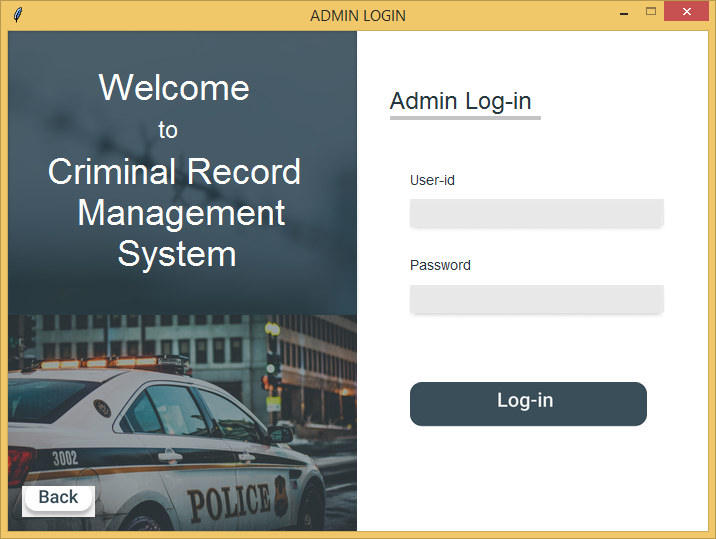
****

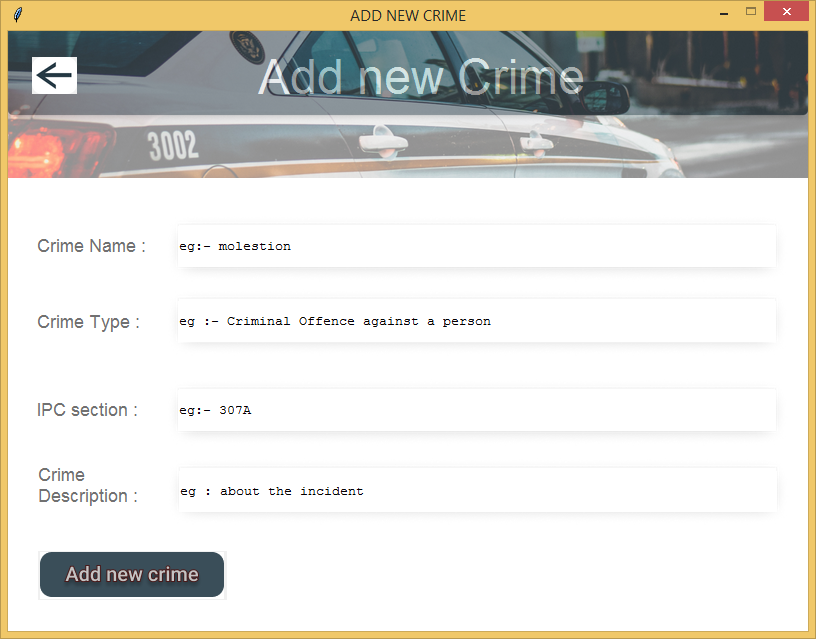
****

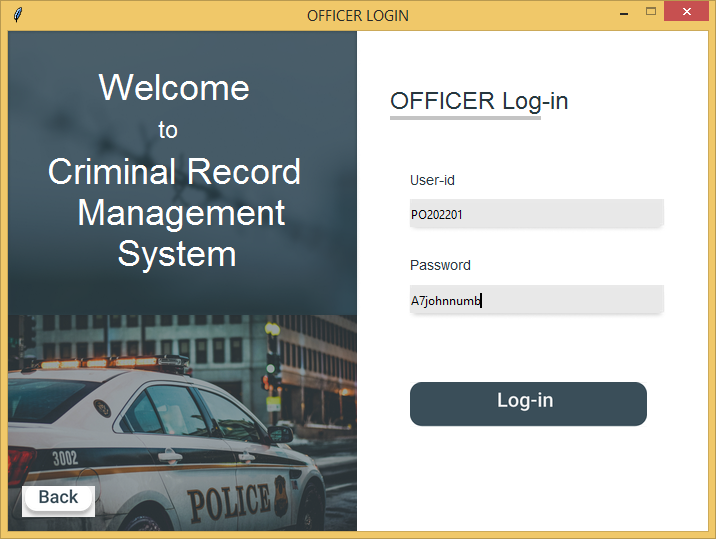
# **5.** **Design**

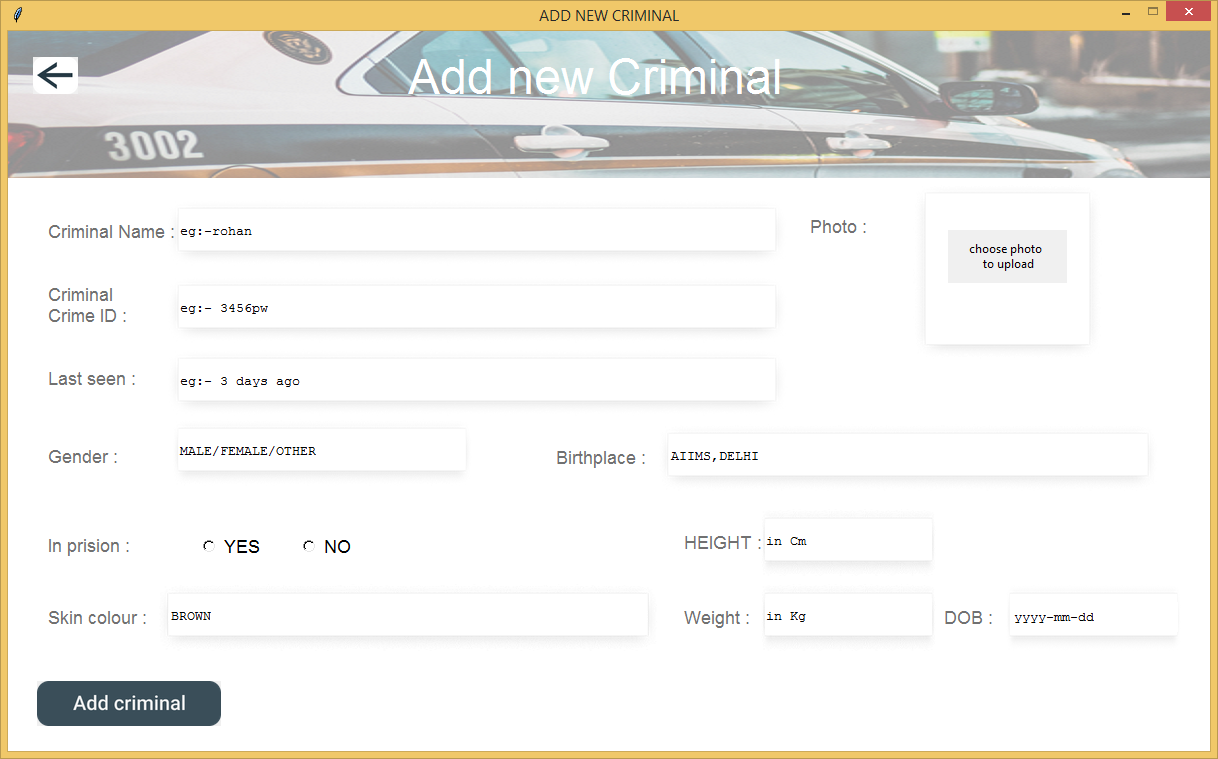
## **5.1** **Screen Design**



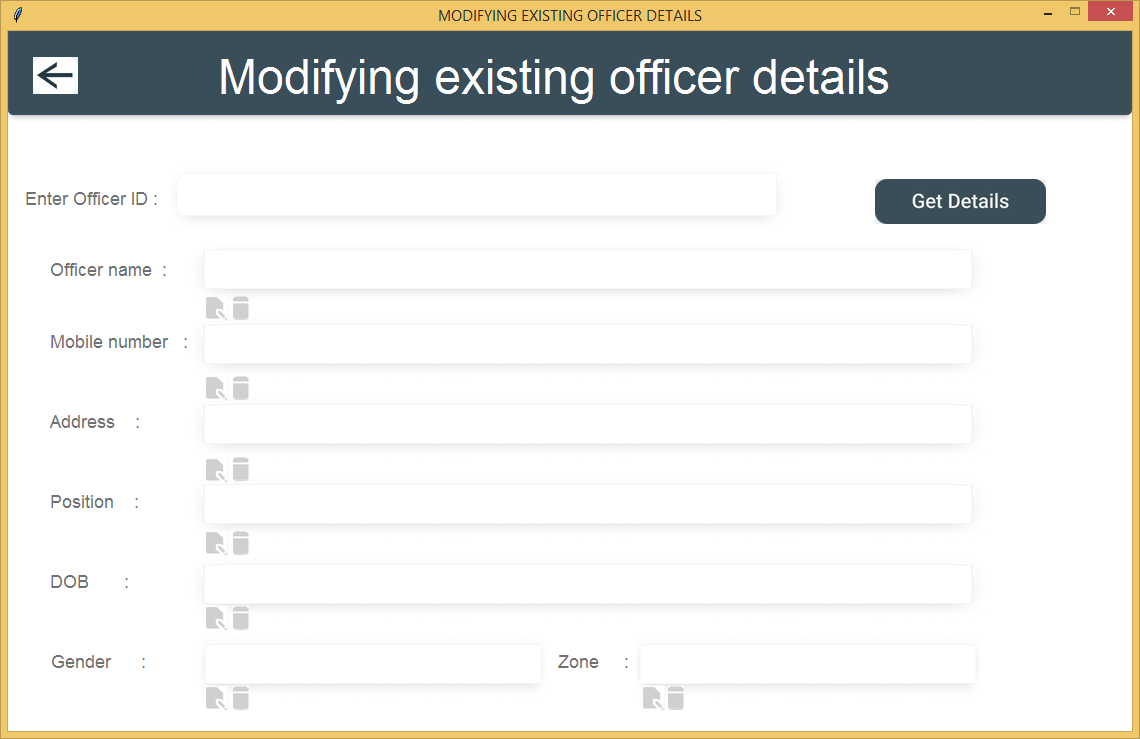


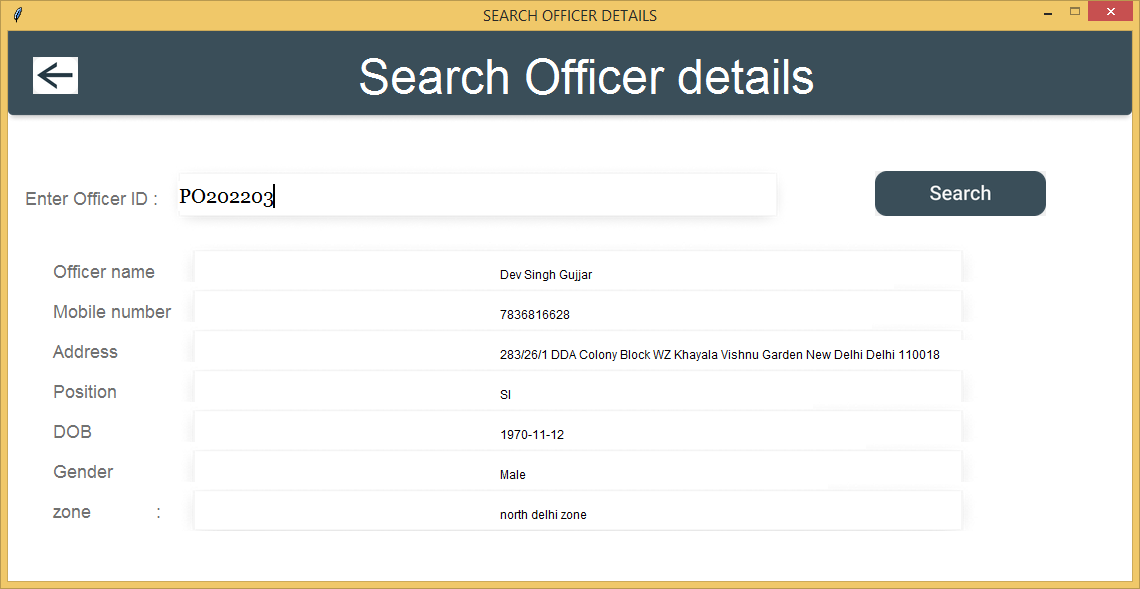
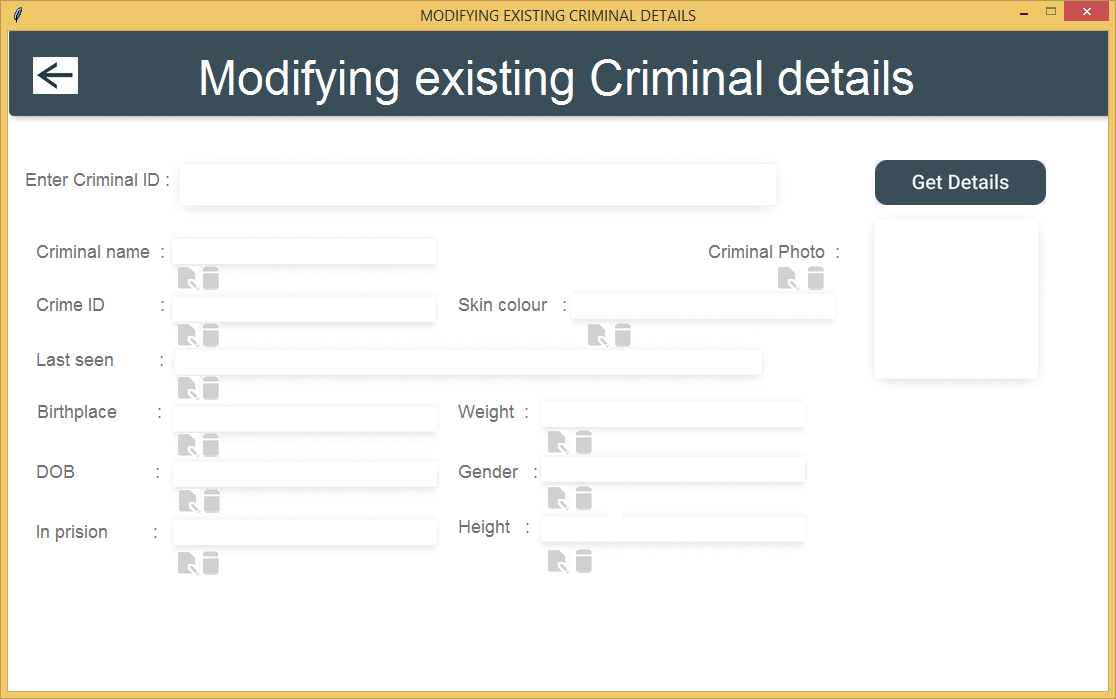


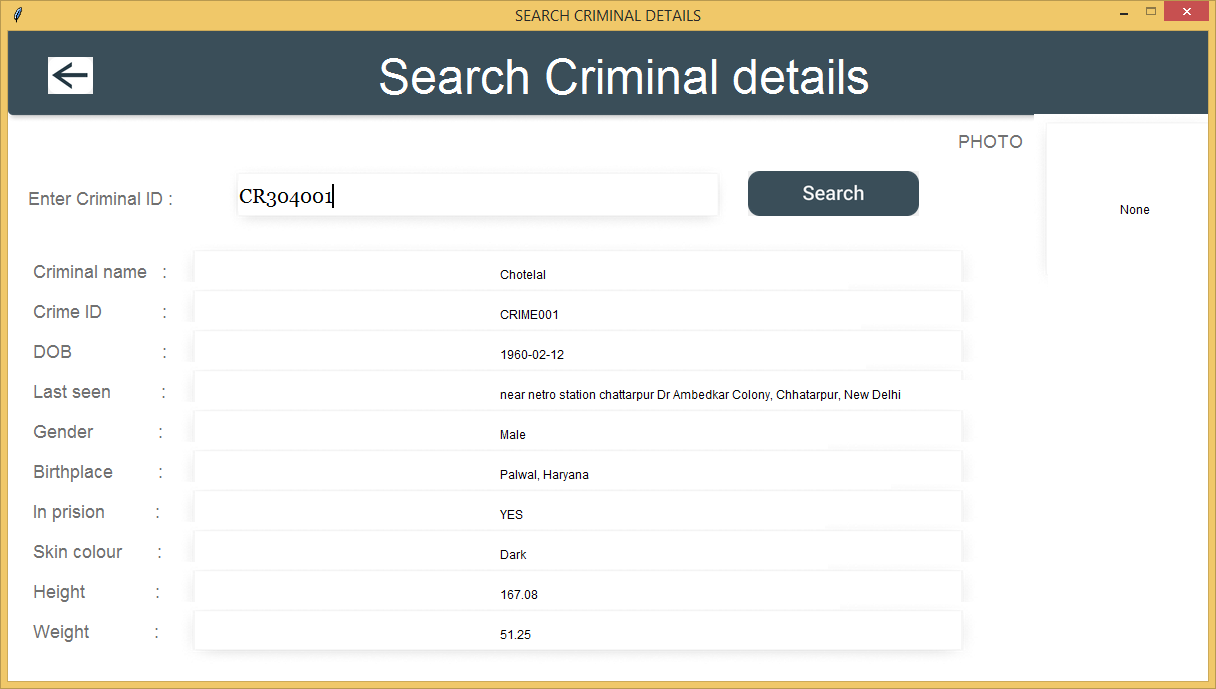


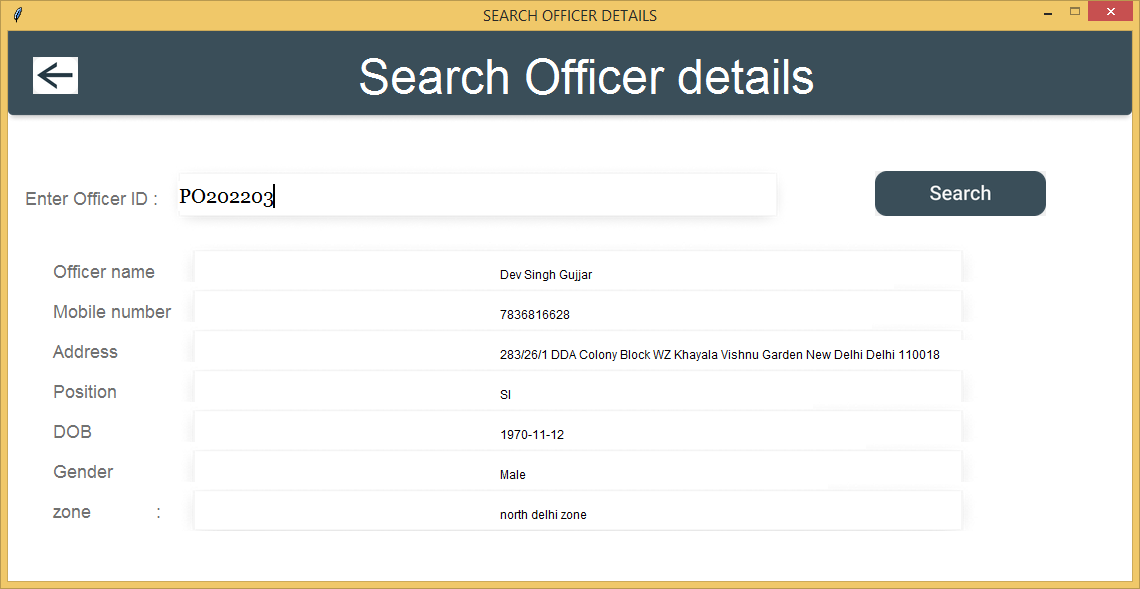








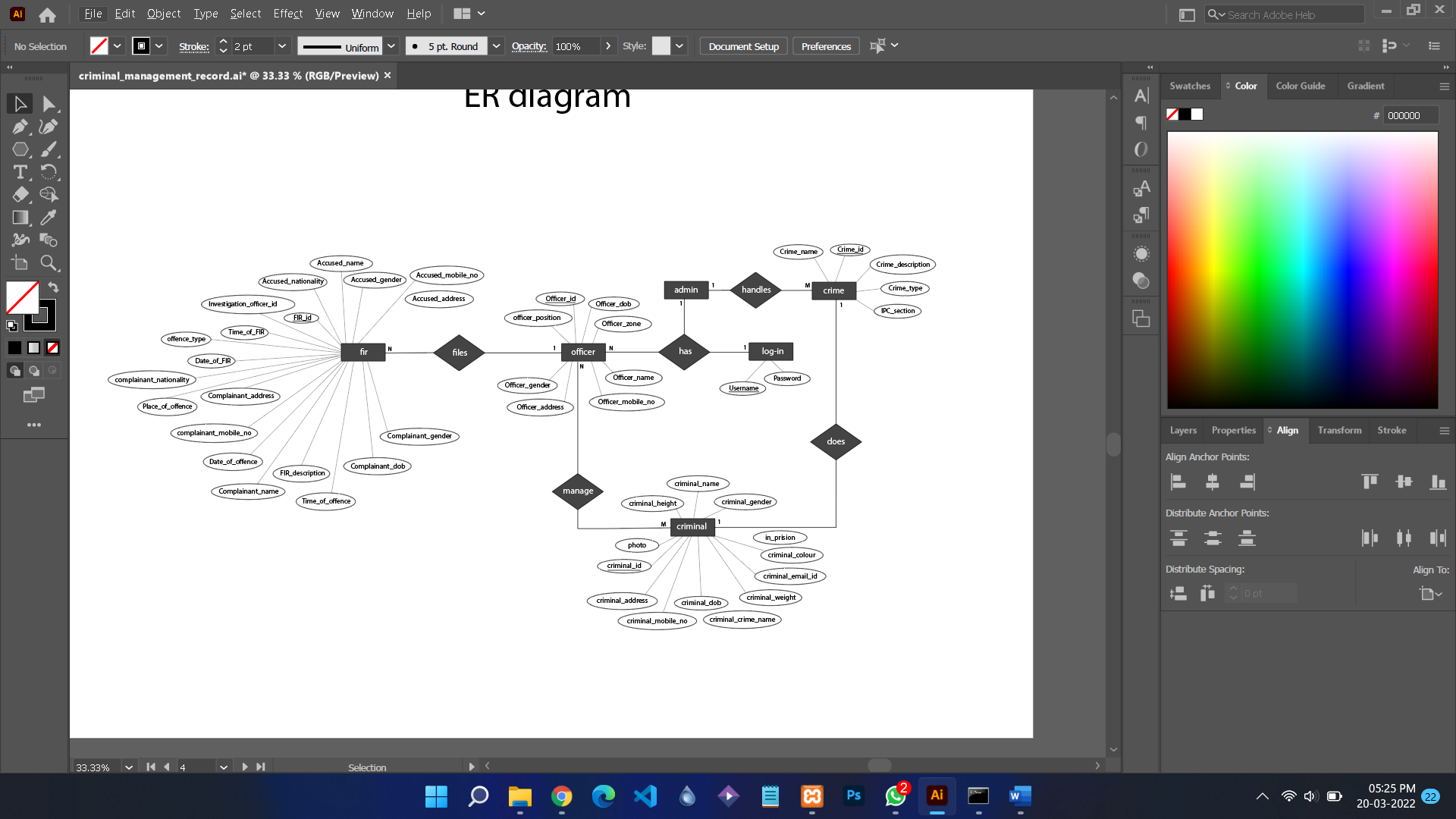




## 

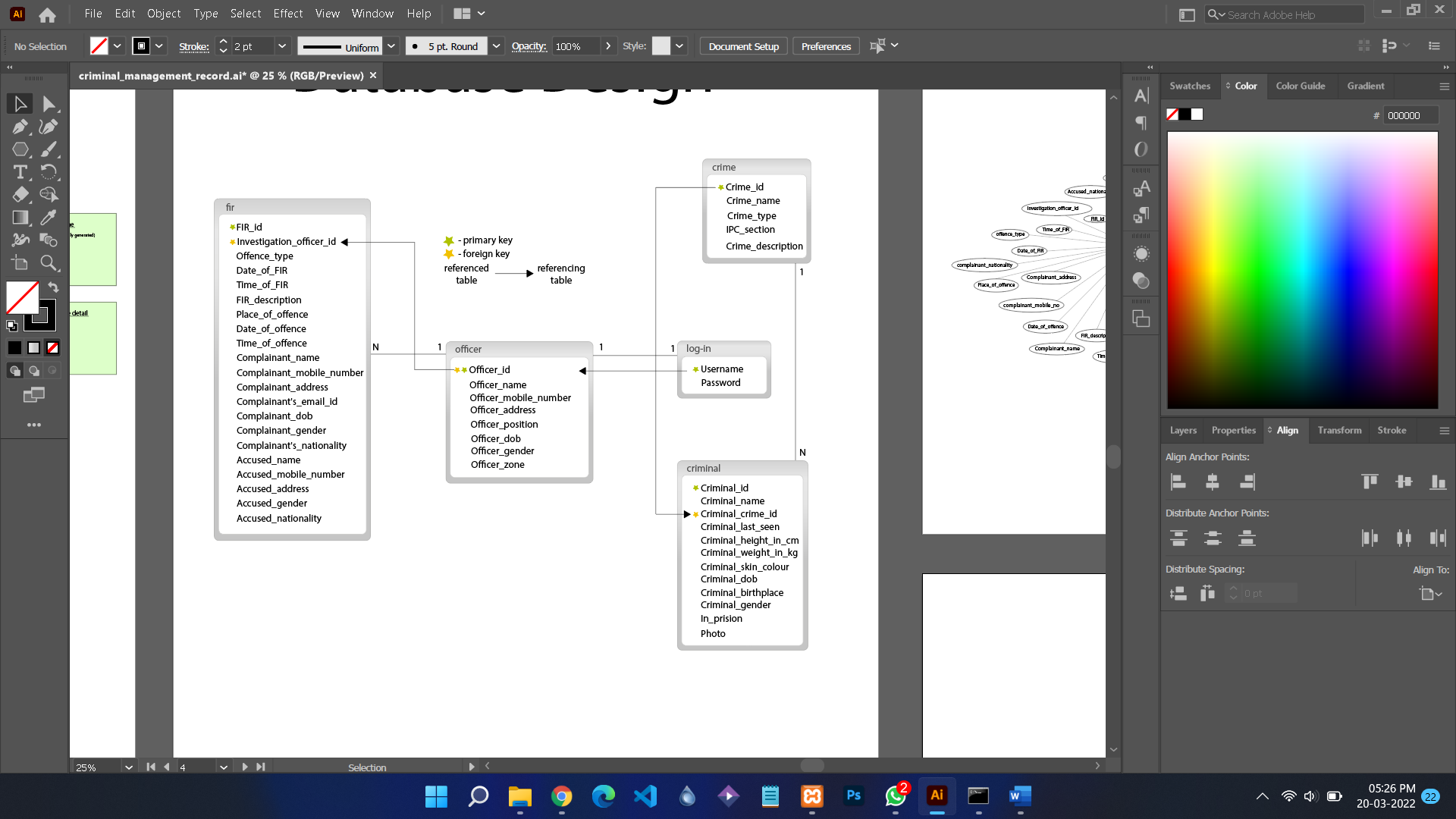
## 

## **5.2** **ER Diagram**



## 

## **5.3** **Database Design**



# 

# **6.** **Coding**

***Main window )***

from pathlib import Path

from tkinter import \*

# Explicit imports to satisfy Flake8

from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage

from database import \*

from ADMINISTRATOR\_MENU1 import \*

from OFFICER\_MENU2 import \*

from tkinter import messagebox

from ADMIN\_LOGIN14 import \*

from OFFICER\_LOGIN15 import \*

OUTPUT\_PATH = Path(\_\_file\_\_).parent

ASSETS\_PATH = OUTPUT\_PATH / Path("./assets")

def relative\_to\_assets(path: str) -> Path:

return ASSETS\_PATH / Path(path)

window = Tk()

window.geometry("600x400")

window.title("Criminal Record Management")

window.configure(bg = "#FFFFFF")

canvas = Canvas(

window,

bg = "#FFFFFF",

height = 400,

width = 600,

bd = 0,

highlightthickness = 0,

relief = "ridge"

)

canvas.place(x = 0, y = 0)

image\_image\_1 = PhotoImage(

file=relative\_to\_assets("image\_1.png"))

image\_1 = canvas.create\_image(

300.0,

196.0,

image=image\_image\_1

)

image\_image\_2 = PhotoImage(

file=relative\_to\_assets("image\_2.png"))

image\_2 = canvas.create\_image(

300.0,

106.0,

image=image\_image\_2

)

canvas.create\_text(

200.0,

10.0,

anchor="nw",

text="Welcome",

fill="#FFFFFF",

font=("Noto Sans", 30 )

)

canvas.create\_text(

270.0,

60.0,

anchor="nw",

text="To",

fill="#FFFFFF",

font=("Noto Sans", 18 )

)

canvas.create\_text(

40.0,

90.0,

anchor="nw",

text="Criminal Record Management",

fill="#FFFFFF",

font=("Noto Sans", 30 )

)

canvas.create\_text(

230.0,

140.0,

anchor="nw",

text="System",

fill="#FFFFFF",

font=("Noto Sans", 30 )

)

button\_image\_1 = PhotoImage(

file=relative\_to\_assets("button\_1.png"))

button\_1 = Button(

image=button\_image\_1,

borderwidth=0,

highlightthickness=0,

command=lambda : ID\_PASSWORD\_AS\_ADMIN(window),

relief="flat"

)

button\_1.place(

x=94.99999999999994,

y=260.0,

width=167.0,

height=29.0

)

button\_image\_2 = PhotoImage(

file=relative\_to\_assets("button\_2.png"))

button\_2 = Button(

image=button\_image\_2,

borderwidth=0,

highlightthickness=0,

command=lambda : ID\_PASSWORD\_AS\_OFFICER(window),

relief="flat"

)

button\_2.place(

x=332.99999999999994,

y=260.0,

width=167.0,

height=29.0

)

image\_image\_3 = PhotoImage(

file=relative\_to\_assets("image\_3.png"))

image\_3 = canvas.create\_image(

300.0,

373.0,

image=image\_image\_3

)

canvas.create\_text(

43.99999999999994,

357.0,

anchor="nw",

text="This Criminal Record Management System is made for the officers to take the whole ",

fill="#6E6A6A",

font=("Noto Sans", 12 \* -1)

)

canvas.create\_text(

112.99999999999994,

373.0,

anchor="nw",

text="process of filing Fir and managing criminal record and digitize it. ",

fill="#6E6A6A",

font=("Noto Sans", 12 \* -1)

)

window.resizable(False, False)

window.mainloop()

***OFFICER LOGIN)***

from pathlib import Path

from tkinter import \*

from tkinter import messagebox

# Explicit imports to satisfy Flake8

from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage

from database import \*

from OFFICER\_MENU2 import \*

OUTPUT\_PATH = Path(\_\_file\_\_).parent

ASSETS\_PATH = OUTPUT\_PATH / Path("./assets\_15")

def relative\_to\_assets(path: str) -> Path:

return ASSETS\_PATH / Path(path)

def ID\_PASSWORD\_AS\_OFFICER(window):

def back():

window\_LOGIN.destroy()

def login\_OFFICER():

if connection\_OFFICER(entry\_username.get(), entry\_password.get()) == True:

OFFICER\_MENU(window)

else:

messagebox.showerror("INVALID", "INVALID USERNAME OR PASSWORD !!\n \t TRY AGAIN ",parent = window\_LOGIN)

window\_LOGIN = Toplevel(window)

window\_LOGIN.geometry("700x500")

window\_LOGIN.title("OFFICER LOGIN")

window\_LOGIN.configure(bg = "#FFFFFF")

canvas = Canvas(

window\_LOGIN,

bg = "#FFFFFF",

height = 500,

width = 700,

bd = 0,

highlightthickness = 0,

relief = "ridge"

)

canvas.place(x = 0, y = 0)

image\_image\_1 = PhotoImage(

file=relative\_to\_assets("image\_1.png"))

image\_1 = canvas.create\_image(

174.0,

250.0,

image=image\_image\_1

)

image\_image\_2 = PhotoImage(

file=relative\_to\_assets("image\_2.png"))

image\_2 = canvas.create\_image(

174.0,

142.0,

image=image\_image\_2

)

canvas.create\_text(

90.0,

36.0,

anchor="nw",

text="Welcome ",

fill="#FFFFFF",

font=("Noto Sans", 36 \* -1)

)

canvas.create\_text(

150.0,

86.0,

anchor="nw",

text="to",

fill="#FFFFFF",

font=("RobotoRoman Light", 24 \* -1)

)

canvas.create\_text(

39.0,

79.0,

anchor="nw",

text="\nCriminal Record\n Management\n System ",

fill="#FFFFFF",

font=("Roboto", 36 \* -1)

)

image\_image\_3 = PhotoImage(

file=relative\_to\_assets("image\_3.png"))

image\_3 = canvas.create\_image(

174.0,

392.0,

image=image\_image\_3

)

button\_image\_1 = PhotoImage(

file=relative\_to\_assets("button\_1.png"))

button\_1 = Button(

window\_LOGIN,

image=button\_image\_1,

borderwidth=0,

highlightthickness=0,

command=back,

relief="flat"

)

button\_1.place(

x=14.0,

y=455.0,

width=72.51907348632812,

height=31.2620849609375

)

canvas.create\_text(

382.0,

57.0,

anchor="nw",

text="OFFICER Log-in",

fill="#26373E",

font=("RobotoRoman Medium", 24 \* -1)

)

image\_image\_4 = PhotoImage(

file=relative\_to\_assets("image\_4.png"))

image\_4 = canvas.create\_image(

457.0,

87.0,

image=image\_image\_4

)

canvas.create\_text(

402.0,

141.0,

anchor="nw",

text="User-id",

fill="#27373F",

font=("RobotoRoman Medium", 14 \* -1)

)

canvas.create\_text(

402.0,

226.0,

anchor="nw",

text="Password",

fill="#27373F",

font=("RobotoRoman Medium", 14 \* -1)

)

entry\_image\_1 = PhotoImage(

file=relative\_to\_assets("entry\_1.png"))

entry\_bg\_1 = canvas.create\_image(

529.0,

185.0,

image=entry\_image\_1

)

entry\_username = Entry(

window\_LOGIN,

bd=0,

bg="#E8E8E8",

highlightthickness=0

)

entry\_username.place(

x=402.0,

y=172.0,

width=250.0,

height=24.0

)

entry\_image\_2 = PhotoImage(

file=relative\_to\_assets("entry\_2.png"))

entry\_bg\_2 = canvas.create\_image(

529.0,

270.5,

image=entry\_image\_2

)

entry\_password = Entry(

window\_LOGIN,

bd=0,

bg="#E8E8E8",

highlightthickness=0

)

entry\_password.place(

x=402.0,

y=257.0,

width=250.0,

height=25.0

)

button\_image\_2 = PhotoImage(

file=relative\_to\_assets("button\_2.png"))

button\_2 = Button(

window\_LOGIN,

image=button\_image\_2,

borderwidth=0,

highlightthickness=0,

command=login\_OFFICER,

relief="flat"

)

button\_2.place(

x=402.03564453125,

y=350.5089111328125,

width=236.6412353515625,

height=45.16539001464844

)

window\_LOGIN.resizable(False, False)

window\_LOGIN.mainloop()

def main():

ID\_PASSWORD\_AS\_OFFICER()

if \_\_name\_\_ == '\_\_main\_\_':

main()

***ADMIN LOGIN)***

from pathlib import Path

from tkinter import \*

from tkinter import messagebox

# Explicit imports to satisfy Flake8

from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage

from database import \*

from ADMINISTRATOR\_MENU1 import \*

OUTPUT\_PATH = Path(\_\_file\_\_).parent

ASSETS\_PATH = OUTPUT\_PATH / Path("./assets\_14")

def relative\_to\_assets(path: str) -> Path:

return ASSETS\_PATH / Path(path)

def ID\_PASSWORD\_AS\_ADMIN(window):

def back():

window\_LOGIN.destroy()

def login\_ADMIN():

if connection\_ADMIN(entry\_username.get(), entry\_password.get()) == True:

ADMINISTRATION\_MENU(window)

else:

messagebox.showerror("INVAILD", "INVALID USERNAME OR PASSWORD !!\n \t TRY AGAIN ",parent = window\_LOGIN)

window\_LOGIN = Toplevel(window)

window\_LOGIN.geometry("700x500")

window\_LOGIN.title("ADMIN LOGIN")

window\_LOGIN.configure(bg = "#FFFFFF")

canvas = Canvas(

window\_LOGIN,

bg = "#FFFFFF",

height = 500,

width = 700,

bd = 0,

highlightthickness = 0,

relief = "ridge"

)

canvas.place(x = 0, y = 0)

image\_image\_1 = PhotoImage(

file=relative\_to\_assets("image\_1.png"))

image\_1 = canvas.create\_image(

174.0,

250.0,

image=image\_image\_1

)

image\_image\_2 = PhotoImage(

file=relative\_to\_assets("image\_2.png"))

image\_2 = canvas.create\_image(

174.0,

142.0,

image=image\_image\_2

)

canvas.create\_text(

90.0,

36.0,

anchor="nw",

text="Welcome ",

fill="#FFFFFF",

font=("Noto Sans", 36 \* -1)

)

canvas.create\_text(

150.0,

86.0,

anchor="nw",

text="to",

fill="#FFFFFF",

font=("RobotoRoman Light", 24 \* -1)

)

canvas.create\_text(

39.0,

79.0,

anchor="nw",

text="\nCriminal Record\n Management\n System ",

fill="#FFFFFF",

font=("Roboto", 36 \* -1)

)

image\_image\_3 = PhotoImage(

file=relative\_to\_assets("image\_3.png"))

image\_3 = canvas.create\_image(

174.0,

392.0,

image=image\_image\_3

)

button\_image\_1 = PhotoImage(

file=relative\_to\_assets("button\_1.png"))

button\_1 = Button(

window\_LOGIN,

image=button\_image\_1,

borderwidth=0,

highlightthickness=0,

command=back,

relief="flat"

)

button\_1.place(

x=14.0,

y=455.0,

width=72.51907348632812,

height=31.2620849609375

)

canvas.create\_text(

382.0,

57.0,

anchor="nw",

text="Admin Log-in",

fill="#26373E",

font=("RobotoRoman Medium", 24 \* -1)

)

image\_image\_4 = PhotoImage(

file=relative\_to\_assets("image\_4.png"))

image\_4 = canvas.create\_image(

457.0,

87.0,

image=image\_image\_4

)

canvas.create\_text(

402.0,

141.0,

anchor="nw",

text="User-id",

fill="#27373F",

font=("RobotoRoman Medium", 14 \* -1)

)

canvas.create\_text(

402.0,

226.0,

anchor="nw",

text="Password",

fill="#27373F",

font=("RobotoRoman Medium", 14 \* -1)

)

entry\_image\_1 = PhotoImage(

file=relative\_to\_assets("entry\_1.png"))

entry\_bg\_1 = canvas.create\_image(

529.0,

185.0,

image=entry\_image\_1

)

entry\_username = Entry(

window\_LOGIN,

bd=0,

bg="#E8E8E8",

highlightthickness=0

)

entry\_username.place(

x=402.0,

y=172.0,

width=250.0,

height=24.0

)

entry\_image\_2 = PhotoImage(

file=relative\_to\_assets("entry\_2.png"))

entry\_bg\_2 = canvas.create\_image(

529.0,

270.5,

image=entry\_image\_2

)

entry\_password = Entry(

window\_LOGIN,

bd=0,

bg="#E8E8E8",

highlightthickness=0

)

entry\_password.place(

x=402.0,

y=257.0,

width=250.0,

height=25.0

)

button\_image\_2 = PhotoImage(

file=relative\_to\_assets("button\_2.png"))

button\_2 = Button(

window\_LOGIN,

image=button\_image\_2,

borderwidth=0,

highlightthickness=0,

command=login\_ADMIN,

relief="flat"

)

button\_2.place(

x=402.03564453125,

y=350.5089111328125,

width=236.6412353515625,

height=45.16539001464844

)

window\_LOGIN.resizable(False, False)

window\_LOGIN.mainloop()

def main():

ID\_PASSWORD\_AS\_ADMIN()

if \_\_name\_\_ == '\_\_main\_\_':

main()

***ADD NEW CRIMINAL )***

from pathlib import Path

from tkinter import \*

from tkinter import messagebox

# Explicit imports to satisfy Flake8

from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage

from tkinter import filedialog

from database import \*

from datetime import datetime

from PIL import Image,ImageTk

OUTPUT\_PATH = Path(\_\_file\_\_).parent

ASSETS\_PATH = OUTPUT\_PATH / Path("./assets\_6")

#defining global variable

radio\_value =''

binaryData = ''

filename = ''

def relative\_to\_assets(path: str) -> Path:

return ASSETS\_PATH / Path(path)

def ADD\_NEW\_CRIMINAL(window):

def validation\_name(event):

name = entry\_CRIMINAL\_NAME.get()

if name.isalpha() and len(name)>3 and len(name)<300:

return True

elif name == '':

entry\_CRIMINAL\_NAME.delete(0,END)

entry\_CRIMINAL\_NAME.insert(0,"\*ENTER THE NAME\*")

else:

messagebox.showerror("ERROR", "WRITE A VALID NAME OR TRY DIFFERENT NAME !!!!",parent = window\_NEW\_CRIMINAL)

def validation\_gender(event):

gen = entry\_GENDER.get()

if gen.isalpha() and gen == 'male' or gen =='MALE' or gen == 'female' or gen =='FEMALE' or gen == 'other' or gen == 'OTHER' :

return True

elif gen =='':

entry\_GENDER.delete(0,END)

entry\_GENDER.insert(0,"\*ENTER THE GENDER\*")

else :

messagebox.showerror("ERROR", "WRITE VALID GENDER \n MALE/FEMALE/OTHER !!!!",parent = window\_NEW\_CRIMINAL)

def validation\_dob(event):

try:

date = entry\_DOB.get()

format = "%Y-%m-%d"

res = bool(datetime.strptime(date, format))

except:

messagebox.showerror("ERROR", "WRITE VALID DATE FORMAT yyyy-mm-dd !!!!",parent =window\_NEW\_CRIMINAL)

if res:

return True

elif entry\_DOB.get() == '':

entry\_DOB.delete(0,END)

entry\_DOB.insert(0,"\*ENTER DATE OF BIRTH\*")

else:

messagebox.showerror("ERROR", "WRITE VALID DATE FORMAT yyyy-mm-dd !!!!",parent =window\_NEW\_CRIMINAL)

def back():

window\_NEW\_CRIMINAL.destroy()

def click\_Criminal\_name(entry):

entry\_CRIMINAL\_NAME.delete(0,END)

entry\_CRIMINAL\_NAME.config(font=('Georgia 12'))

entry\_CRIMINAL\_NAME.config(validate ='focusout',

validatecommand=validation\_name)

def click\_Crime\_ID(entry):

entry\_CRIME\_ID.delete(0,END)

entry\_CRIME\_ID.config(font=('Georgia 12'))

def click\_last\_seen(entry):

entry\_LAST\_SEEN.delete(0,END)

entry\_LAST\_SEEN.config(font=('Georgia 12'))

def click\_gender(entry):

entry\_GENDER.delete(0,END)

entry\_GENDER.config(font=('Georgia 12'))

entry\_GENDER.config(validate ='focusout',

validatecommand=validation\_gender)

def click\_BP(entry):

entry\_BIRTH\_PLACE.delete(0,END)

entry\_BIRTH\_PLACE.config(font=('Georgia 12'))

def click\_height(entry):

entry\_HEIGHT.delete(0,END)

entry\_HEIGHT.config(font=('Georgia 12'))

def click\_skin\_color(entry):

entry\_SKIN\_COLOUR.delete(0,END)

entry\_SKIN\_COLOUR.config(font=('Georgia 12'))

def click\_weight(entry):

entry\_WEIGHT.delete(0,END)

entry\_WEIGHT.config(font=('Georgia 12'))

def click\_dob(entry):

entry\_DOB.delete(0,END)

entry\_DOB.config(font=('Georgia 12'))

entry\_DOB.config(validate ='focusout',

validatecommand=validation\_dob)

def radio(value):

global radio\_value

if var.get() == 1:

radio\_value = 'YES'

elif var.get() == 2:

radio\_value ='NO'

def file\_to\_binary():

global binaryData

global filename

filename = filedialog.askopenfilename(filetypes =(

("image", ".jpeg"),

("image", ".png"),

("image", ".jpg")),parent=window\_NEW\_CRIMINAL)

# Convert digital data to binary format

with open(filename, 'rb') as file:

binaryData = file.read()

def add\_new\_criminal(photo):

try:

mydb = mysql.connector.connect(user ='root',

password= 'Raman@9045',

host= 'localhost',

database='crms')

mycursor=mydb.cursor()

s="select \* from criminal"

mycursor.execute(s)

data=mycursor.fetchall()

count=mycursor.rowcount

criminal\_id=data[count-1][0]

next\_criminal\_id="CR"+str(int(criminal\_id[2:])+1)

global binaryData

c\_name=str(entry\_CRIMINAL\_NAME.get())

c\_crime\_id=str(entry\_CRIME\_ID.get())

c\_last\_seen=str(entry\_LAST\_SEEN.get())

c\_ht=(entry\_HEIGHT.get())

c\_wt=(entry\_WEIGHT.get())

c\_skin\_color=str(entry\_SKIN\_COLOUR.get())

c\_dob=str(entry\_DOB.get())

c\_birthplace=str(entry\_BIRTH\_PLACE.get())

c\_gender=str(entry\_GENDER.get())

in\_prs=radio\_value

p = binaryData

s="insert into criminal(Criminal\_id,Criminal\_name,Criminal\_crime\_id,Criminal\_last\_seen,Criminal\_height\_in\_cm,Criminal\_weight\_in\_kg,Criminal\_skin\_colour,Criminal\_dob,Criminal\_birthplace,Criminal\_gender,In\_prision,p) value ('{}','{}','{}','{}',{},{},'{}','{}','{}','{}','{}','{}')".format(next\_criminal\_id,c\_name,c\_crime\_id,c\_last\_seen,c\_ht,c\_wt,c\_skin\_color,c\_dob,c\_birthplace,c\_gender,in\_prs,p)

mycursor.execute(s)

mydb.commit()

messagebox.showinfo("OFFICER\_ID", "Successfully REGISTERED!! \nOFFICER ID IS {}".format(next\_criminal\_id),parent = window\_NEW\_CRIMINAL)

except:

messagebox.showerror("ERROR", "write valid data !!!",parent =window\_NEW\_CRIMINAL)

window\_NEW\_CRIMINAL = Toplevel(window)

window\_NEW\_CRIMINAL.geometry("1202x720")

window\_NEW\_CRIMINAL.title("ADD NEW CRIMINAL")

window\_NEW\_CRIMINAL.configure(bg = "#FFFFFF")

canvas = Canvas(

window\_NEW\_CRIMINAL,

bg = "#FFFFFF",

height = 700,

width = 1202,

bd = 0,

highlightthickness = 0,

relief = "ridge"

)

canvas.place(x = 0, y = 0)

image\_image\_1 = PhotoImage(

file=relative\_to\_assets("image\_1.png"))

image\_1 = canvas.create\_image(

601.0,

44.0,

image=image\_image\_1

)

entry\_image = PhotoImage(

file=relative\_to\_assets("image\_3.png"))

entry\_bg = canvas.create\_image(

601.0,

44.0,

image=entry\_image

)

canvas.create\_text(

400.0,

18.0,

anchor="nw",

text="Add new Criminal",

fill="#FFFFFF",

font=("RobotoRoman Bold", 48 \* -1)

)

button\_image\_1 = PhotoImage(

file=relative\_to\_assets("button\_1.png"))

button\_1 = Button(

window\_NEW\_CRIMINAL,

image=button\_image\_1,

borderwidth=0,

highlightthickness=0,

command=back,

relief="flat"

)

button\_1.place(

x=25.0,

y=26.0,

width=45.0,

height=37.0

)

canvas.create\_text(

40.0,

190.0,

anchor="nw",

text="Criminal Name :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

canvas.create\_text(

40.0,

337.0,

anchor="nw",

text="Last seen :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

canvas.create\_text(

40.0,

576.0,

anchor="nw",

text="Skin colour :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

canvas.create\_text(

676.0,

576.0,

anchor="nw",

text="Weight :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

canvas.create\_text(

936.0,

576.0,

anchor="nw",

text="DOB :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

canvas.create\_text(

676.0,

501.0,

anchor="nw",

text="HEIGHT :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

canvas.create\_text(

548.0,

416.0,

anchor="nw",

text="Birthplace :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

canvas.create\_text(

802.0,

185.0,

anchor="nw",

text="Photo :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

canvas.create\_text(

40.0,

253.0,

anchor="nw",

text="Criminal\nCrime ID :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

canvas.create\_text(

40.0,

415.0,

anchor="nw",

text="Gender :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

canvas.create\_text(

40.0,

504.0,

anchor="nw",

text="In prision :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

image\_image\_2 = PhotoImage(

file=relative\_to\_assets("image\_2.png")) ## IMAGE OF CRIMINAL

image\_2 = canvas.create\_image(

999.0,

240.0,

image=image\_image\_2

)

button\_image\_2 = PhotoImage(

file=relative\_to\_assets("button\_2.png"))

button\_2 = Button(

window\_NEW\_CRIMINAL,

image=button\_image\_2,

borderwidth=0,

highlightthickness=0,

command=lambda:add\_new\_criminal(binaryData),

relief="flat"

)

button\_2.place(

x=29.0,

y=650.0,

width=184.0,

height=45.0

)

photo\_button = Button(

window\_NEW\_CRIMINAL,

text = "choose photo \n to upload",

borderwidth=0,

highlightthickness=0,

command=file\_to\_binary,

relief="ridge" ,

padx =20,pady = 10)

photo\_button.place(x= 940,y =199 )

entry\_image\_2 = PhotoImage(

file=relative\_to\_assets("entry\_2.png"))

entry\_bg\_2 = canvas.create\_image(

469.0,

200.5,

image=entry\_image\_2

)

entry\_CRIMINAL\_NAME = Entry(

window\_NEW\_CRIMINAL,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Courier 10')

)

entry\_CRIMINAL\_NAME.place(

x=171.0,

y=185.0,

width=589.0,

height=30.0

)

entry\_CRIMINAL\_NAME.insert(0,"eg:-rohan ")

entry\_CRIMINAL\_NAME.bind("<Button-1>",click\_Criminal\_name)

entry\_CRIMINAL\_NAME.bind('<FocusOut>',validation\_name)

entry\_image\_3 = PhotoImage(

file=relative\_to\_assets("entry\_3.png"))

entry\_bg\_3 = canvas.create\_image(

469.0,

277.5,

image=entry\_image\_3

)

entry\_CRIME\_ID = Entry(

window\_NEW\_CRIMINAL,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Courier 10')

)

entry\_CRIME\_ID.place(

x=171.0,

y=262.0,

width=589.0,

height=30.0

)

entry\_CRIME\_ID.insert(0,"eg:- 3456pw")

entry\_CRIME\_ID.bind("<Button-1>",click\_Crime\_ID)

entry\_image\_4 = PhotoImage(

file=relative\_to\_assets("entry\_4.png"))

entry\_bg\_4 = canvas.create\_image(

469.0,

350.5,

image=entry\_image\_4

)

entry\_LAST\_SEEN = Entry(

window\_NEW\_CRIMINAL,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Courier 10')

)

entry\_LAST\_SEEN.place(

x=171.0,

y=335.0,

width=589.0,

height=30.0

)

entry\_LAST\_SEEN.insert(0,"eg:- 3 days ago ")

entry\_LAST\_SEEN.bind("<Button-1>",click\_last\_seen)

entry\_image\_5 = PhotoImage(

file=relative\_to\_assets("entry\_5.png"))

entry\_bg\_5 = canvas.create\_image(

314.0,

420.5,

image=entry\_image\_5

)

entry\_GENDER = Entry(

window\_NEW\_CRIMINAL,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Courier 10')

)

entry\_GENDER.place(

x=171.0,

y=405.0,

width=270.0,

height=30.0

)

entry\_GENDER.insert(0,"MALE/FEMALE/OTHER ")

entry\_GENDER.bind("<Button-1>",click\_gender)

entry\_GENDER.bind('<FocusOut>',validation\_gender)

entry\_image\_6 = PhotoImage(

file=relative\_to\_assets("entry\_6.png"))

entry\_bg\_6 = canvas.create\_image(

900.0,

425.5,

image=entry\_image\_6

)

entry\_BIRTH\_PLACE = Entry(

window\_NEW\_CRIMINAL,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Courier 10')

)

entry\_BIRTH\_PLACE.place(

x=662.0,

y=410.0,

width=460.0,

height=30.0

)

entry\_BIRTH\_PLACE.insert(0,"AIIMS,DELHI")

entry\_BIRTH\_PLACE.bind("<Button-1>",click\_BP)

var = IntVar()

R1 = Radiobutton(window\_NEW\_CRIMINAL, text="YES", variable=var, bg="#FFFFFF",value=1,font=("Noto Sans", 18 \* -1),command = lambda : radio(var.get))

R1.place(

x=190,

y=500)

R2 = Radiobutton(window\_NEW\_CRIMINAL, text="NO", variable=var, value=2,bg="#FFFFFF",font=("Noto Sans", 18 \* -1),command = lambda: radio(var.get))

R2.place(

x=290,

y=500)

entry\_image\_7 = PhotoImage(

file=relative\_to\_assets("entry\_11.png"))

entry\_bg\_7 = canvas.create\_image(

840.0,

510.5,

image=entry\_image\_7

)

entry\_HEIGHT = Entry(

window\_NEW\_CRIMINAL,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Courier 10')

)

entry\_HEIGHT.place(

x=757.0,

y=495.0,

width=165.0,

height=30.0

)

entry\_HEIGHT.insert(0,"in Cm ")

entry\_HEIGHT.bind("<Button-1>",click\_height)

entry\_image\_8 = PhotoImage(

file=relative\_to\_assets("entry\_8.png"))

entry\_bg\_8 = canvas.create\_image(

400.0,

585.5,

image=entry\_image\_8

)

entry\_SKIN\_COLOUR = Entry(

window\_NEW\_CRIMINAL,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Courier 10')

)

entry\_SKIN\_COLOUR.place(

x=162.0,

y=570.0,

width=460.0,

height=30.0

)

entry\_SKIN\_COLOUR.insert(0,"BROWN ")

entry\_SKIN\_COLOUR.bind("<Button-1>",click\_skin\_color)

entry\_image\_9 = PhotoImage(

file=relative\_to\_assets("entry\_9.png"))

entry\_bg\_9 = canvas.create\_image(

840.0,

585.5,

image=entry\_image\_9

)

entry\_WEIGHT = Entry(

window\_NEW\_CRIMINAL,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Courier 10')

)

entry\_WEIGHT.place(

x=757.0,

y=570.0,

width=165.0,

height=30.0

)

entry\_WEIGHT.insert(0,"in Kg ")

entry\_WEIGHT.bind("<Button-1>",click\_weight)

entry\_image\_10 = PhotoImage(

file=relative\_to\_assets("entry\_10.png"))

entry\_bg\_10 = canvas.create\_image(

1085.0,

585.5,

image=entry\_image\_10

)

entry\_DOB = Entry(

window\_NEW\_CRIMINAL,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Courier 10')

)

entry\_DOB.place(

x=1005.0,

y=570.5,

width=165.0,

height=30.0

)

entry\_DOB.insert(0,"yyyy-mm-dd ")

entry\_DOB.bind("<Button-1>",click\_dob)

entry\_DOB.bind('<FocusOut>',validation\_dob)

window\_NEW\_CRIMINAL.resizable(False, False)

window\_NEW\_CRIMINAL.mainloop()

def main():

ADD\_NEW\_CRIMINAL()

if \_\_name\_\_ == '\_\_main\_\_':

main()

***MODIFYING OFFICER DETAILS)***

from pathlib import Path

from tkinter import \*

# Explicit imports to satisfy Flake8

from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage

from ADMINISTRATOR\_MENU1 import \*

from database import \*

OUTPUT\_PATH = Path(\_\_file\_\_).parent

ASSETS\_PATH = OUTPUT\_PATH / Path("./assets\_9")

def relative\_to\_assets(path: str) -> Path:

return ASSETS\_PATH / Path(path)

def MODIFY\_EXISTING\_OFFICER\_DETAILS(window):

def back():

window\_MODIFY\_OFFICER\_DETAILS.destroy()

def search\_officer():

mydb = mysql.connector.connect(user ='root',

password= 'Raman@9045',

host= 'localhost',

database='crms')

mycursor = mydb.cursor()

input\_o\_id=entry\_OFFICER\_ID.get()

s="select \* from officer where officer\_id = '{}'".format(input\_o\_id)

mycursor.execute(s)

fetch\_officer = mycursor.fetchone()

var\_OFN.set(fetch\_officer[1])

var\_OMN.set(fetch\_officer[2])

var\_OADD.set(fetch\_officer[3])

var\_POS.set(fetch\_officer[4])

var\_ODOB.set(fetch\_officer[5])

var\_OGEN.set(fetch\_officer[6])

var\_ZONE.set(fetch\_officer[7])

def button\_modify\_OFFICER\_NAME():

def modify\_OFN(event):

mydb = mysql.connector.connect(user ='root',

password= 'Raman@9045',

host= 'localhost',

database='crms')

mycursor=mydb.cursor()

input\_c\_id=entry\_OFFICER\_ID.get()

s="select \* from officer where officer\_id='{}'".format(input\_c\_id)

mycursor.execute(s)

fetch\_officer=mycursor.fetchone()

if entry\_OFFICER\_NAME.get() == '':

entry\_OFFICER\_NAME.insert(fetch\_officer[1])

else :

s="update officer set officer\_name = '{}' where officer\_id = '{}'".format(entry\_OFFICER\_NAME.get(),entry\_OFFICER\_ID.get())

mycursor.execute(s)

mydb.commit()

entry\_OFFICER\_NAME = Entry(

window\_MODIFY\_OFFICER\_DETAILS,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Georgia 15')

)

entry\_OFFICER\_NAME.place(

x=198.0,

y=225.0,

width=750.0,

height=30.0

)

entry\_OFFICER\_NAME.bind('<Return>',modify\_OFN)

def button\_modify\_MOBILE\_NUMBER():

def modify\_OMN(event):

mydb = mysql.connector.connect(user ='root',

password= 'Raman@9045',

host= 'localhost',

database='crms')

mycursor=mydb.cursor()

input\_c\_id=entry\_OFFICER\_ID.get()

s="select \* from officer where officer\_id='{}'".format(input\_c\_id)

mycursor.execute(s)

fetch\_officer=mycursor.fetchone()

if entry\_MOBILE\_NUMBER.get() == '':

entry\_MOBILE\_NUMBER.insert(fetch\_officer[2])

else :

s="update officer set officer\_mobile\_number = '{}' where officer\_id = '{}'".format(entry\_MOBILE\_NUMBER.get(),entry\_OFFICER\_ID.get())

mycursor.execute(s)

mydb.commit()

entry\_MOBILE\_NUMBER = Entry(

window\_MODIFY\_OFFICER\_DETAILS,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Georgia 15')

)

entry\_MOBILE\_NUMBER.place(

x=198.0,

y=300.0,

width=750.0,

height=30.0

)

entry\_MOBILE\_NUMBER.bind('<Return>',modify\_OMN)

def button\_modify\_ADDRESS():

def modify\_OADD(event):

mydb = mysql.connector.connect(user ='root',

password= 'Raman@9045',

host= 'localhost',

database='crms')

mycursor=mydb.cursor()

input\_c\_id=entry\_OFFICER\_ID.get()

s="select \* from officer where officer\_id='{}'".format(input\_c\_id)

mycursor.execute(s)

fetch\_officer=mycursor.fetchone()

if entry\_ADDRESS.get() == '':

entry\_ADDRESS.insert(fetch\_officer[3])

else :

s="update officer set officer\_address = '{}' where officer\_id = '{}'".format(entry\_ADDRESS.get(),entry\_OFFICER\_ID.get())

mycursor.execute(s)

entry\_ADDRESS = Entry(

window\_MODIFY\_OFFICER\_DETAILS,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Georgia 15')

)

entry\_ADDRESS.place(

x=198.0,

y=380.0,

width=750.0,

height=30.0

)

entry\_ADDRESS.bind('<Return>',modify\_OADD)

def button\_modify\_POSITION():

def modify\_POS(event):

mydb = mysql.connector.connect(user ='root',

password= 'Raman@9045',

host= 'localhost',

database='crms')

mycursor=mydb.cursor()

input\_c\_id=entry\_OFFICER\_ID.get()

s="select \* from officer where officer\_id='{}'".format(input\_c\_id)

mycursor.execute(s)

fetch\_officer=mycursor.fetchone()

if entry\_POSITION.get() == '':

entry\_POSITION.insert(fetch\_officer[4])

else :

s="update officer set officer\_position = '{}' where officer\_id = '{}'".format(entry\_POSITION.get(),entry\_OFFICER\_ID.get())

mycursor.execute(s)

mydb.commit()

entry\_POSITION = Entry(

window\_MODIFY\_OFFICER\_DETAILS,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Georgia 15')

)

entry\_POSITION.place(

x=198.0,

y=460.0,

width=750.0,

height=30.0

)

entry\_POSITION.bind('<Return>',modify\_POS)

def button\_modify\_DOB():

def modify\_DOB(event):

mydb = mysql.connector.connect(user ='root',

password= 'Raman@9045',

host= 'localhost',

database='crms')

mycursor=mydb.cursor()

input\_c\_id=entry\_OFFICER\_ID.get()

s="select \* from officer where officer\_id='{}'".format(input\_c\_id)

mycursor.execute(s)

fetch\_officer=mycursor.fetchone()

if entry\_DOB.get() == '':

entry\_DOB.insert(fetch\_officer[5])

else :

s="update officer set officer\_dob = '{}' where officer\_id = '{}'".format(entry\_DOB.get(),entry\_OFFICER\_ID.get())

mycursor.execute(s)

mydb.commit()

entry\_DOB = Entry(

window\_MODIFY\_OFFICER\_DETAILS,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Georgia 15')

)

entry\_DOB.place(

x=198.0,

y=540.0,

width=750.0,

height=30.0

)

entry\_DOB.bind('<Return>',modify\_DOB)

def button\_modify\_GENDER():

def modify\_GEN(event):

mydb = mysql.connector.connect(user ='root',

password= 'Raman@9045',

host= 'localhost',

database='crms')

mycursor=mydb.cursor()

input\_c\_id=entry\_OFFICER\_ID.get()

s="select \* from officer where officer\_id='{}'".format(input\_c\_id)

mycursor.execute(s)

fetch\_officer=mycursor.fetchone()

if entry\_GENDER.get() == '':

entry\_GENDER.insert(fetch\_officer[6])

else :

s="update officer set officer\_gender = '{}' where officer\_id = '{}'".format(entry\_GENDER.get(),entry\_OFFICER\_ID.get())

mycursor.execute(s)

mydb.commit()

entry\_GENDER = Entry(

window\_MODIFY\_OFFICER\_DETAILS,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Georgia 15')

)

entry\_GENDER.place(

x=198.0,

y=620.0,

width=315.0,

height=30.0

)

entry\_GENDER.bind('<Return>',modify\_GEN)

def button\_modify\_ZONE():

def modify\_ZONE(event):

mydb = mysql.connector.connect(user ='root',

password= 'Raman@9045',

host= 'localhost',

database='crms')

mycursor=mydb.cursor()

input\_c\_id=entry\_OFFICER\_ID.get()

s="select \* from officer where officer\_id='{}'".format(input\_c\_id)

mycursor.execute(s)

fetch\_officer=mycursor.fetchone()

if entry\_ZONE.get() == '':

entry\_ZONE.insert(fetch\_officer[7])

else :

s="update officer set Criminal\_name = '{}' where officer\_id = '{}'".format(entry\_ZONE.get(),entry\_OFFICER\_ID.get())

mycursor.execute(s)

mydb.commit()

entry\_ZONE = Entry(

window\_MODIFY\_OFFICER\_DETAILS,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Georgia 15')

)

entry\_ZONE.place(

x=635.0,

y=620.0,

width=315.0,

height=30.0

)

entry\_ZONE.bind('<Return>',modify\_ZONE)

''' clearing the space'''

def button\_deleting\_OFFICER\_NAME():

var\_ON = StringVar()

label\_OFFICER\_NAME = Label(window\_MODIFY\_OFFICER\_DETAILS,textvariable = var\_ON,bg= "white", font=("Noto Sans", 12 \* -1),padx=380,pady = 5)

label\_OFFICER\_NAME.place(x=198,y=225)

button\_modify\_OFFICER\_NAME()

def button\_deleting\_MOBILE\_NUMBER():

var\_OMN = StringVar()

label\_OFFICER\_MOBILE\_NUMBER = Label(window\_MODIFY\_OFFICER\_DETAILS,textvariable = var\_OMN,bg= "white",font=("Noto Sans", 12 \* -1),padx=380,pady = 5)

label\_OFFICER\_MOBILE\_NUMBER.place(x=198,y=300)

button\_modify\_MOBILE\_NUMBER()

def button\_deleting\_ADDRESS():

var\_OADD = StringVar()

label\_OFFICER\_ADDRESS = Label(window\_MODIFY\_OFFICER\_DETAILS,textvariable = var\_OADD,bg= "white",font=("Noto Sans", 12 \* -1),padx=380,pady = 5)

label\_OFFICER\_ADDRESS.place(x=198,y=380)

button\_modify\_ADDRESS()

def button\_deleting\_POSITION():

var\_POS = StringVar()

label\_OFFICER\_POSITION = Label(window\_MODIFY\_OFFICER\_DETAILS,textvariable = var\_POS,wraplength=770,bg= "white",font=("Noto Sans", 7 ),padx=380,pady = 5)

label\_OFFICER\_POSITION.place(x=198,y=460)

button\_modify\_POSITION()

def button\_deleting\_DOB():

var\_ODOB = StringVar()

label\_OFFICER\_DOB = Label(window\_MODIFY\_OFFICER\_DETAILS,textvariable = var\_ODOB,bg= "white",font=("Noto Sans", 7 ),padx=380,pady = 5)

label\_OFFICER\_DOB.place(x=198,y=540)

button\_modify\_DOB()

def button\_deleting\_GENDER():

var\_OGEN = StringVar()

label\_OFFICER\_GENDER = Label(window\_MODIFY\_OFFICER\_DETAILS,textvariable = var\_OGEN,bg= "white",font=("Noto Sans", 7 ),padx=160,pady = 5)

label\_OFFICER\_GENDER.place(x=198,y=620)

button\_modify\_GENDER()

def button\_deleting\_ZONE():

var\_ZONE = StringVar()

label\_OFFICER\_ZONE = Label(window\_MODIFY\_OFFICER\_DETAILS,bg ='white',textvariable = var\_ZONE,font=("Noto Sans", 7 ),padx=150,pady = 5)

label\_OFFICER\_ZONE.place(x=635,y=620)

button\_modify\_ZONE()

window\_MODIFY\_OFFICER\_DETAILS = Toplevel(window)

window\_MODIFY\_OFFICER\_DETAILS.geometry("1124x700")

window\_MODIFY\_OFFICER\_DETAILS.title("MODIFYING EXISTING OFFICER DETAILS")

window\_MODIFY\_OFFICER\_DETAILS.configure(bg = "#FFFFFF")

canvas = Canvas(

window\_MODIFY\_OFFICER\_DETAILS,

bg = "#FFFFFF",

height = 770,

width = 1124,

bd = 0,

highlightthickness = 0,

relief = "ridge"

)

canvas.place(x = 0, y = 0)

image\_image\_1 = PhotoImage(

file=relative\_to\_assets("image\_1.png"))

image\_1 = canvas.create\_image(

562.0,

44.0,

image=image\_image\_1

)

canvas.create\_text(

210.0,

18.0,

anchor="nw",

text="Modifying existing officer details",

fill="#FFFFFF",

font=("RobotoRoman Bold", 48 \* -1)

)

button\_image\_1 = PhotoImage(

file=relative\_to\_assets("button\_1.png"))

button\_1 = Button(

window\_MODIFY\_OFFICER\_DETAILS,

image=button\_image\_1,

borderwidth=0,

highlightthickness=0,

command=back,

relief="flat"

)

button\_1.place(

x=25.0,

y=26.0,

width=45.0,

height=37.0

)

canvas.create\_text(

17.0,

157.0,

anchor="nw",

text="Enter Officer ID :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

entry\_image\_1 = PhotoImage(

file=relative\_to\_assets("entry\_1.png"))

entry\_bg\_1 = canvas.create\_image(

470.0,

165.5,

image=entry\_image\_1

)

entry\_OFFICER\_ID = Entry(

window\_MODIFY\_OFFICER\_DETAILS,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Georgia 15')

)

entry\_OFFICER\_ID.place(

x=170.0,

y=150.0,

width=590.0,

height=30.0

)

canvas.create\_text(

42.0,

228.0,

anchor="nw",

text="Officer name :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

canvas.create\_text(

42.0,

300.0,

anchor="nw",

text="Mobile number :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

canvas.create\_text(

42.0,

380.0,

anchor="nw",

text="Address :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

canvas.create\_text(

42.0,

460.0,

anchor="nw",

text="Position :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

canvas.create\_text(

42.0,

540.0,

anchor="nw",

text="DOB :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

canvas.create\_text(

42.5,

620.0,

anchor="nw",

text="Gender :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

canvas.create\_text(

550.0,

620.0,

anchor="nw",

text="Zone :",

fill="#717171",

font=("Noto Sans", 18 \* -1)

)

button\_image\_2 = PhotoImage(

file=relative\_to\_assets("button\_2.png"))

button\_2 = Button(

window\_MODIFY\_OFFICER\_DETAILS,

image=button\_image\_2,

borderwidth=0,

highlightthickness=0,

command=search\_officer,

relief="flat"

)

button\_2.place(

x=867.0,

y=148.0,

width=171.0,

height=45.0

)

button\_image\_3 = PhotoImage(

file=relative\_to\_assets("button\_3.png"))

button\_3 = Button(

window\_MODIFY\_OFFICER\_DETAILS,

image=button\_image\_3,

borderwidth=0,

highlightthickness=0,

command=button\_modify\_OFFICER\_NAME,

relief="flat"

)

button\_3.place(

x=198.0,

y=265.0,

width=22.048828125,

height=25.0

)

button\_image\_4 = PhotoImage(

file=relative\_to\_assets("button\_4.png"))

button\_4 = Button(

window\_MODIFY\_OFFICER\_DETAILS,

image=button\_image\_4,

borderwidth=0,

highlightthickness=0,

command=button\_modify\_MOBILE\_NUMBER,

relief="flat"

)

button\_4.place(

x=198.0,

y=345.0,

width=22.048828125,

height=25.0

)

button\_image\_5 = PhotoImage(

file=relative\_to\_assets("button\_5.png"))

button\_5 = Button(

window\_MODIFY\_OFFICER\_DETAILS,

image=button\_image\_5,

borderwidth=0,

highlightthickness=0,

command=button\_modify\_ADDRESS,

relief="flat"

)

button\_5.place(

x=198.0,

y=427.0,

width=22.048828125,

height=25.0

)

button\_image\_6 = PhotoImage(

file=relative\_to\_assets("button\_6.png"))

button\_6 = Button(

window\_MODIFY\_OFFICER\_DETAILS,

image=button\_image\_6,

borderwidth=0,

highlightthickness=0,

command=button\_modify\_POSITION,

relief="flat"

)

button\_6.place(

x=198.0,

y=500.0,

width=22.048828125,

height=25.0

)

button\_image\_7 = PhotoImage(

file=relative\_to\_assets("button\_7.png"))

button\_7 = Button(

window\_MODIFY\_OFFICER\_DETAILS,

image=button\_image\_7,

borderwidth=0,

highlightthickness=0,

command=button\_modify\_DOB,

relief="flat"

)

button\_7.place(

x=198.0,

y=575.0,

width=22.048828125,

height=25.0

)

button\_image\_8 = PhotoImage(

file=relative\_to\_assets("button\_8.png"))

button\_8 = Button(

window\_MODIFY\_OFFICER\_DETAILS,

image=button\_image\_8,

borderwidth=0,

highlightthickness=0,

command=button\_modify\_GENDER,

relief="flat"

)

button\_8.place(

x=198.0,

y=655.0,

width=22.048828125,

height=25.0

)

button\_image\_9 = PhotoImage(

file=relative\_to\_assets("button\_9.png"))

button\_9 = Button(

window\_MODIFY\_OFFICER\_DETAILS,

image=button\_image\_9,

borderwidth=0,

highlightthickness=0,

command=button\_modify\_ZONE,

relief="flat"

)

button\_9.place(

x=635.0,

y=655.0,

width=22.048828125,

height=25.0

)

button\_image\_17 = PhotoImage(

file=relative\_to\_assets("button\_17.png"))

button\_17 = Button(

window\_MODIFY\_OFFICER\_DETAILS,

image=button\_image\_17,

borderwidth=0,

highlightthickness=0,

command=button\_deleting\_OFFICER\_NAME,

relief="flat"

)

button\_17.place(

x=225.0,

y=265.0,

width=16.0,

height=24.220855712890625

)

button\_image\_18 = PhotoImage(

file=relative\_to\_assets("button\_18.png"))

button\_18 = Button(

window\_MODIFY\_OFFICER\_DETAILS,

image=button\_image\_18,

borderwidth=0,

highlightthickness=0,

command=button\_deleting\_MOBILE\_NUMBER,

relief="flat"

)

button\_18.place(

x=225.0,

y=345.0,

width=16.0,

height=24.220855712890625

)

button\_image\_19 = PhotoImage(

file=relative\_to\_assets("button\_19.png"))

button\_19 = Button(

window\_MODIFY\_OFFICER\_DETAILS,

image=button\_image\_19,

borderwidth=0,

highlightthickness=0,

command=button\_deleting\_ADDRESS,

relief="flat"

)

button\_19.place(

x=225.0,

y=426.0,

width=16.0,

height=24.220855712890625

)

button\_image\_20 = PhotoImage(

file=relative\_to\_assets("button\_20.png"))

button\_20 = Button(

window\_MODIFY\_OFFICER\_DETAILS,

image=button\_image\_20,

borderwidth=0,

highlightthickness=0,

command=button\_deleting\_POSITION,

relief="flat"

)

button\_20.place(

x=225.0,

y=500.0,

width=16.0,

height=24.220855712890625

)

button\_image\_21 = PhotoImage(

file=relative\_to\_assets("button\_21.png"))

button\_21 = Button(

window\_MODIFY\_OFFICER\_DETAILS,

image=button\_image\_21,

borderwidth=0,

highlightthickness=0,

command=button\_deleting\_DOB,

relief="flat"

)

button\_21.place(

x=225.0,

y=575.0,

width=16.0,

height=24.220855712890625

)

button\_image\_22 = PhotoImage(

file=relative\_to\_assets("button\_22.png"))

button\_22 = Button(

window\_MODIFY\_OFFICER\_DETAILS,

image=button\_image\_22,

borderwidth=0,

highlightthickness=0,

command=button\_deleting\_GENDER,

relief="flat"

)

button\_22.place(

x=225.0,

y=655.0,

width=16.0,

height=24.220855712890625

)

button\_image\_23 = PhotoImage(

file=relative\_to\_assets("button\_23.png"))

button\_23 = Button(

window\_MODIFY\_OFFICER\_DETAILS,

image=button\_image\_23,

borderwidth=0,

highlightthickness=0,

command=button\_deleting\_ZONE,

relief="flat"

)

button\_23.place(

x=660.0,

y=655.0,

width=16.0,

height=24.220855712890625

)

entry\_image\_1 = PhotoImage(

file=relative\_to\_assets("entry\_1.png"))

entry\_bg\_1 = canvas.create\_image(

470.0,

165.5,

image=entry\_image\_1

)

entry\_OFFICER\_ID= Entry(

window\_MODIFY\_OFFICER\_DETAILS,

bd=0,

bg="#FFFFFF",

highlightthickness=0,

font=('Georgia 15')

)

entry\_OFFICER\_ID.place(

x=170.0,

y=150.0,

width=590.0,

height=30.0

)

var\_OFN = StringVar()

out\_image\_1 = PhotoImage(

file=relative\_to\_assets("output\_1.png"))

out\_bg\_1 = canvas.create\_image(

580.0,

241.0,

image=out\_image\_1

)

label\_OFFICER\_NAME = Label(window\_MODIFY\_OFFICER\_DETAILS,textvariable = var\_OFN,bg= "white", font=("Noto Sans", 12 \* -1),pady = 5)

label\_OFFICER\_NAME.place(x=198,y=225)

var\_OMN = StringVar()

out\_image\_2 = PhotoImage(

file=relative\_to\_assets("output\_2.png"))

out\_bg\_2 = canvas.create\_image(

580.0,

316.0,

image=out\_image\_2

)

label\_OFFICER\_MOBILE\_NUMBER = Label(window\_MODIFY\_OFFICER\_DETAILS,textvariable = var\_OMN,bg= "white",font=("Noto Sans", 12 \* -1),pady = 5)

label\_OFFICER\_MOBILE\_NUMBER.place(x=198,y=300)

var\_OADD = StringVar()

out\_image\_3 = PhotoImage(

file=relative\_to\_assets("output\_3.png"))

out\_bg\_3 = canvas.create\_image(

580.0,

396.0,

image=out\_image\_3

)

label\_OFFICER\_ADDRESS = Label(window\_MODIFY\_OFFICER\_DETAILS,textvariable = var\_OADD,bg= "white",font=("Noto Sans", 12 \* -1),pady = 5)

label\_OFFICER\_ADDRESS.place(x=198,y=380)

var\_POS =StringVar()

out\_image\_4 = PhotoImage(

file=relative\_to\_assets("output\_4.png"))

out\_bg\_4 = canvas.create\_image(

580.0,

476.0,

image=out\_image\_4

)

label\_OFFICER\_POSITION = Label(window\_MODIFY\_OFFICER\_DETAILS,textvariable = var\_POS,wraplength=770,bg= "white",font=("Noto Sans", 7 ),pady = 5)

label\_OFFICER\_POSITION.place(x=198,y=460)

var\_ODOB =StringVar()

out\_image\_5 = PhotoImage(

file=relative\_to\_assets("output\_5.png"))

out\_bg\_5 = canvas.create\_image(

580.0,

556.0,

image=out\_image\_5

)

label\_OFFICER\_DOB = Label(window\_MODIFY\_OFFICER\_DETAILS,textvariable = var\_ODOB,bg= "white",font=("Noto Sans", 7 ),pady = 5)

label\_OFFICER\_DOB.place(x=198,y=540)

var\_OGEN =StringVar()

out\_image\_6 = PhotoImage(

file=relative\_to\_assets("output\_6.png"))

out\_bg\_6 = canvas.create\_image(

365.0,

636.0,

image=out\_image\_6

)

label\_OFFICER\_GENDER = Label(window\_MODIFY\_OFFICER\_DETAILS,textvariable = var\_OGEN,bg= "white",font=("Noto Sans", 7 ),pady = 5)

label\_OFFICER\_GENDER.place(x=198,y=620)

var\_ZONE =StringVar()

out\_image\_7 = PhotoImage(

file=relative\_to\_assets("output\_7.png"))

out\_bg\_7 = canvas.create\_image(

800.0,

636.0,

image=out\_image\_7

)

label\_OFFICER\_ZONE = Label(window\_MODIFY\_OFFICER\_DETAILS,bg ='white',textvariable = var\_ZONE,font=("Noto Sans", 7 ),pady = 5)

label\_OFFICER\_ZONE.place(x=635,y=620)

window\_MODIFY\_OFFICER\_DETAILS.resizable(False, False)

window\_MODIFY\_OFFICER\_DETAILS.mainloop()

def main():

MODIFY\_EXISTING\_OFFICER\_DETAILS()

if \_\_name\_\_ == "\_\_main\_\_":

main()

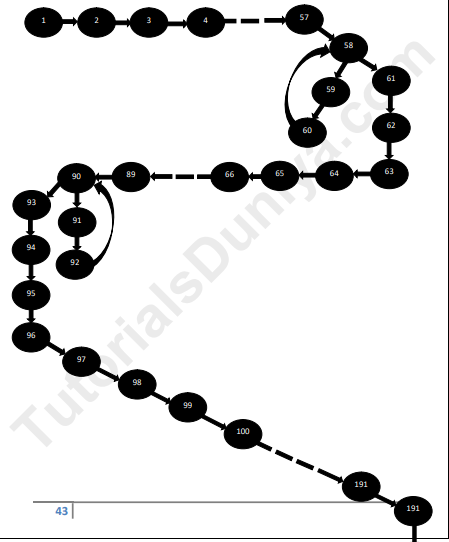
# 

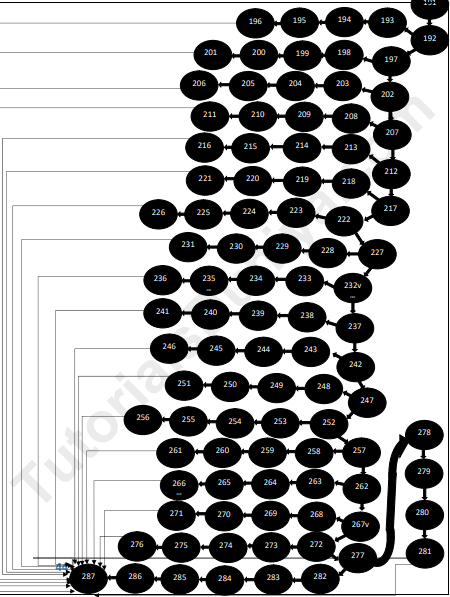
# 

# 

# **7.** **Testing**

## **7.1** **Flowgraph**





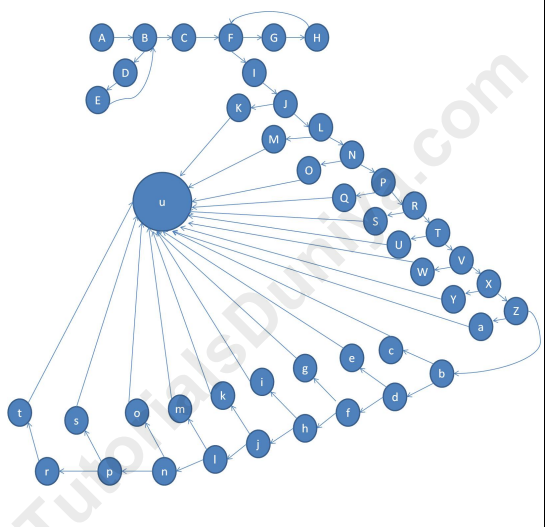
## 

## 

## 

## 

## **7.2** **DD Flow Graph**



## 

## **7.3** **Cyclomatic Complexity**

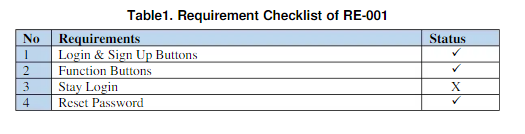
V(G) = E-N+2P

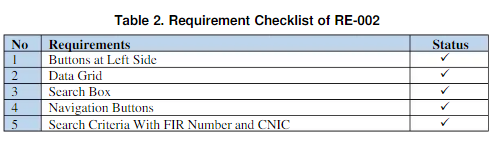
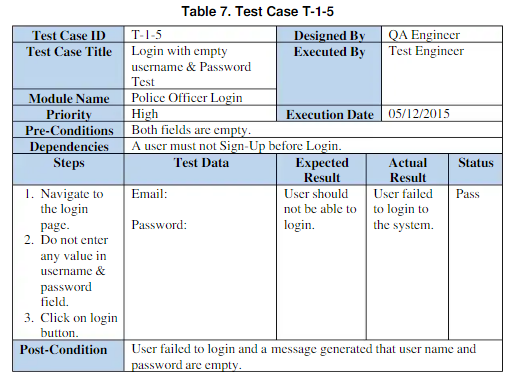
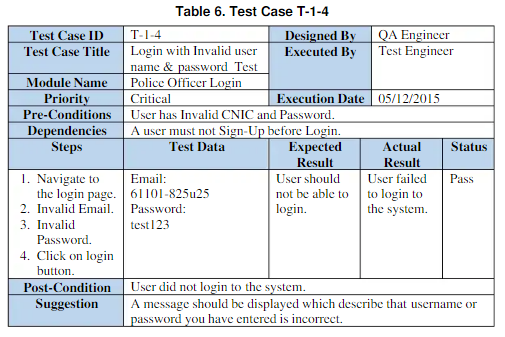
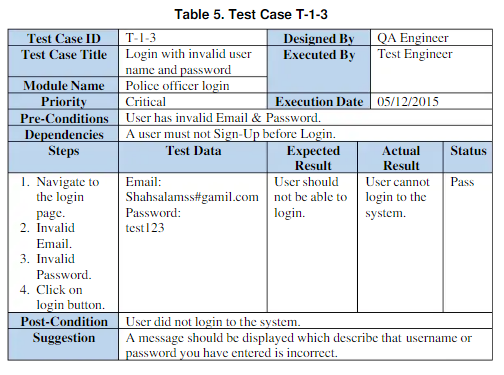
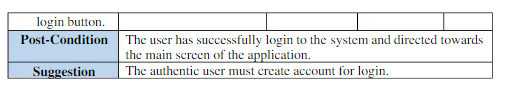
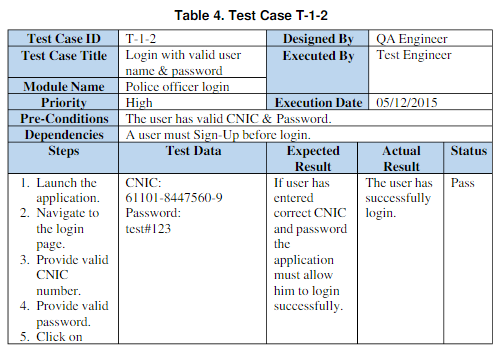
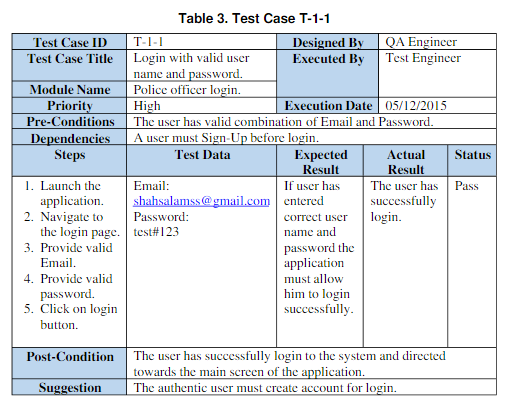
= 64-47 +2

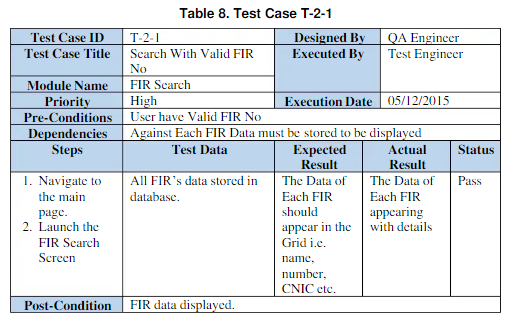
= 19

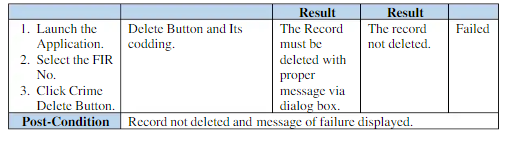
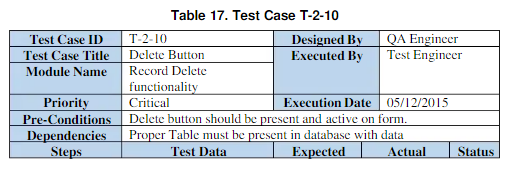
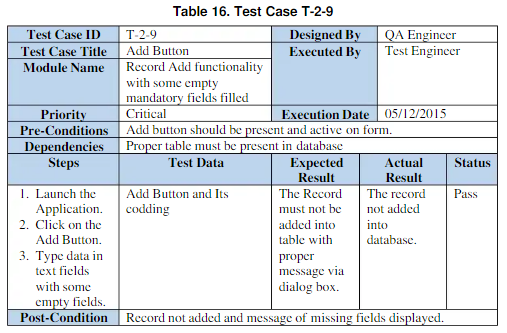
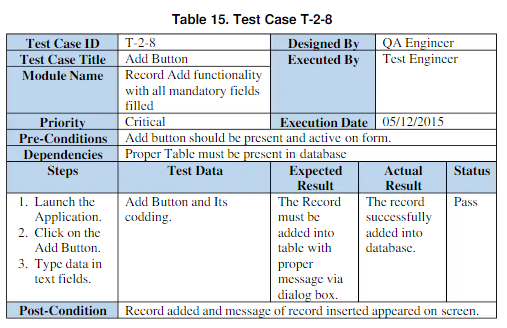
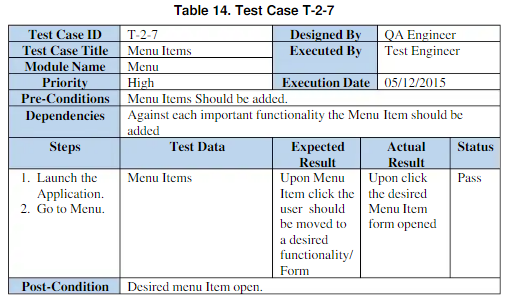
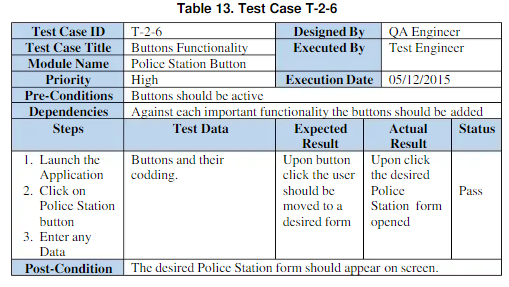
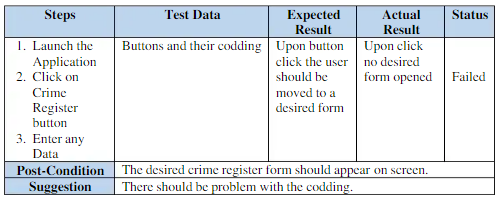
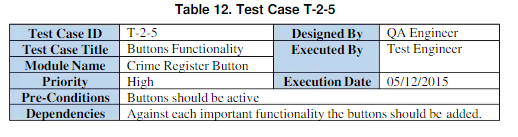
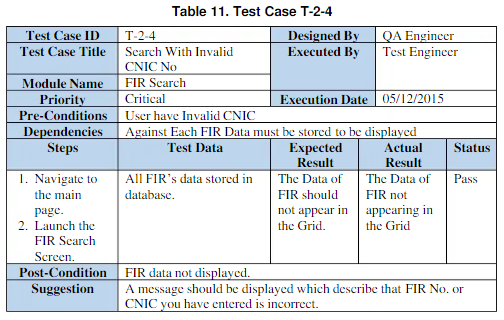
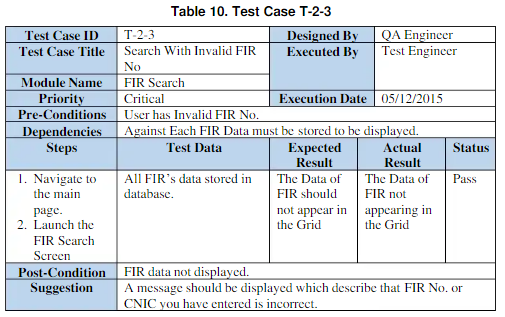
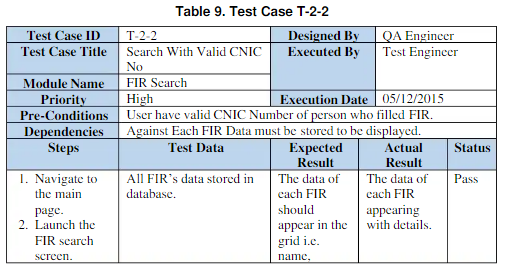
## 

## **7.4** **Test Cases**









# **8.** **Future Scope**

1. This Criminal Record Management System can be further advanced so that it can accommodate different places in the database.
2. We can add the feature of identifying the criminal from their photo. When an officer uploads any photo of an unknown person to check if that person is in the existing criminal database.
3. We can add other users such as judges and other crime bureau so that more users can benefit from this system.
4. We can bring this project on cloud to eliminate the repeatative creation of database on each system.

# **9.** **References**

1. Pressman, R. S., & Maxim, B. R. (2015). Software Engineering: A Practitioner’s Approach. 8th edition. McGraw-Hill
2. Aggarwal, K. K., & Singh, Y. (2007). Software Engineering. 3rd edition. New Age International Publishers
3. <https://www.geeksforgeeks.org/software-engineering-prototyping-model/>
4. <https://youtu.be/mFjE2-rbpm8>