

CORRECTIONAL FACILITY MANAGEMENT SYSTEM



**Research Project Seminar
CS-670-D**

Team #5

**Created For:
Mostafa Omar**

**Sacred Heart University
5151 Park Avenue
Fairfield, CT 06825**

**Fall 2024
Created 09/04/2024**

Name	Email	Telephone Number
Naresh Rayapati	rayapatin@mail.sacredheart.edu	475-332-5803
Vikram Singh Navsinde	navsindev@mail.sacredheart.edu	203-220-1519
Jagadeshwar Alladi	Alladij2@mail.sacredheart.edu	203-410-3051

**Application Used:
Microsoft Office 2016**

Table of Content :

1.0 Introduction to Correctional Facility Management System

1.1 Cover or Title page

1.1.1. Systems Analysis & Design, CS-620-Section#D, Team#5

1.1.2. Sacred Heart University, 5151 Park Avenue, Fairfield, CT.

1.1.3. Trimester Year and Date Created

1.1.4. Team Members

1.1.5. Applications used

1.2 Table of Contents

1.3 Abstract..... 9

2.0 Preliminary Investigation Phase.....10

2.1 Summary of problems, opportunities and/or directives10

2.2 Statement of preliminary scope..... 10

2.2.1 Type of data 10

2.2.2 Business processes 10

2.2.3 System interfaces 11

2.3 Correctional Facility Management System Application cost vs. Value12

2.4 Preliminary project plan13

2.4.1. Master Schedule13

2.4.2 Resource assignments14

3.0 Problem Analysis Phase.....15

3.1 Study the problem domain15

3.1.1 Data collected by the system.....15

3.1.1.1 Methods15

3.1.1.2 Storage15

3.1.1.3 Personnel involved16

3.1.1.4 Time involved	16
3.1.1.5 Sample data model	16
3.1.2 For each report reproduced by the system	17
3.1.2.1 List name and format	17
3.1.2.2 List inputs and outputs	17
3.1.2.3 List responsible personnel	17
3.1.3 Processes currently implemented	18
3.1.3.1 List all processes	18
3.1.3.2 List all hardware and software used	18
3.1.3.3 Functional decomposition diagram of current system	18
3.1.4 System interfaces	18
3.1.4.1 Locations served by the system	18
3.1.4.2 Users served by system	19
3.1.4.3 Other systems it interacts with if any	19
3.1.4.4 Context diagrams of current system	19
3.1.5 Current business structure hierarchy chart	19
3.2 Analyze problems and opportunities	20
3.2.1 Define cause and effect for each problem	20
3.2.2 Update Problem from Preliminary Phase.....	20
3.3 Establish system improvement objectives	20
3.3.1 State new system objectives	20
3.3.2 List new system constraints	21
3.3.2.1 Schedule	21
3.3.2.2 Cost	21
3.3.2.3 Technology	21
3.3.2.4 Policy	21
3.4 Re-evaluate and update project scope	21

4.0 Requirements Analysis Phase.....	22
4.1 Identify requirements	22
4.1.1 List, describe, and defend functional requirements in terms of	22
4.1.1.1 Inputs – data	22
4.1.1.2 Outputs – information	23
4.1.1.3 Processes - manual, automatic	23
4.1.1.4 Storage – database	23
4.1.1.5 Control - authorization, can & cannot do, or see	23
4.1.2 List and defend non-functional requirements	24
4.1.2.1 Performance	24
4.1.2.2 Ease of use	24
4.1.2.3 Cost savings	24
4.1.2.4 Timelines and deadlines	25
4.1.2.5 Training	25
4.1.2.6 Quality Management	25
4.1.2.7 Security and Audits	25
4.2 Master list of all requirements in terms of	26
4.2.1 Priority	26
4.2.2 Deadlines	27
4.2.3 Preliminary Interface model - Context diagram.....	28
4.2.4 Event Table.....	29
4.3 Master list of all requirements in terms of.....	29
4.3.1. Functional Requirements.....	29
4.3.2. Non-Functional Requirements.....	30

5.0 Decision Analysis Phase.....	31
5.1 Candidate solutions	31
5.2 Analyze candidate solutions	31
5.2.1 Feasibility Analysis	31
5.2.2 Cost-Benefit Analysis	32
5.2.2.1 Chart cost analysis of design	35
5.2.2.2 Chart cost analysis of hardware, software, miscellanies purchases	36
5.2.2.3 Chart cost analysis of system operation, and maintenance	36
5.3 Compare candidate solutions	38
5.4 Recommend a final "best" solution	38
5.4.1 Operating environment or constraints	38
5.4.2 Hardware new and existing	38
5.4.3 Software new and existing	39
5.4.4 Staffing new and existing	39
5.4.5 Training new and existing	39
5.4.6 Installation requirements	39
5.4.7 Performance requirements	40
5.4.8 Development requirements	40
5.4.9 Reports to be delivered	40
5.4.10 Security requirements	40
5.4.11 Auditing requirements	40
6.0Design Phase.....	41
6.1 Design the application architecture	41
6.1.1 Networks - intranet and/or internet	41
6.1.2 Database distribution - client/server or network	41
6.1.3 Customization and integration of “off the shelf” software	41
6.1.4 User interface technology – with other users	41
6.1.5 System interface technology – with other systems	42

6.2 Construct detailed models	42
6.2.1 Context model	42
6.2.2 Data flow diagram (DFD) decomposition to system modules	43
6.2.3 Software structure chart showing module hierarchy	68
6.2.4 Use Case Diagram for this Application	69
6.2.5 Use Case Narratives for this Application.....	70
6.2.6 Sequence Diagrams for this Application	82
6.2.7 Activity Diagrams for this Application	84
6.3 Design the system database	85
6.3.1 Construct the detailed entity relationship (ER) diagram	85
6.3.2 Perform the 3 normalization forms, listing dependencies	86
6.3.3 Data dictionary of all the attributes	87
6.4 Design the system interface for each model include the following	89
6.4.1 Module name	89
6.4.2 Parameters passed and their meaning	89
6.4.3 Description of module function	90
7.0 Construction Phase	93
8.0 Implementation Phase	100
8.1 Schedule - use the start and end dates of this Academic year	100
8.2 Staffing - show owners responsibility	100
8.3 Acceptance testing	100
8.3.1 Plan	100
8.3.2 Schedule	103
8.3.3 Acceptance Criteria	103
8.4. Training	104
8.4.1 Plan	104
8.4.2 Schedule	104
8.4.3 Training manuals	104

8.5 Tools to be used in the proposed system	104
8.5.1 Data Base	104
8.5.2 Screen generators	104
8.5.3 Hardware.....	104
8.6 Maintenance documentation.....	104
8.6.1 List of modules	104
8.6.2 Auditing procedures	104
8.6.3 Backup and restore procedures	104
8.6.4 User Manual	104
8.6.5 List of maintenance contacts with phone numbers.....	104
8.6.6 List of vendors, consultants and external contacts	104
8.7 Staffing - hierarchy chart.....	104
8.8 Future enhancements – Clearly documented.....	104
8.9 What did we learned – Openly documented.....	104
9.0 Alphabetical Glossary & Index.....	105
9.1 Glossary.....	105
9.2 Index.....	105
10.0 Bibliography, References & Citations.....	106
10.1. Books.....	106
10.2. Articles.....	106
10.3. Internet or Web sites.....	106
10.4. Other sources.....	106
11.0 Contacts	
11.1 List each stakeholder’s contact information and expertise as related to the system.	

12.0 List of Figures:

FIGURE 1 : FUNCTIONAL DECOMPOSITION DIAGRAM OF CURRENT SYSTEM	20
FIGURE 2 : CONTEXT DIAGRAMS OF CURRENT SYSTEM	21
FIGURE 3 : CURRENT BUSINESS STRUCTURE HIERARCHY CHART.....	22
FIGURE 4 : PRELIMINARY DATA MODEL - ENTITY RELATIONSHIP (ER) DIAGRAM.	28
FIGURE 5 : PRELIMINARY PROCESS MODEL - DATA FLOW DIAGRAM (DFD).	29
FIGURE 6 : PRELIMINARY INTERFACE MODEL - CONTEXT DIAGRAM.	30
FIGURE 7 : DESIGN THE APPLICATION ARCHITECTURE.....	36
FIGURE 8 : CONTEXT MODEL DIAGRAM.	37
FIGURE 9: DATA FLOW DIAGRAM (DFD) DECOMPOSITION TO SYSTEM MODULES, AND TASKS.	38
FIGURE 10 : LEVEL 1.0: DFD	39
FIGURE 11: LEVEL 1.1: DFD	40
FIGURE 12 : LEVEL 2.0: DFD	41
FIGURE 13 : LEVEL 2.1: DFD	42
FIGURE 14 : LEVEL 2.2: DFD	43
FIGURE 15 : LEVEL 2.3: DFD	43
FIGURE 16 : LEVEL 3.0: DFD	44
FIGURE 17 : LEVEL 3.1: DFD	45
FIGURE 18: LEVEL 3.2: DFD	46
FIGURE 19 : LEVEL 3.3: DFD	46
FIGURE 20 : LEVEL 4.0: DFD	47
FIGURE 21 : LEVEL 4.1: DFD	48
FIGURE 22 : LEVEL 4.2: DFD	48
FIGURE 23 : LEVEL 4.3: DFD	49
FIGURE 24 : LEVEL 5.0: DFD	49
FIGURE 25 : LEVEL 5.1: DFD	50
FIGURE 26 : LEVEL 5.2 DFD	50
FIGURE 27 : LEVEL 5.3 DFD	51
FIGURE 28 : LEVEL 6.0 DFD	51
FIGURE 29 : LEVEL 6.1 DFD	52
FIGURE 30 : LEVEL 6.2 DFD	52
FIGURE 31 : LEVEL 6.3 DFD	53
FIGURE 32 : LEVEL 7.0 DFD	53
FIGURE 33 : LEVEL 7.1 DFD	54
FIGURE 34 : LEVEL 7.2 DFD	54
FIGURE 35 : LEVEL 7.3 DFD	55
FIGURE 36 : LEVEL 8.0 DFD	55
FIGURE 37 : LEVEL 8.1 DFD	56
FIGURE 38 : LEVEL 8.2 DFD	56
FIGURE 39 : LEVEL 8.3 DFD	57
FIGURE 40 : LEVEL 9.0 DFD	57
FIGURE 41 : LEVEL 9.1 DFD	58

FIGURE 42 : LEVEL 9.2 DFD	58
FIGURE 43 : LEVEL 9.3 DFD	59
FIGURE 44 : LEVEL 10.0 DFD	59
FIGURE 45 : LEVEL 10.1 DFD	60
FIGURE 46 : LEVEL 10.2 DFD	60
FIGURE 47 : LEVEL 10.3 DFD	61
FIGURE 48 : LEVEL 11.0 DFD	61
FIGURE 49 : LEVEL 11.1 DFD	62
FIGURE 50 : LEVEL 11.1 DFD	62
FIGURE 51: LEVEL 11.2 DFD	63
FIGURE 52 : LEVEL 11.3 DFD	63
FIGURE 53 : LEVEL 12.0 DFD	64
FIGURE 54 : LEVEL 12.1 DFD	65
FIGURE 55 : LEVEL 12.2 DFD	65
FIGURE 56 : LEVEL 12.3 DFD	66
FIGURE 57 : SOFTWARE STRUCTURE CHART SHOWING MODULE HIERARCHY.	66
FIGURE 58 : USE-CASE MODEL DIAGRAM.	67
FIGURE 59 : SYSTEMS SEQUENCE DIAGRAM FOR USE CASE	80
FIGURE 60 : SEQUENCE STAFF MEMBER ROLE	81
FIGURE 61 : ACTIVITY DIAGRAM	82
FIGURE 62 : DETAILED ENTITY RELATIONSHIP (ER) DIAGRAM.	83
FIGURE 63 : PERFORM THE 3 NORMALIZATION FORMS, LISTING DEPENDENCIES.	84
FIGURE 64: STEP 1 INSTALLING MYSQL.....	99
FIGURE 65 : STEP 2 INSTALLING MYSQL.....	100
FIGURE 66: STEP 3 INSTALLING MYSQL.....	101

13.0 List of Tables:

TABLE 1: ASSESS PROJECT WORTH IN TERMS OF COST VS VALUE.....	14
TABLE 2: ASSESS PROJECT WORTH IN TERMS OF COST VS VALUE.....	15
TABLE 3 : PRELIMINARY PROJECT PLAN	16
TABLE 4 : STAFF MEMBER REGISTRATION	18
TABLE 5 : OFFENDER REGISTRATION	19
TABLE 6 : DEFINE CAUSE AND EFFECT FOR EACH PROBLEM.....	22
TABLE 7 : ESTABLISH SYSTEM IMPROVEMENT OBJECTIVES.....	23
TABLE 8: FUNCTIONAL REQUIREMENTS IN TERMS OF INPUTS, OUTPUTS, PROCESSES, STORAGE, AND CONTROL.	25
TABLE 9 : LIST AND DEFEND NON-FUNCTIONAL REQUIREMENTS.....	27
TABLE 10 : EVENT TABLE	31
TABLE 11 : FUNCTIONAL REQUIREMENTS.	32
TABLE 12 : NON-FUNCTIONAL REQUIREMENTS.	32

TABLE 13 :FEASIBILITY ANALYSIS.....	33
TABLE 14: RECOMMEND A FINAL "BEST" SOLUTION.....	35
TABLE 15 : USE CASE NARRATIVE FOR THIS APPLICATION.	68
TABLE 16 : USE CASE 2 NARRATIVE FOR THIS APPLICATION	69
TABLE 17 : USE CASE 3 NARRATIVE FOR THIS APPLICATION	70
TABLE 18 : USE CASE 4 NARRATIVE FOR THIS APPLICATION	71
TABLE 19 : USE CASE 5 NARRATIVE FOR THIS APPLICATION	72
TABLE 20 : USE CASE 6 NARRATIVE FOR THIS APPLICATION	73
TABLE 21: USE CASE 7 NARRATIVE FOR THIS APPLICATION	74
TABLE 22 : USE CASE 8 NARRATIVE FOR THIS APPLICATION	75
TABLE 23 : USE CASE 9 NARRATIVE FOR THIS APPLICATION	76
TABLE 24 : USE CASE 10 NARRATIVE FOR THIS APPLICATION	77
TABLE 25 : USE CASE 11 NARRATIVE FOR THIS APPLICATION	78
TABLE 26 : USE CASE 12 NARRATIVE FOR THIS APPLICATION	79
TABLE 27 : STAFF MEMBER/JAILER REGISTRATION.....	85
TABLE 28 : OFFENDER REGISTRATION	86
TABLE 29 : STAFF MEMBER DASHBOARD.	87
TABLE 30 : CHANGE PASSWORD.	88
TABLE 31 : FORGOT PASSWORD.	89
TABLE 32 : ADD OFFENDER.....	90
TABLE 33 : SCHEDULE - USE THE START AND END DATES OF THIS ACADEMIC YEAR.....	93
TABLE 34 : SCHEDULE.....	94
TABLE 35 : LIST OF MAINTENANCE CONTACTS WITH EMAIL AND PHONE NUMBERS.	97
TABLE 36 : LIST OF VENDORS, CONSULTANTS AND EXTERNAL CONTACT.....	97

1.3 Abstract for Correctional Facility Management System:

The main objective of project is to modernize the legacy systems to web based management applications and still some countries using traditional systems and maintaining ledgers books. These takes long time to finish up a single data flow entry and this led to loss of information of some cases (crime files), insecurity and data redundancy. Similarly, in some highly confidential sensitive cases have been reported where some staff members connive with clients(victims) to hide some files and miss lead the evidence hence leading to compromising the evidence of the matter. This has consequently resulted in time wastage to handle cases, increased corruption and insecurity of important files hence making entire process costly. When prison reports are needed in case study, it takes a long time to reach our hands, and it leads delay in urgent decisions. In between this process a lot of loopholes created in the system because lack of tracking or maintaining of the information available in the departments and there are no security measures in place to safeguard the available data. These necessity makes us automate the system to make it more secure, efficient and effective. These application track and maintain the activities of the offender from admit to release and its controlled by the staff member.

2.0 Preliminary Investigation Phase

2.1 Summary of Problems, Opportunities, and Directives

In this system each staff member is the primary owner to maintain and track the offenders in the jail. Staff members need to stop originate the prisons in the blocks. If young person arrest in the drug trafficking, violent warfare ,gang fights, the staff member need to associate a sperate cell and he need to look non-affiliated with other prisoners mental patients are joined, the officer need to manage the timing and meetings because it disturb the peace of their neighbor prisons and officer look after the mental health issues of prisoners, such as having to arrange for psychiatric services. The officer needs to maintain schedules and court cases, hearings of the prisons and transportation with security.

This application is web-based application, it maintains a single electronic record id for each offender and these id helps us to monitor the offender status by the officer. This application is very scalable and efficient with browser compatibility.

2.2 Statement of Preliminary Scope

2.2.1 Description of Data in use by System Study

The current system includes text, graphical output, and a specified knowledgebase.

The following data will be recorded, transcribed, monitored, evaluated for progress and stored.

- *Global Search:* These Procedure explains how to retrieve any offender record in the system where they are active or inactive.
- *Create offender:* These create offender form is used to assign a unique ID number. These number will be theirs for life.
- *Bookings:* This module includes the following and these modules use to locate the bed in the jail and update the personal, educational/work, physical characters and alerts with address.
- *Incident reporting:* These module records the incident that occur at your facility.
- *Scheduling:* This module explains about the scheduling the offenders to the court.
- *Legal Case management system:* Court cases and hearing notifications with charges are managed here.

2.2.2 Business Processes

The Correctional Facility Management System is the web-based application which runs in browser, and it keeps the responsibility for offender who have been accused or convicted of criminal offense. It maintains single electronic “offender Id” for each prison in the system. Which helps us to track and maintain case legal records from “Admission” until “Release”. These system holds offender begins no later than date on which court convicts the person and before accused person may be held in jail waiting trail. Upon conviction, the court usually order a pre-sentence investigation to assess sentencing options based on the particular of the case. Sentences can range from probation to serving time in prison, with intermediate sanctions to bail/release. This system

can integrate with external computers operated by other agencies in the criminal justice system: law enforcement police or court.

2.2.3 System Interface with Users, Locations, and Other Systems

Interface with Users:

This Application is an intranet system which utilizes a web browser to deploy the application.

- **Create offender:** These create offender form is used to assign a unique ID number. These number will be theirs for life. By using this Offender id, staff member can track and maintain the offender records in the system.
- **Aliases and Other Identifiers:** This form used to record the offenders known aliases name and to designate a working name. The working name is the name entered when the offender Id number was assigned. These is the name appears when we retrieve the offender record.
- **Alerts:** This form is used to record the medical, security or transport alerts on the offender
- **Personal Information:** This form is used to record the personal information like place of birth, state and country.
- **Physical identifiers:** This form is used to record the physical characteristics (height, weight, hair and eye color etc.,)
- **Offender Address:** This form used to record all addresses has residents. Information includes mail address, phone number, global phone number and email address).
- **Education/Work:** This form used to record the educational and employment history
- **Languages:** This form used to record the offender languages
- **Incident detail recording:** This form is used to record the details of incidents occur at and also record the information about damages resulting from incident, or any officer involved in the incident, in situations where there is offender involvement
- **Incident Logs:** This form used to retrieve all incident logs and It generates id.
- **Offense in Custody:** These offense in custody form manages any details associated with the offense in custody process. These includes recordings details of any charges, notices, hearings, investigations and penalties associated with incident .
- **Schedule appointment:** This form can be used to schedule the offender to court hearings and cases
- **Legal Cases:** This form is used to create a court case with dates with offenses and sentences and bail details.

Interface with Locations:

Correctional Facility Management System-- Since this Application is an intranet system using a web-browser, the application can be accessed using either a PC or MAC. In addition, either a tablet or Apple iPad can be used since the webpage will automatically configure to the screen size. The constraint will be that the website can only be accessed from a local server. This will limit cyber security vulnerabilities.

Interface with Other Systems:

Application will interface with web browser of user's choice to deploy the application. In addition, the system will also interface with SQL Server to query data needed for the application. Data will include Offender Data.

2.3 Assess Project worth in terms of Cost Vs Value

In the cost value analysis, the following chart displays the total development cost, the projected revenue for the first six years of system implementation, and six years of operating costs. This analysis assumes a five percent increase in operating costs each year. Based on this analysis, the new system will be profitable and the benefits outweigh the costs. In a further payback analysis, the system will reach profitability just prior to the third year of operation. For an in-depth analysis of the following figures, reference the cost-benefit analysis of 5.2.2.

Cost Value Analysis		
Expected Revenue (Years 1-6)		\$660,000.00
Development Cost		
<i>Total Personnel Cost</i>	\$65,000.00	
<i>Total Hardware Cost</i>	\$16,600.00	
Total Development Cost		(\$81,600.00)
Expected Operating Costs (Years 1 to 6)		(\$251,670.77)
Total Profit		\$326,729.23
<i>*In-depth analysis is presented in 5.2.2</i> <i>*All formulas are derived from 5.2.2</i> <i>*Note: Preliminary estimates do not consider the time value of money</i>		

Table 1: Assess Project worth in terms of Cost Vs Value

Cost Value Analysis		
Expected Revenue (Years 1-6)		=SUM('Net Value'!B10:H10) Present
Development Cost		
<i>Total Personnel Cost</i>	=SUM('Estimated Costs'!E5:E10)	
<i>Total Hardware Cost</i>	=SUM('Estimated Costs'!E14:E20)	
Total Development Cost		=-'Estimated Costs'!E22
Expected Operating Costs (Years 1 to 6)		=SUM('Payback Analysis'!B5:H5)
Total Profit		=SUM(C2:C7)
<i>*In-depth analysis is presented in 5.2.2</i> <i>*All formulas are derived from 5.2.2</i> <i>*Note: Preliminary estimates do not consider the time value of money</i>		

Table 2: Assess Project worth in terms of Cost Vs Value

2.4 Preliminary Project Plan

2.4.1 Master Schedule for Entire Project

Start Date: 09/04/2024

ID	Task Name	Duration	Start	Finish	Resource Names	% Complete
1	Introduction	1 day				
2	Cover or title page	1 day	09/04/2024	09/04/2024	Naresh	1.429
3	Preliminary Investigation Phase	4 days				
4	Summary of problems, opportunities, and directives	1 day	09/04/2024	09/04/2024	Vikram Singh	2.587

5	Statement of Preliminary Scope	1 day	09/11/2024	09/11/2024	Jagadeshwar	4.286
6	Assess Project worth in terms of Cost Vs Value	1 day				5.714
7	Preliminary Project Plan	1 day	09/04/2024	09/04/2024	Vikram Singh	7.143

Table 3 : Preliminary Project Plan

2.4.2 Resource Assignment

Personnel

- 1 System Analyst
- 2 Python Programmers
- 1 MYSQL Database Specialist

Software

- Python (free)
- MYSQL Database (free)
- AWS (Paid)

Hardware

- Laptops
- External Monitors
- Any device with Browser Installed

3.0 Problem Analysis Phase

3.1 Study the problem domain

In the current system, the complete data storage and maintenance is carried out manually. This requires more manpower and is a time-consuming process. In some cases, there may be chances of making errors in data representation or storage which leads to hassles. In the current system, if the data is increased, it is difficult to organize and retrieve the accurate data when required. Another important drawback is, the manual systems will have the less scope for quick data retrieval from huge data stored. Another major problem in the current system is the maximum chances of data compromise as each information is represented and stored in the paper.

3.1.1 Data collected by the system

In this Correctional Facility Management System, we usually collect and maintain the data of the offenders arrested in various crimes or malpractices. We maintain a database with respect to crime branch and the offender age and imprisonment. This system also holds a database to maintain the schedules of offenders for their medical checkups, counselling sessions, court sessions or family meetings. For each offender a unique ID is if helps in retrieving their respective data.

3.1.1.1. Methods

- In this system, staff member is the owner/administrator who adds the data of each offender and updates all available statuses for all offenders.
- Staff member adds all the schedules of each offender.

3.1.1.2. Storage

- The complete data of the system is stored in Sequential format in a database. This format allows in easy store & retrieve of data.
- It is hosted in the server making the system to accessible from anywhere through the valid credentials.
- We use SQL for data storing and managing. Maintaining backups for data is taken care to ensure performance of the whole system.

3.1.1.3. Personnel involved

This system is proposed to remove the manual process in the legacy systems making it more secure in data representation and storage. In this the primary personnel involved are the staff members who manages all the data of the system. And the offenders.

3.1.1.4 Time Involved

This system requires less time to store and retrieve the data as it is developed and maintained with regular updates to the software and technologies used.

3.1.1.5. Sample data model

As mentioned above, in this system there are two different users: Staff Member and Offender. Each of these users will have certain roles, staff members being the key role player with all authorities.

3.1.1.5.1. Staff Member Registration

ID	Field Name	Data Type	Constraint
0	Unique Id	int	Primary Key
1	Name	varchar(20)	Primary Key
2	Designation	varchar(20)	Primary Key
3	Contact/Email	bigint	Primary Key
4	Branch	varchar(20)	Primary Key
5	Password	varchar(8)	Primary Key

Table 4 : Staff Member Registration

3.1.1.5.1. Offender Registration

ID	Field Name	Data Type	Constraint
0	Unique Id	int	Primary Key
1	Name	varchar(20)	Primary Key
2	Gender	varchar(20)	Primary Key
3	Age	int	Primary Key
4	Pardoning Power	varchar(20)	Primary Key
5	Password	varchar(8)	Primary Key
6	Duration	int	Primary Key
7	Contact/Email	bigint	Primary Key
8	Evidences Produced	varchar(20)	Primary Key

Table 5 : Offender Registration

3.1.2 For each report reproduced by the system

3.1.2.1 List name and format

Usually in the reports, the staff members maintain the offender's personal data, pardoning power information and their schedules from attending counseling to visiting court for regular interrogations.

3.1.2.2 List inputs and outputs

Offender's data like Name, contact, Pardoning power/punishment, Term of punishment, Gender are the primary inputs. Adding to which staff member also inputs the schedules of each offender to the system. Once for each input, there comes a unique ID to quick fetch the data as the output of the system.

3.1.2.3 List responsible personnel

In this whole system, there are two key personnel as:

- Staff Member to record and manage data
- Offender, the source of data

3.1.3 Processes currently implemented

3.1.3.1 List all processes

In the existing system, all details are to be maintained by the staff member to store and secure the data without losing or compromising. For this, soon after the offender is produced staff member collects their basic data and punishments or the term of imprisonment etc. and save them manually in a physical database (Books/Papers). They also note each of the offender's schedules for court visits, hospital visits and counselling sessions to avoid the overlapping with their fellow offenders.

Offenders belong to the area of the prison and hence they submit all their details to the staff member of the location which are maintained in the datasheets.

3.1.3.2 List all hardware and software used

As this is a complete manual system where each data is stored in datasheets, there is no specific software to functionality maintenance or hardware to access the data.

3.1.3.3 Functional decomposition diagram of current system

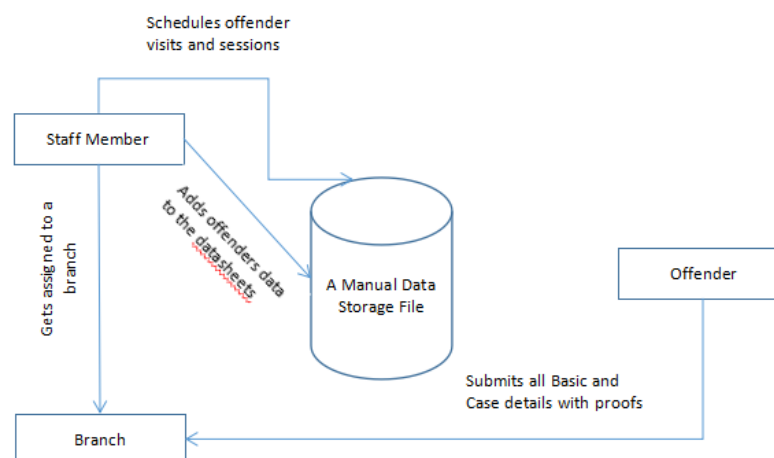


Figure 1 : Functional decomposition diagram of current system

3.1.4 System interfaces

3.1.4.1 Location Served by the System

The current system is the manual system that have no remote data storage or databases. For this reason, it has no access points than to check and retrieve manually. Every bit of data is manually stored in this system and that is to be accessible manually by checking each instance of data even to retrieve old data.

3.1.4.2. Users Served

This system is used majorly in the judicial centers where they need to manage the data of all the offenders. Hence the possible users are likely to be the staff in charge of the prison and the offender or the prisoner.

3.1.4.3. Other systems it interacts with if any

As this is a complete offline procedure, there are no other systems interlinked to this.

3.1.4.4 Context diagrams of current system



Figure 2 : Context diagrams of current system

Fig: Context diagrams of current system

3.1.5 Current business structure hierarchy chart

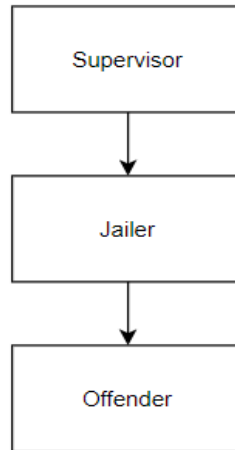


Figure 3 : Current business structure hierarchy chart

Fig: Current business structure hierarchy chart

3.2 Analyze problems and opportunities in current system

3.2.1. Define cause and effect for each problem

Problem or Opportunity	Cause and Effect
Current system is paper based	<ul style="list-style-type: none">• There is no standard procedure to secure the data for easy retrieval
Method of data input all manual	<ul style="list-style-type: none">• Allows for potential mistakes in data entry.• Data entry and retrieval takes up more time
Lack of online presence	<ul style="list-style-type: none">• Online presence is important to stay competitive against other serv.

Table 6 : Define cause and effect for each problem

3.2.2 Update Problem from Preliminary Phase

In addition to the problems listed in the preliminary phase, the current system problems involve a lack of Security, the absence of a data structure. Staff members maintain all the offender data and case related proofs manually in the office locations which may have the chances of proof loss or compromises. The current system is proposed to address all these issues maintaining the data secure and accessible to authorized users only.

3.3 Establish system improvement objectives

3.3.1. State new system objectives.

After crucial examining from the cause & Effect analysis, the drawbacks of current system were shortlisted to address. We have proposed this system to fully automate the process with minimal man power and in less time.

Problem or Opportunity	System Objective
Current system is paper based	<ul style="list-style-type: none">● Create a software and a user interface for end users to be able to access the offender data.
Method of data input is not user friendly	<ul style="list-style-type: none">● Make it easier for administrators to update or record new offender data.● Allow for updates and changes to be made but revisions will also be timestamped.
Lack of online presence	<ul style="list-style-type: none">● Make a user-friendly interface for easier navigation on the Webpage.● Make the Webpage a place of reference for any important information or announcements.● Creating a dashboard that depicts the previous activity or schedules or the case proofs submitted.

Table 7 : Establish system improvement objectives

3.3.2. List new system constraints

All the major constraints of new system like Schedule, Cost, Technology profile and scope are clearly specified in the **Preliminary Project Plan (2.4)** section.

3.4 Re-evaluate and if needed, update the project scope (2.5)

Scope of Project is addressed and listed in the section 2.5

4.0 Requirements Analysis Phase

4.1 Identify requirements (for objectives stated in 3.3.1)

4.1.1 Functional requirements in terms of inputs, outputs, processes, storage, and control.

Requirement	Input	Output	Processes	Storage	Control
Easy-to-use program application	System will provide login credentials for portal administrators/ staff members.	System will display all the registered members the data in textual format, and in numerical format	System will create new user profile/dashb oard.	System will store offender data. Their medical schedules, court schedules and the counselling schedules. The day-to-day sales and purchase data.	System will grant access to staff members through validated login credentials.
Fillable online forms	System will allow new profiles to be created for offenders and staff members, and modified`	-	System will create new referring offender profile	System will store offender profile to offender database, Staff member data to its database.	This can access registered Staff members and offenders' data.
User-friendly web interface for users	System will offer a simplified UI for easy navigation of various parts of	System will display simple, easy to understand navigation tools. Drop	System will display drop down menus when user clicks on them	-	Errors stored in database

	the dashboards and result pages	down menus and categories used to allow users to get to where they need quickly			
Software communication	System will provide easy-to-use communication interfaces for fast and easy communication between database and staff members for data retrieval	-	-	System will Store messages sent and received	Staff members can login to view or add or modify offender data
Application runs on different platforms	System authenticates user's credentials	System will display mobile friendly UI.	System will run on multiple devices	System will have access to all functions no matter the device	Staff Members will have access to the system on all devices
Privacy of Information	No staff besides administrators should have access to offender data	Any confidential information will be accessed only by a lock password.	Saves cost of releasing any confidential information that could result in a lawsuit	Three months to test and code.	No erroneous leakage of offender data

Table 8: Functional requirements in terms of inputs, outputs, processes, storage, and control.

4.1.2 List and defend non-functional requirements

Non-functional Requirements

Requirement	Performance	Ease of use	Cost savings	Timelines and deadlines	Training	Quality Management	Security and Audits
Forms to Open	Response time < 100 milliseconds	Intuitive forms and buttons direct user.	Fewer costs as mostly administrator manages the system	One month to test and code.	User Data Entry and management	Mean system errors < 5 per business day.	User authentication and verification procedures.
Auto-Complete Query	Response time < 250 milliseconds	Just few clicks to get any data required	-	Two months to test and code.	Help menu	Backup and restore time no more than 30 minutes.	Secure data transmittal and storage.
Populate forms	Response time < 250 milliseconds	One mouse click tab or add/edit button.	-	Two months to test and code.	N/A	Mean system errors < 5 per business day.	Backup and restore and off-site storage.
Report	Production of any end report will take less than 2 seconds in 95% of the cases.	Check box to vary graph views, tab to select report.	-	Three months to test and code.	Help menu	Mean system errors < 5 per business day.	Secure data transmittal and storage.

Manage Referrals	Process each new farmer profile addition per second at peak load.	One mouse click add/edit patient profile.	25%-45% compared with outsourced data storage	Three months to test and code.	Help menu	Backup and restore time no more than 30 minutes.	Backup and restore and off-site storage.
Follow-Up Requests	Process each new follow-up request per second at peak load.	One mouse click add new/select existing.	25%-45% compared with outsourced data storage	Three months to test and code.	Help menu	Backup and restore time no more than 30 minutes.	Secure data transmittal and storage.

Table 9 : List and defend non-functional requirements

4.2 Analyze Functional Requirements for New System

4.2.1 Preliminary Data Model - Entity Relationship (ER) diagram

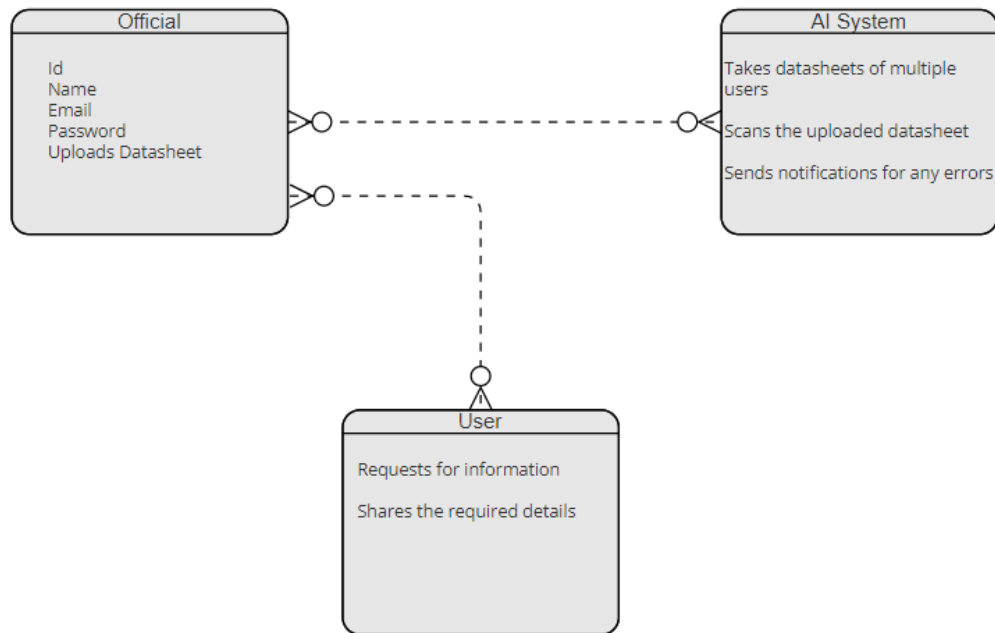


Figure 4 : Preliminary Data Model - Entity Relationship (ER) diagram.

Fig: Preliminary Data Model - Entity Relationship (ER) diagram

4.2.2 Preliminary Process Model - Data Flow diagram (DFD)

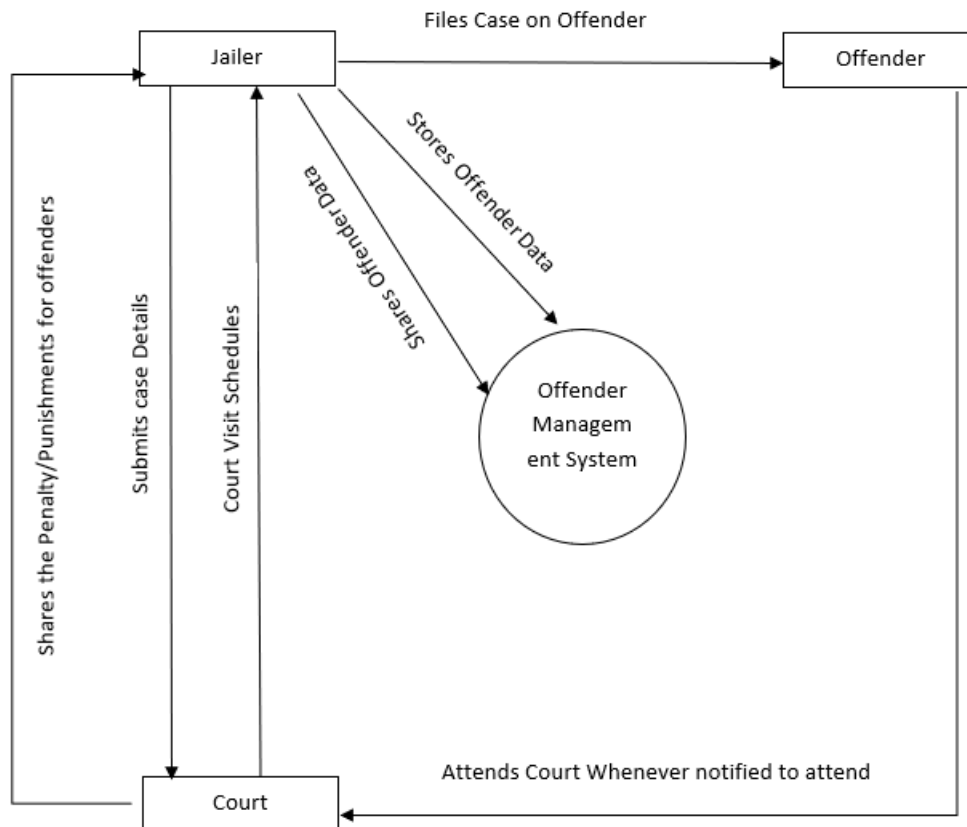


Figure 5 : Preliminary Process Model - Data Flow diagram (DFD).

Fig: Preliminary Process Model - Data Flow diagram (DFD)

4.2.3 Preliminary Interface model - Context diagram

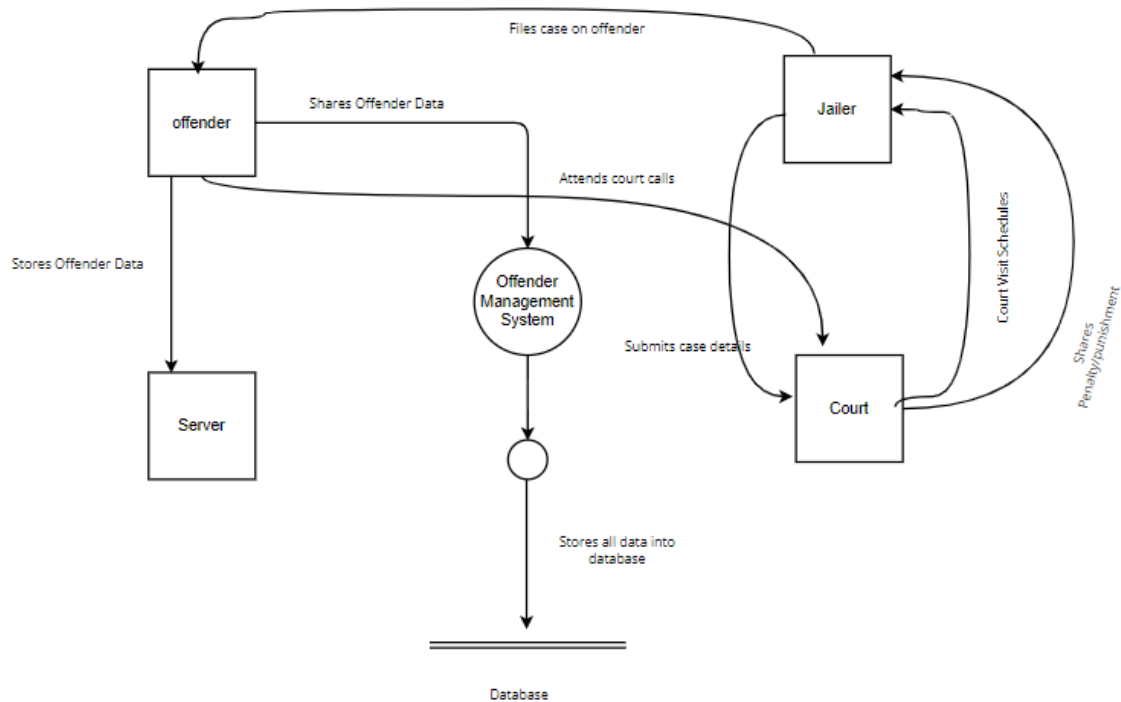


Figure 6 : Preliminary Interface model - Context diagram.

Fig: Preliminary Interface model - Context diagram

4.2.4 Event Table

Event Table						
Event	Event Type	Trigger	Source	Use Case	Response	Destination
Staff Member gets registered into the system	External	Staff member dashboard created	User	Set Staff	Staff Member ID Created.	Offender System Database
Adds the offender's data	State	Background Check	User	Check Eligibility	Confirmation of added offender details	Offender System Database
Adds offender case details	Internal	Notification of added details	User	Validate details	Notification of addition	Offender System Database
Modifies if any updates or changes	Internal	Emergency circumstance	User	Validate details		User

Table 10 : Event Table

4.3 Master list of all requirements in terms of

4.3.1. Functional Requirements

Functional Requirements			
Requirements	Priority	Deadline	Supporting requirements
Staff Member Registration/Sign up	High	NA	Staff Personal and professional data
Login & Logout	Medium	NA	Email and Phone Number of Staff Member

Dashboard	High	NA	Offenders Data
Add/Edit/Delete for Administrator	Medium	NA	Offender schedules information
Modifications auto update Staff Member dashboard	Medium	NA	Only when changes to be made

Table 11 : Functional Requirements.

4.3.2. Non-Functional Requirements

Non-Functional Requirements	
Requirements	Priority
Adjustability & Flexibility	Medium
Accessibility & Maintainability	High
Security & Privacy	High
User Friendly	High
Performance and Stability	High

Table 12 : Non-Functional Requirements.

4.4 Re-evaluation and update project scope

Project scope does not need to be updated.

5.0 Decision Analysis Phase

5.1 Candidate solutions

There are no such systems for Offender data Management. Even if the systems are available they are confined to only specific goals and with restrictions on data storage and maintenance. Those systems are best suited for the small scale Judicial Systems. For the systems like Correctional Facility Management System, it requires high-end systems to run with the necessary

hardware to run on software. Our Correctional Facility Management System aims to offer professional, easy-to-use features in a more cost-friendly manner. Below are the feasibility points listed from the feasibility analysis phase on comparing with the existing commercial solutions.

5.2 Analyze candidate solutions

5.2.1 Feasibility Analysis

Technical Feasibility	The framework configuration is in fact attainable to build up the framework. Specialized Staff, Developers required to structure the framework. The structured framework can be bolstering new improvement effectively and can coordinate with it easily.
Operational Feasibility	The project has been developed in such a way that it becomes very easy even for a person with little computer knowledge to operate it. This software is very user friendly and does not require any technical person to operate .Thus the project is even operationally feasible.
Economic Feasibility	Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis. .net using visual C# and Sequential Query Language database is easily available on the internet.
Schedule Feasibility	The framework will require additional time and exertion for structure and improvement. The time assigned right now is sufficient to plan and build up the framework.

Table 13 :Feasibility Analysis.

5.3 Compare candidate solutions

We have closely observed the existing system and the drawbacks of all the manual processes in it. Our team have found the solutions at every point of the existing system and put all together to device a software which addresses the major drawbacks of the current judicial system. Listed are the final approaches from the team to start with the Correctional Facility Management System that makes the data storage and maintenance easier in time.

- Firstly, this Correctional Facility Management System is a web-based system to manage all the activities.
- This system includes below functionalities:

- The primary user of this system is the staff member/administrator of the particular area who keeps track of all the offender data.
- Ability to manage the offender schedules like court visits, medical checkups, and other activities assigned for each offender based on their duration of course.

Idea: Ultimately the idea of this proposed system is to save time and maintain the data in a more secure way. This system is proposed with the intention of maintaining the offender data, case proofs and evidence in a virtual storage with a limited access authority. Along with this, it eases the way data is represented to make it clearly understandable even after the longer durations of case.

5.4 Recommend a final "best" solution

Criteria	New (Recommended)	Existing
Operating Environment or Constraints	<ul style="list-style-type: none"> • Must need Wi-Fi/data connection to communicate with the server. • Off-line viewings are not available 	<ul style="list-style-type: none"> • It is a manual system that has no operating environment.
Hardware new and existing	<p>Windows</p> <ul style="list-style-type: none"> • 2+ GHz processor. • 8+ GB RAM. • 500 MB of available hard-disk space • Sequential Query Language Database Server <p>Mac</p> <ul style="list-style-type: none"> • 2.1+ GHz Intel™ processor. • 8 GB RAM. • 500 MB of available hard-disk space. 	<ul style="list-style-type: none"> • Existing is a manual system where a person is assigned to data enter and retrieval. • No Virtual data storage and hence no hardware to access the data.

Software new and existing	<ul style="list-style-type: none"> • Own personal development • WEEKLY UPDATES/BUG FIXES • Many customer services options as well as Quick Chatting with professionals at any time. 	<ul style="list-style-type: none"> • Phone number or meeting in person is the only medium to talk to helping professionals • No updates/bug fixes
Staffing new and existing	IT and Human Resources	Staff Member In charge
Training new and existing	Online functions/classes to teach new users on how to use the software	User manual or from direct interactions.
Installation requirements	Microsoft Windows Installer 3.0	Any generation web browser
Performance requirements	All page material is loaded in under 3 seconds	No such factor to influence as it is a manual system.
Development requirements	Client-Server model Sequential Query Language Database	No software can be suggested as it is a manual system.
Reports to be delivered	Each offender's daily/weekly report on court visits, Duration of imprisonment etc.	Reports generation is manually retrieving data from the data sheets.
Security requirements	<ul style="list-style-type: none"> • All personal data secured • Secure passwords 	No software to maintain security factors.
Auditing requirements	Login and access log trails	NA

Table 14: Recommend a final "best" solution.

6.0 Design Phase

6.1 Design the application architecture

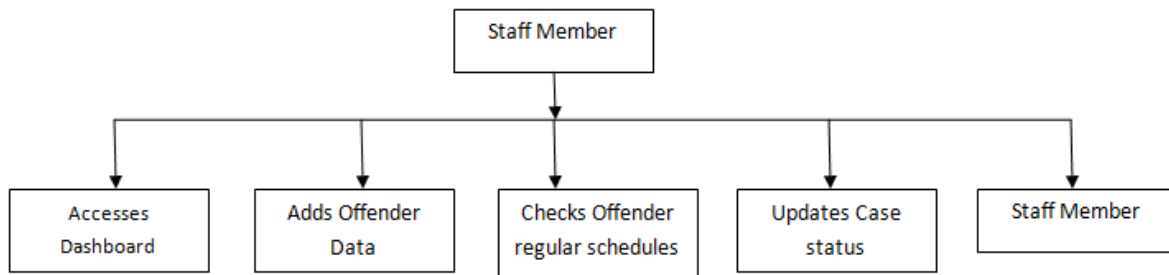


Figure 7 : Design the application architecture.

Fig: Application Architecture

6.1.1 Networks: Internet.

6.1.2 Database Distribution: Network.

6.1.3 Customization and integration of “Off the shelf” software

As for off the shelf software, in our system we used the sequential query language as our database management system. This makes the data storing, retrieval easier and faster.

6.1.4 User Interface Technology--With other Users

There is no outside user access for this management system as it includes all the confidential data secured by the authorized staff member.

This system can be used by multiple staff in charge of a particular location or branch.

6.1.5 System Interface Technology--With other Systems

The system will not be utilizing any other systems for the interface technology.

6.2 Construct detailed models

6.2.1 Context Model Diagram

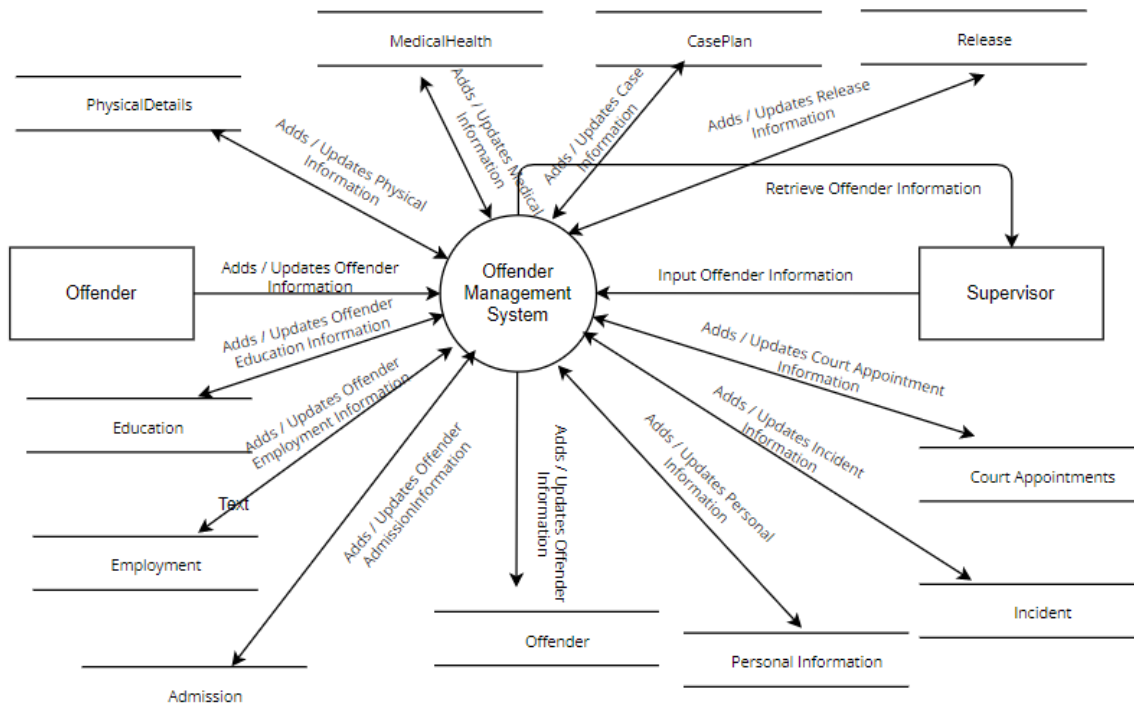


Figure 8 : Context Model Diagram.

Fig: Context Diagram

6.2.2 Data flow diagram (DFD) Decomposition to System Modules, and Tasks

Level 0: DFD

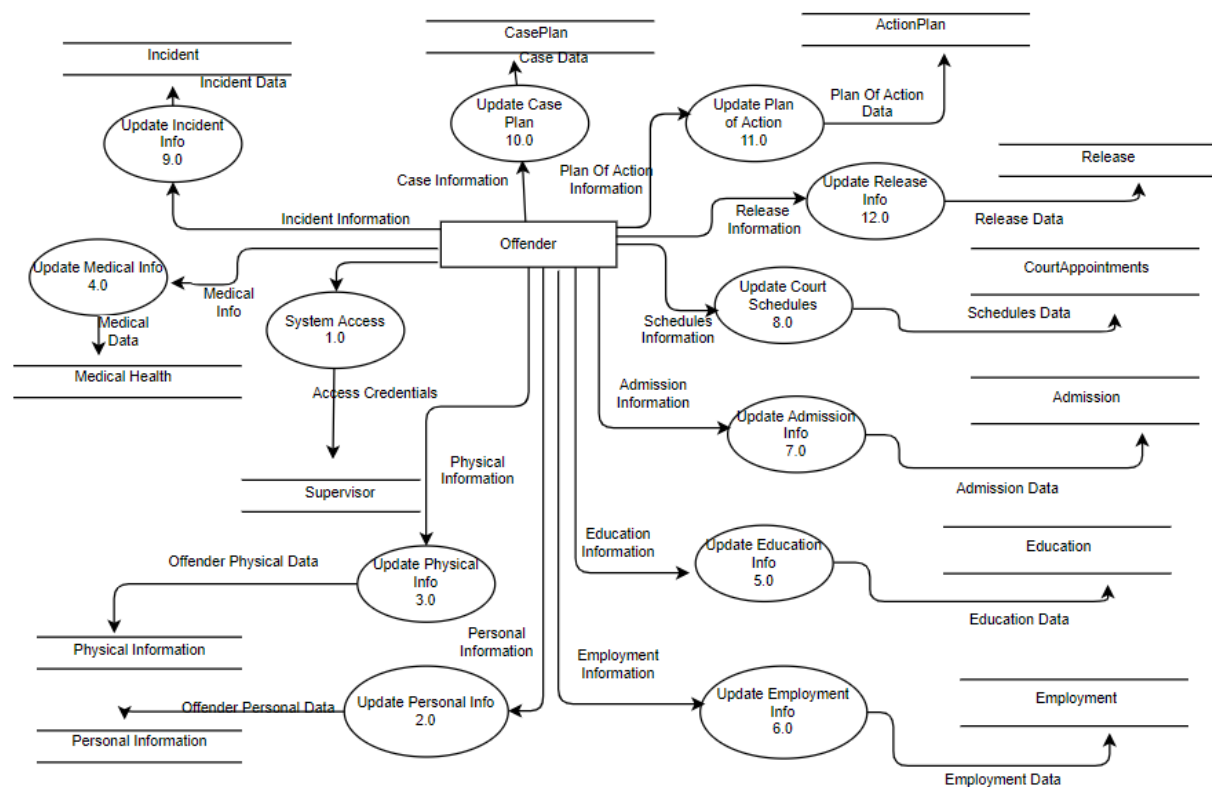


Figure 9: Data flow diagram (DFD) Decomposition to System Modules, and Tasks.

Level 1.0: DFD

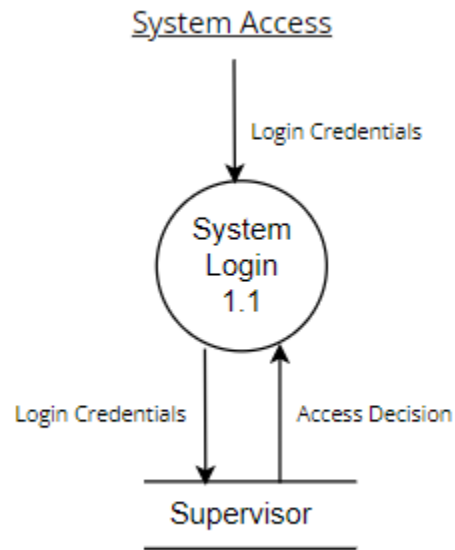


Figure 10 : Level 1.0: DFD

Level 1.1: DFD

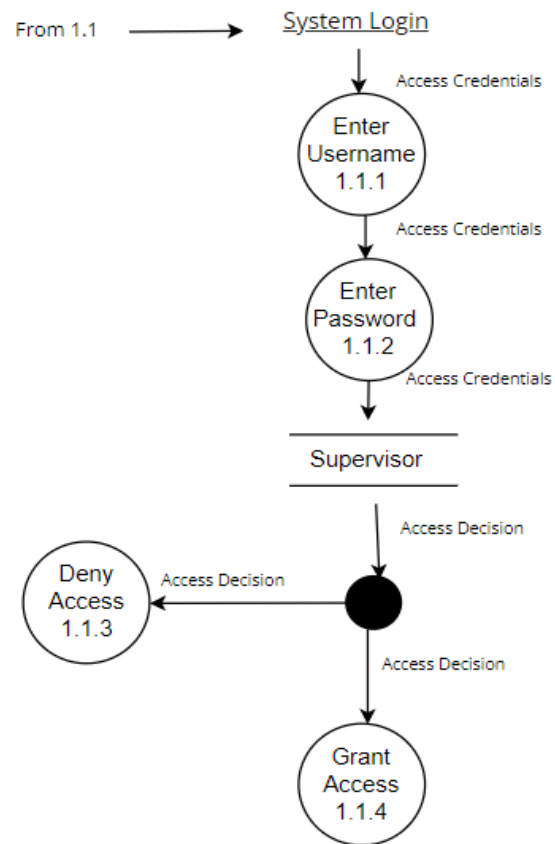


Figure 11: Level 1.1: DFD

Level 2.0: DFD

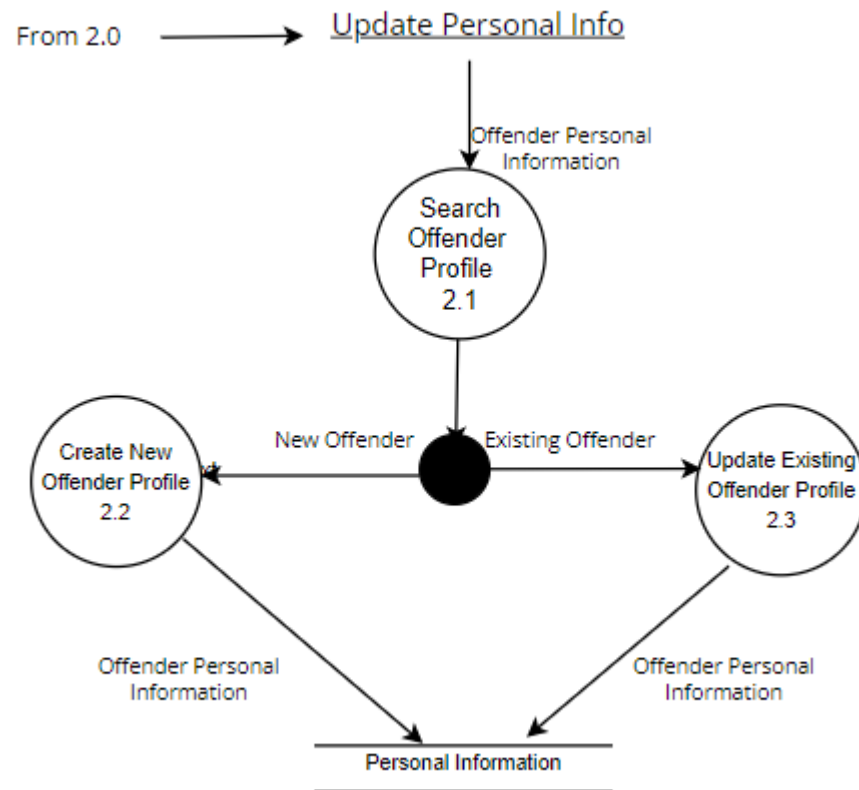


Figure 12 : Level 2.0: DFD

Level 2.1: DFD

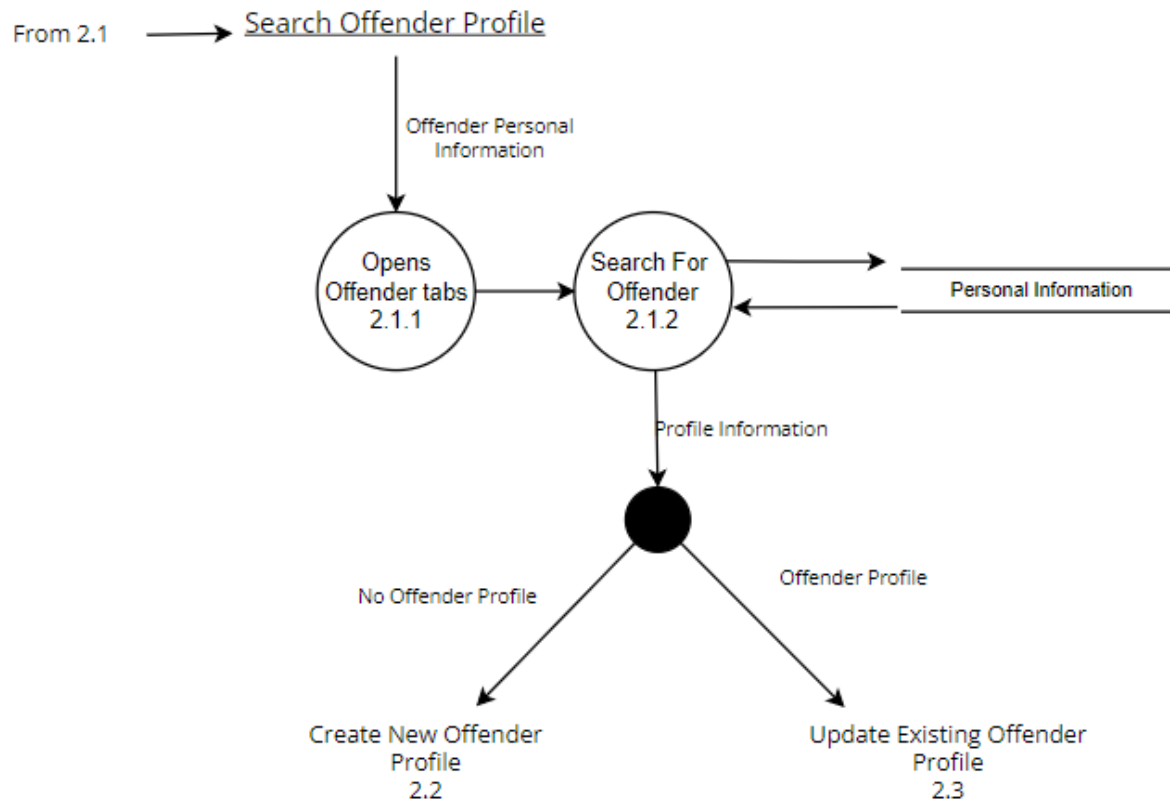


Figure 13 : Level 2.1: DFD

Level 2.2: DFD

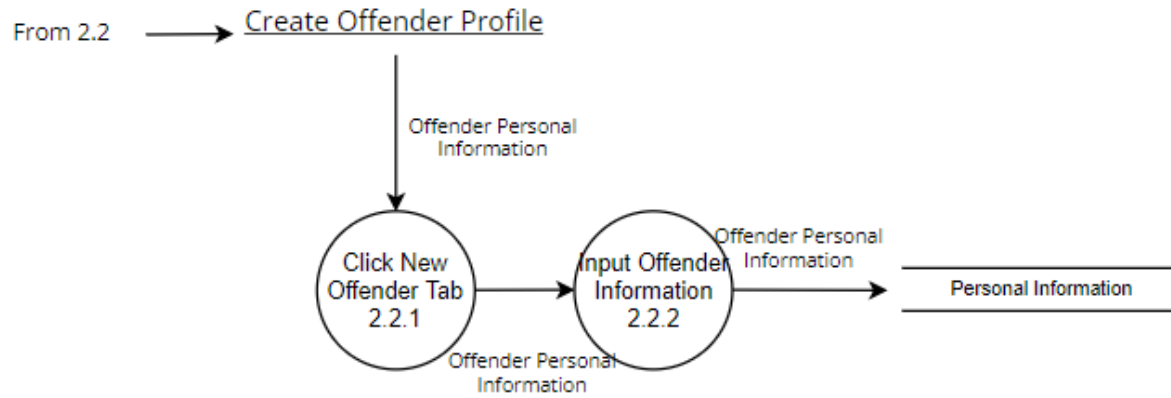


Figure 14 : Level 2.2: DFD

Level 2.3: DFD

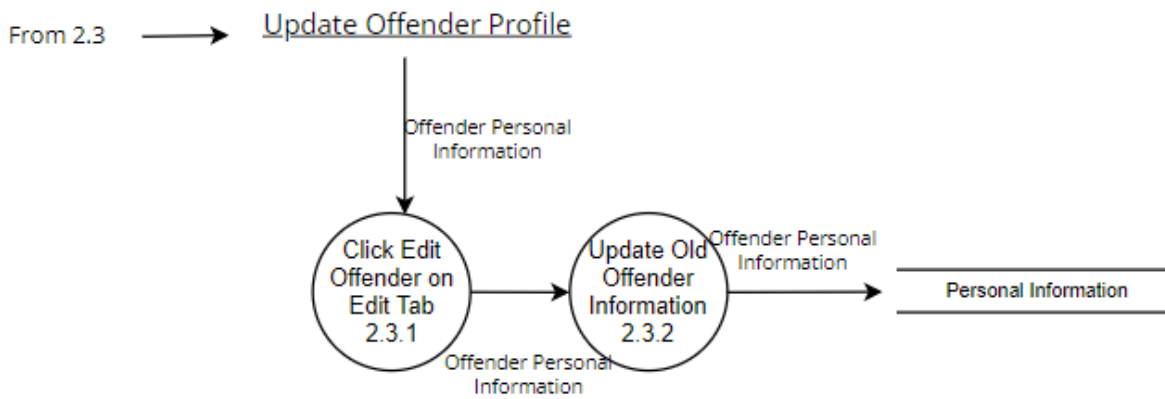


Figure 15 : Level 2.3: DFD

Level 3.0: DFD

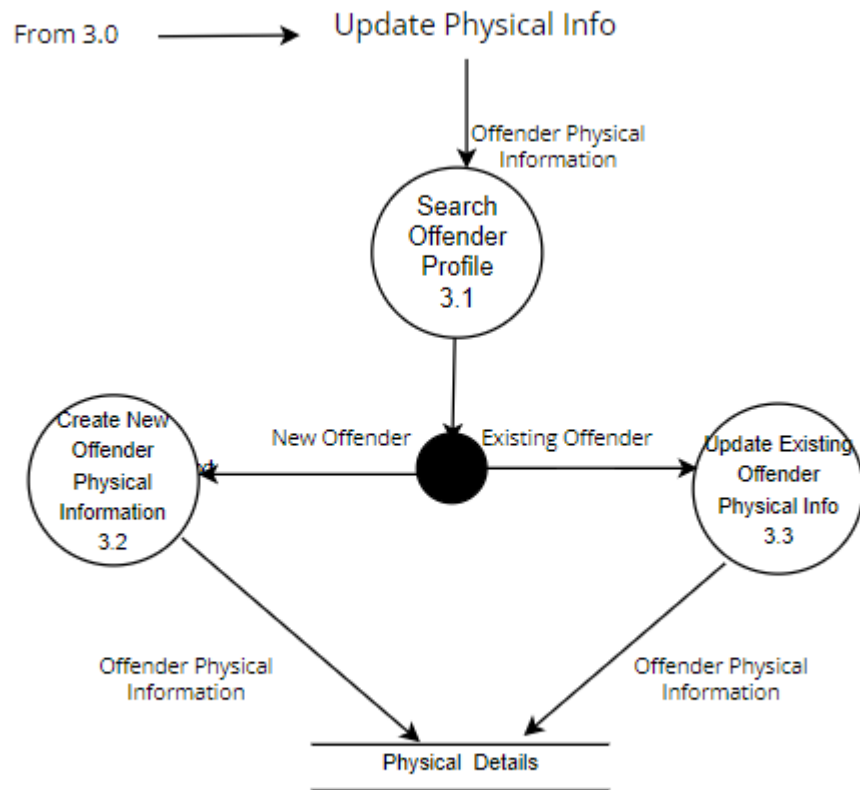


Figure 16 : Level 3.0: DFD

Level 3.1: DFD

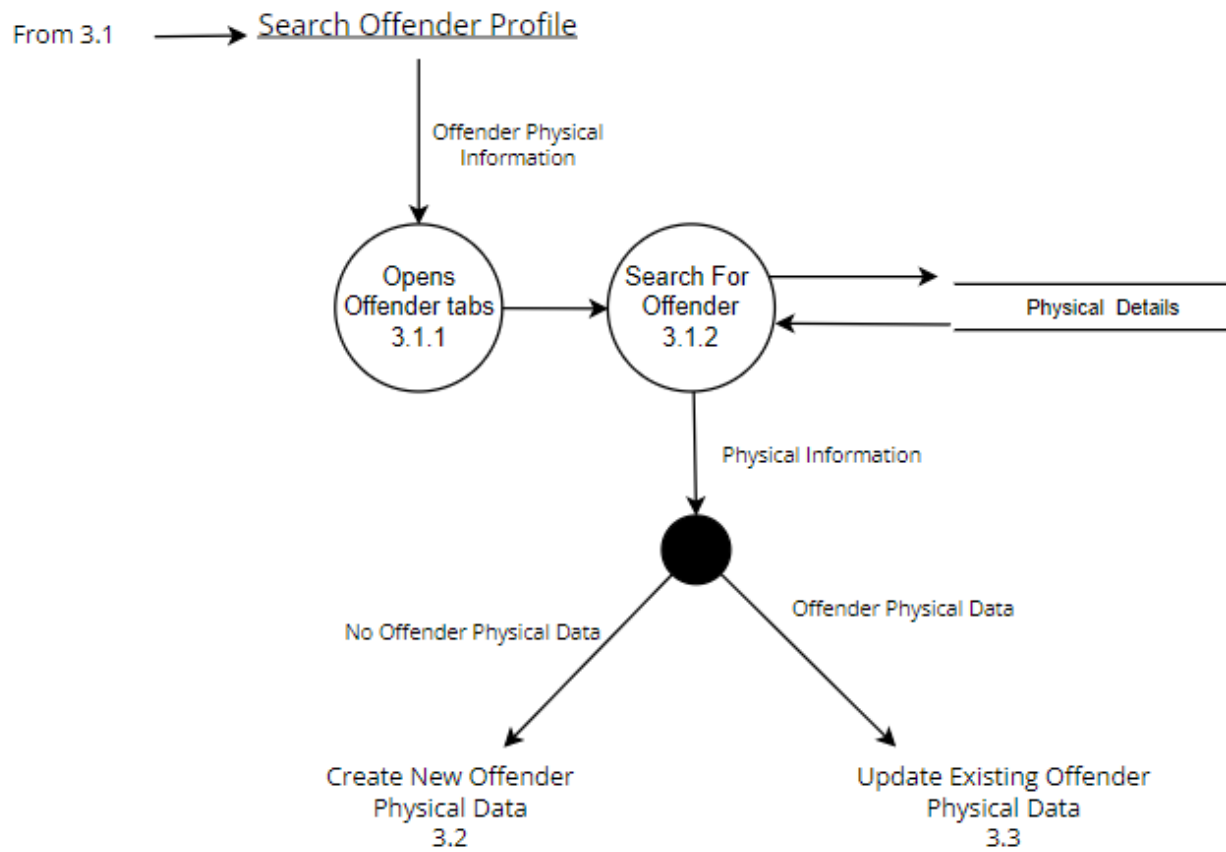


Figure 17 : Level 3.1: DFD

Level 3.2: DFD

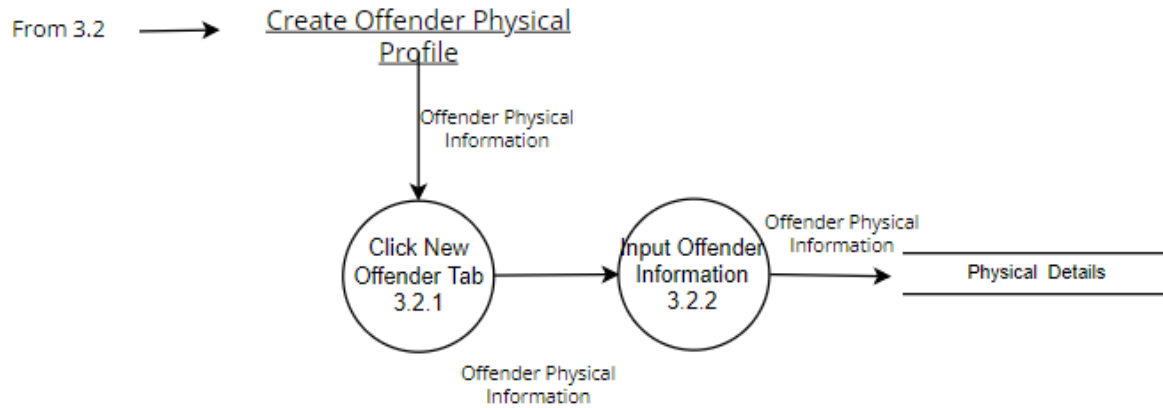


Figure 18: Level 3.2: DFD

Level 3.3: DFD

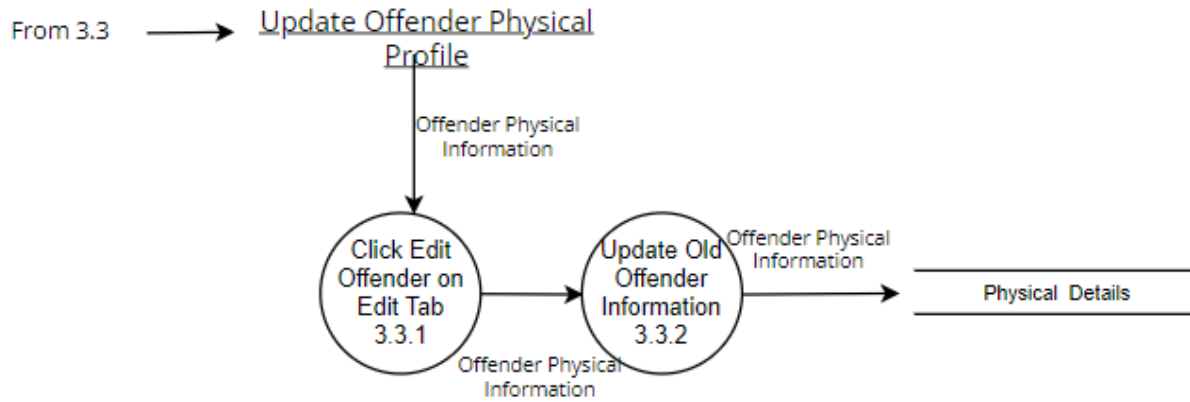


Figure 19 : Level 3.3: DFD

Level 4.0: DFD

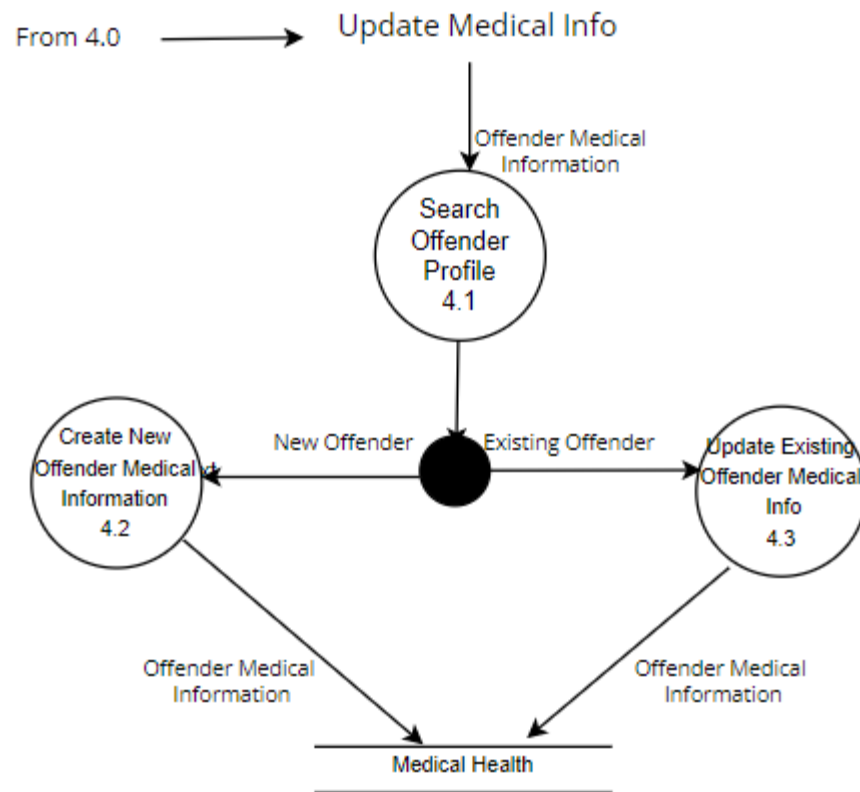


Figure 20 : Level 4.0: DFD

Level 4.1: DFD

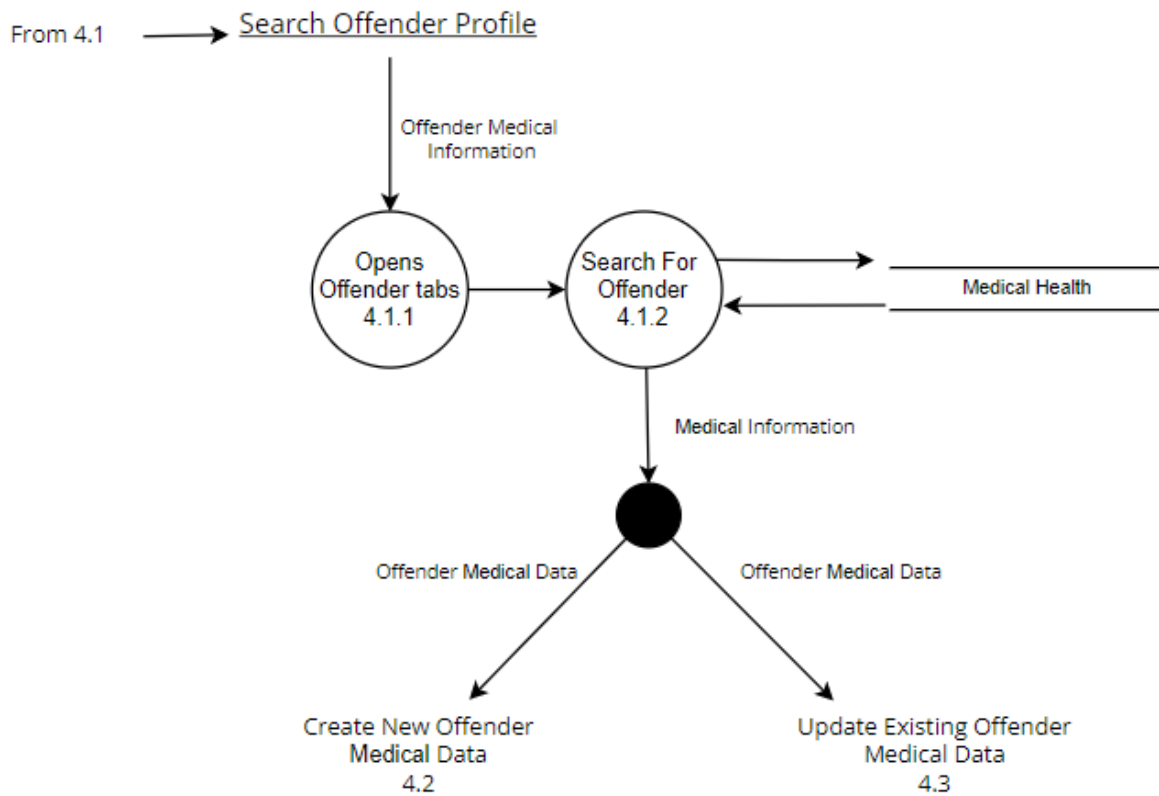


Figure 21 : Level 4.1: DFD

Level 4.2: DFD

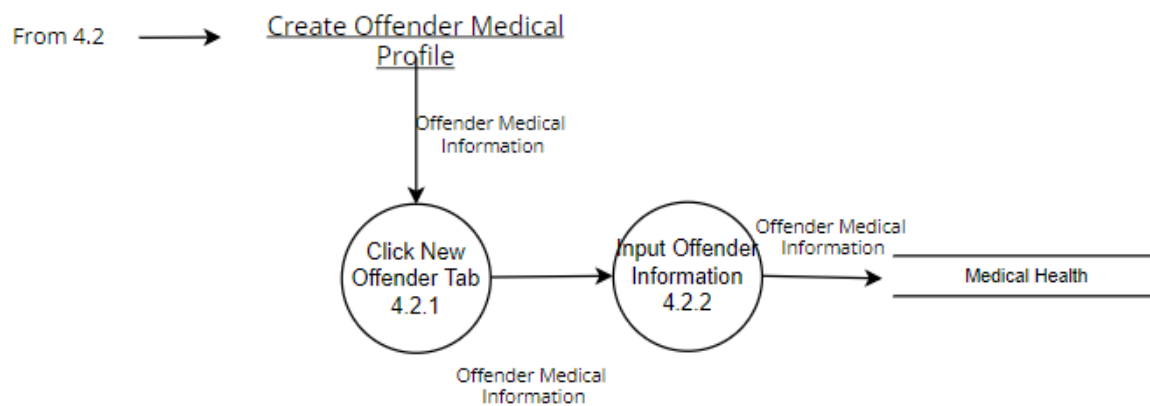


Figure 22 : Level 4.2: DFD

Level 4.3: DFD

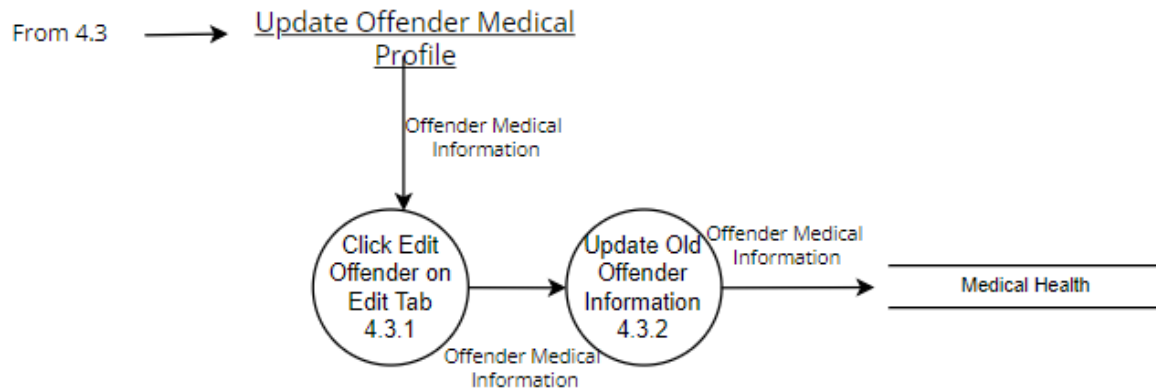


Figure 23 : Level 4.3: DFD

Level 5.0: DFD

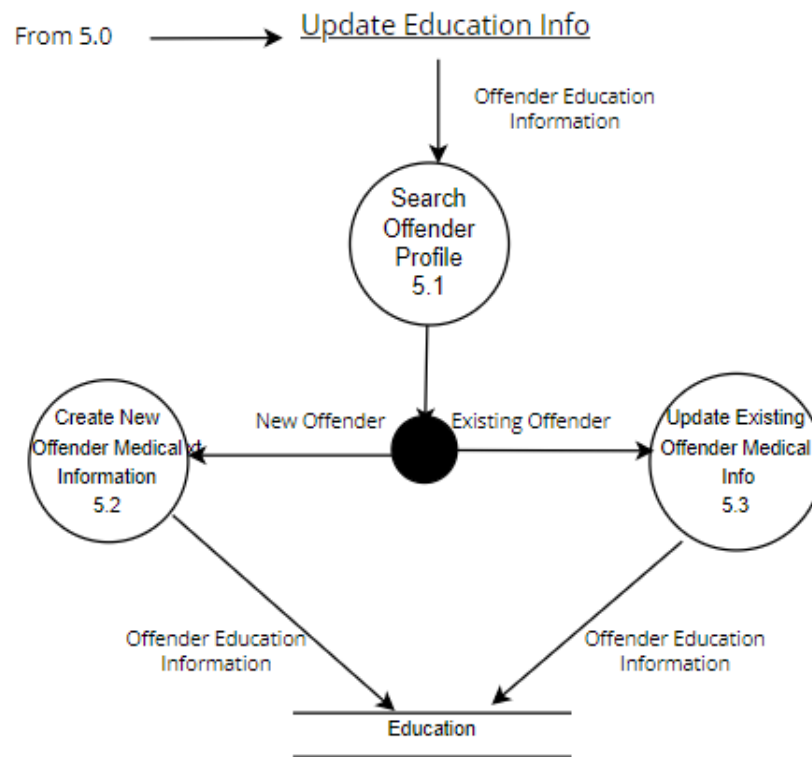


Figure 24 : Level 5.0: DFD

Level 5.1: DFD

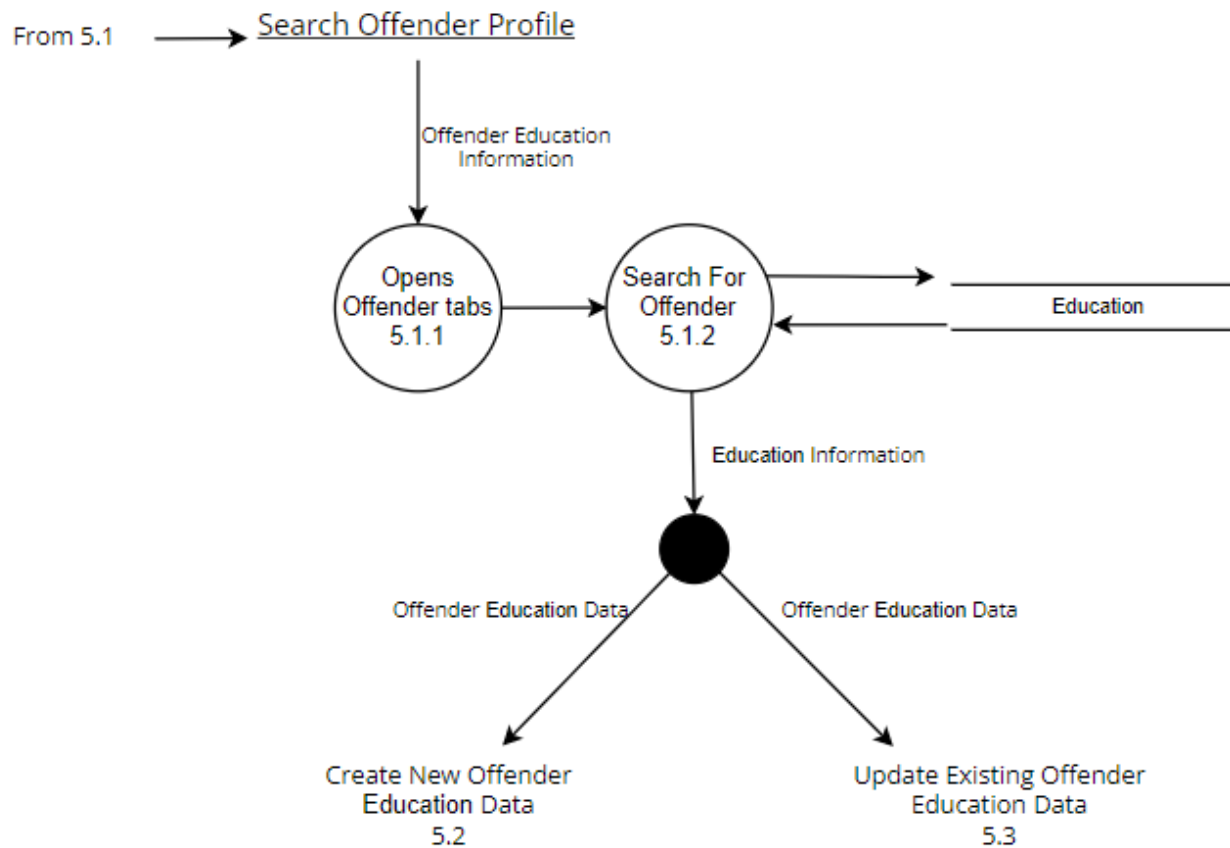


Figure 25 : Level 5.1: DFD

Level 5.2 DFD

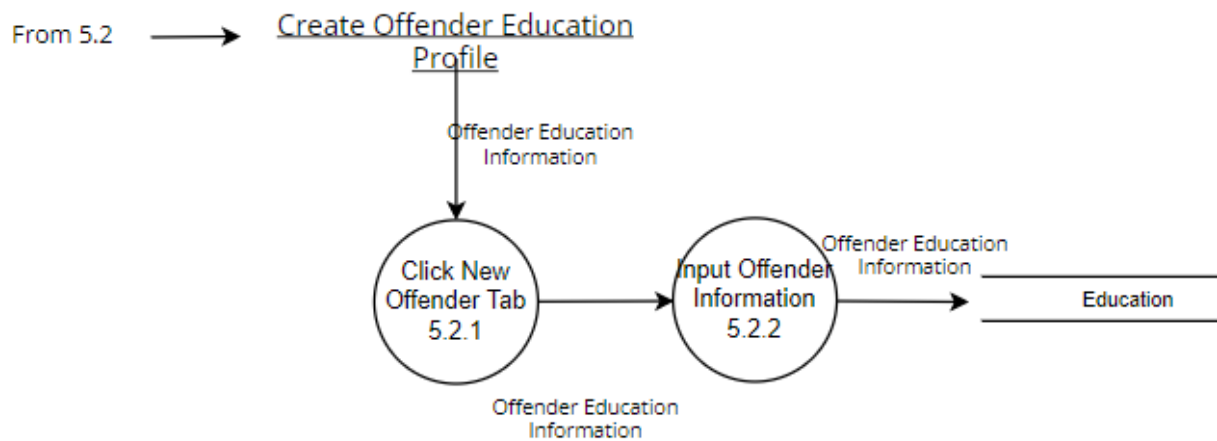


Figure 26 : Level 5.2 DFD

Level 5.3 DFD

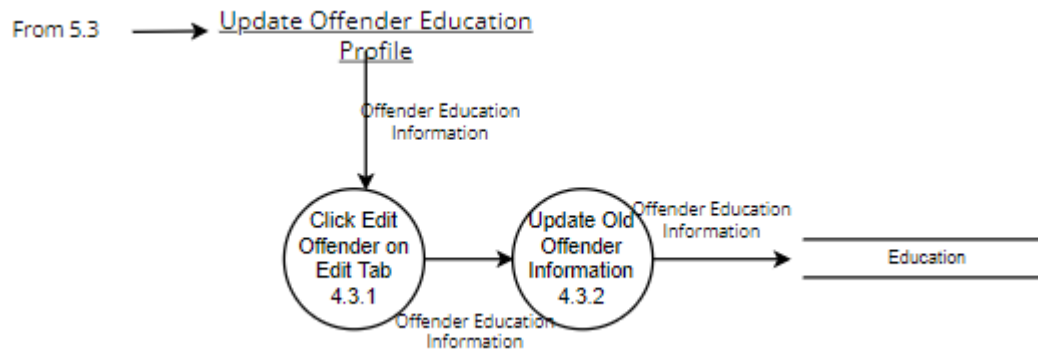


Figure 27 : Level 5.3 DFD

Level 6.0 DFD

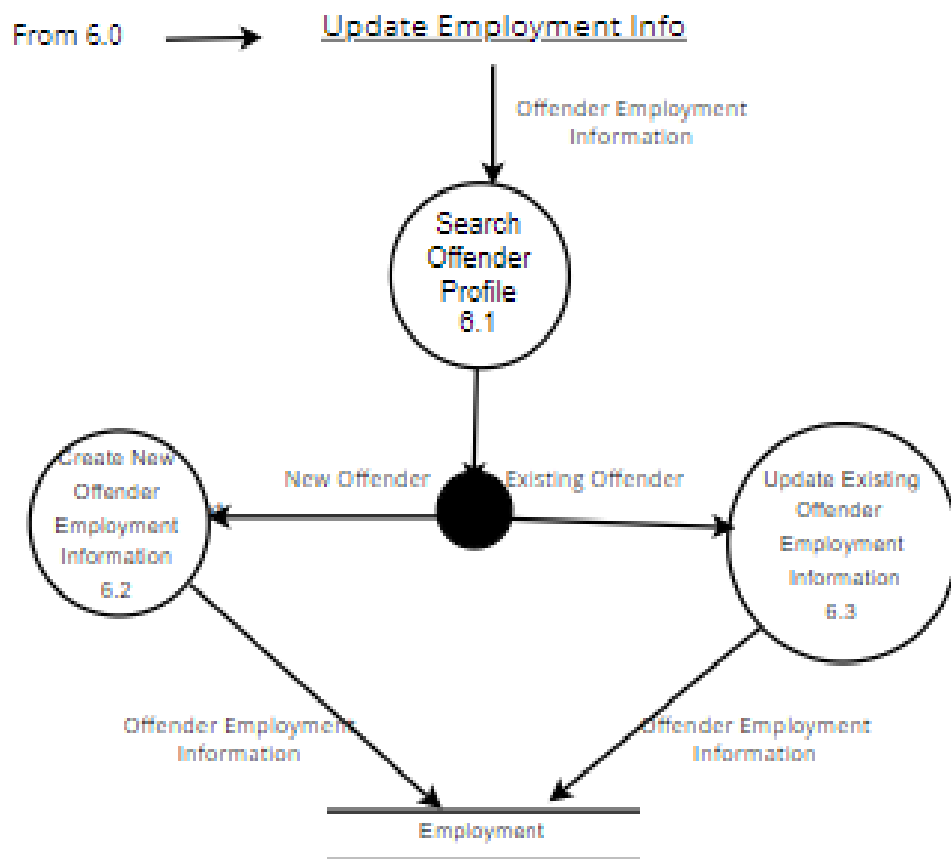


Figure 28 : Level 6.0 DFD

Level 6.1 DFD

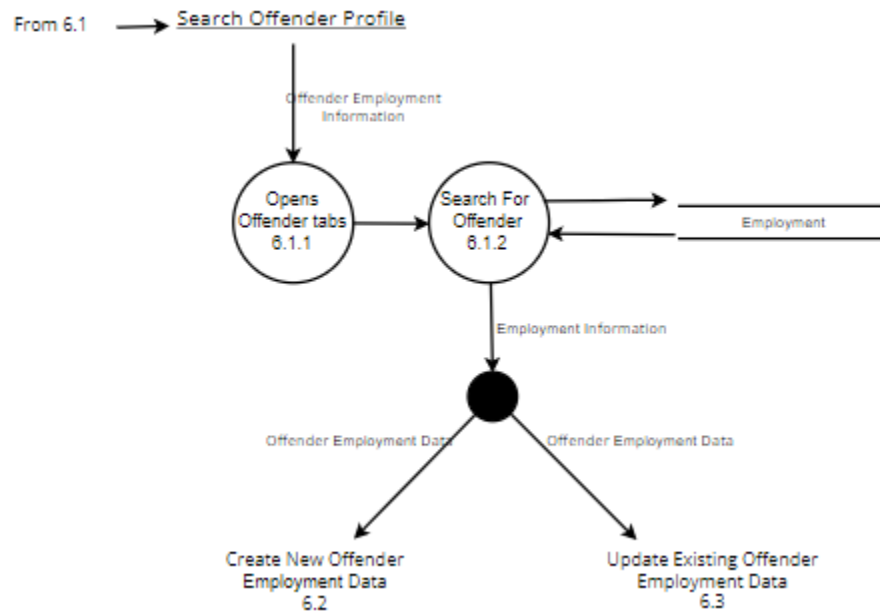


Figure 29 : Level 6.1 DFD

Level 6.2 DFD

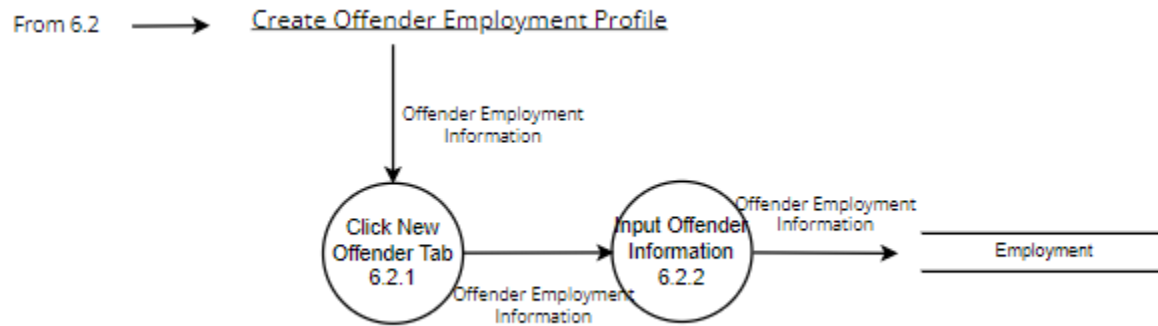


Figure 30 : Level 6.2 DFD

Level 6.3 DFD

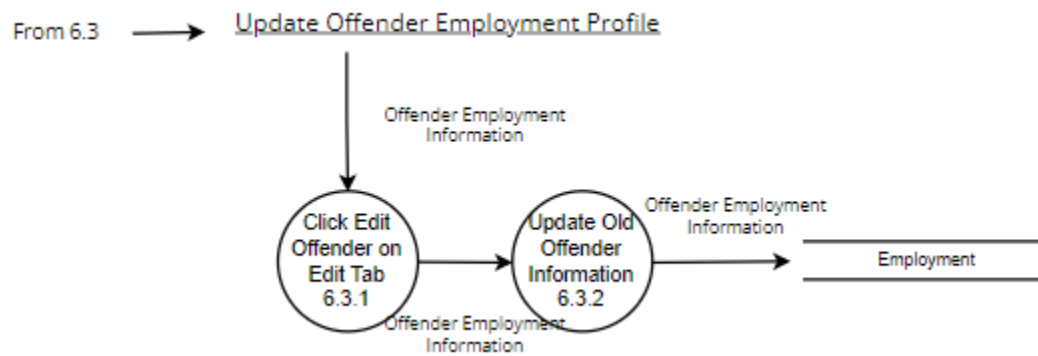


Figure 31 : Level 6.3 DFD

Level 7.0 DFD

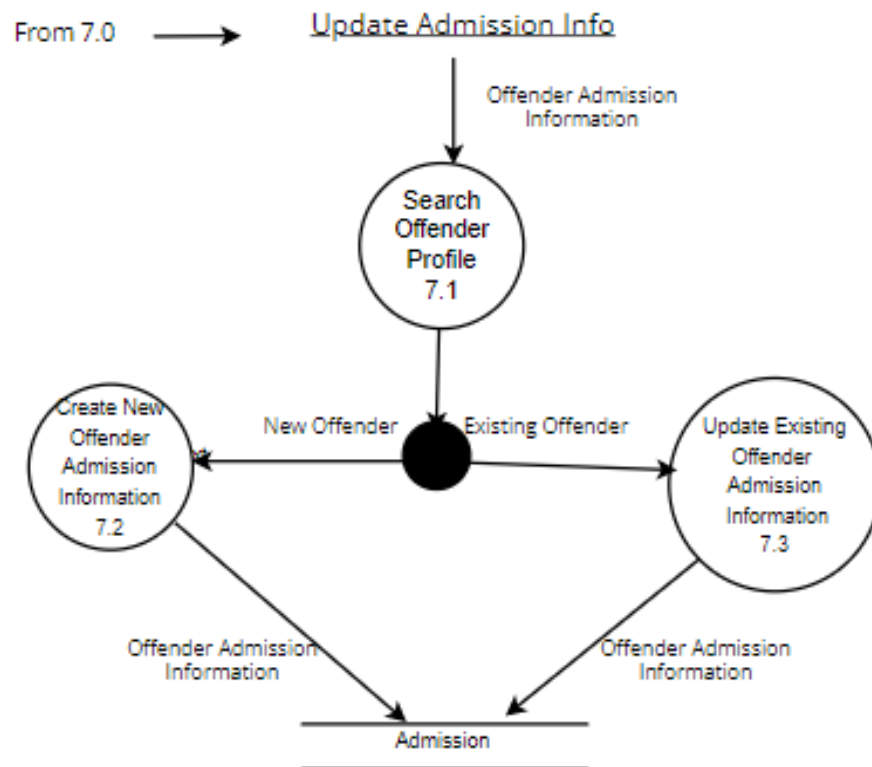


Figure 32 : Level 7.0 DFD

Level 7.1 DFD

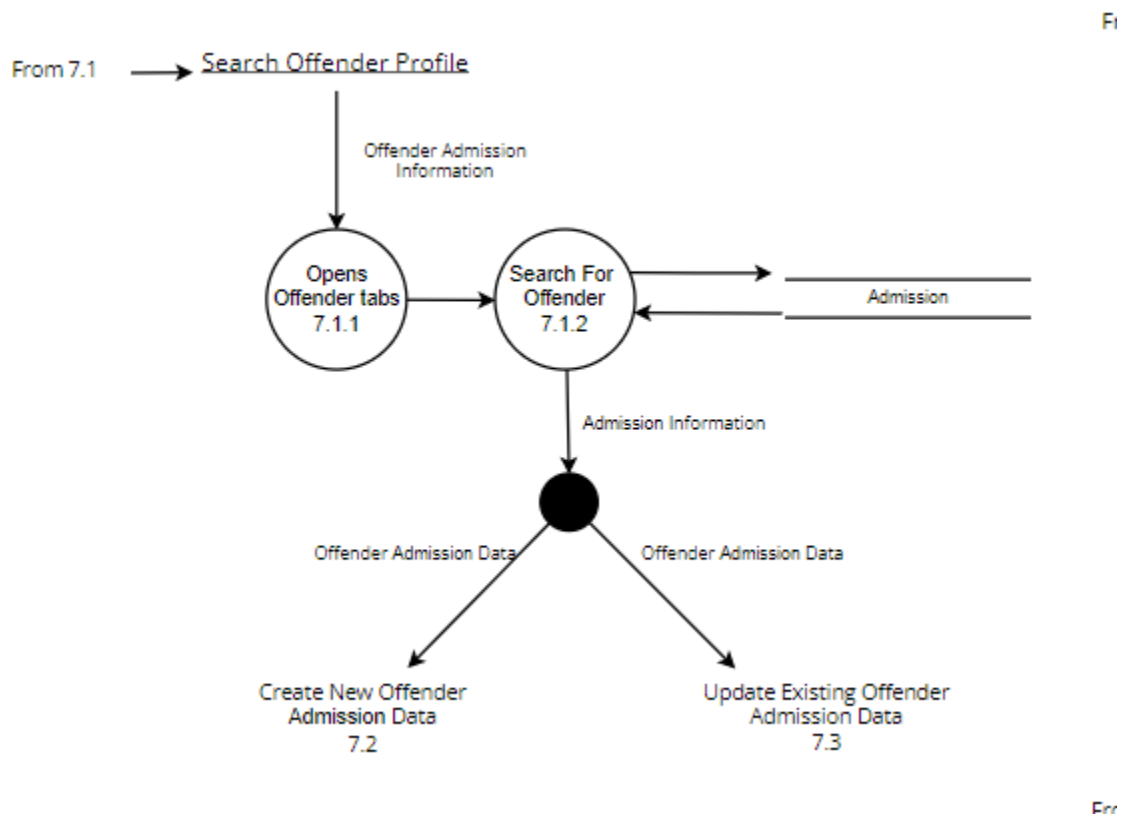


Figure 33 : Level 7.1 DFD

Level 7.2 DFD

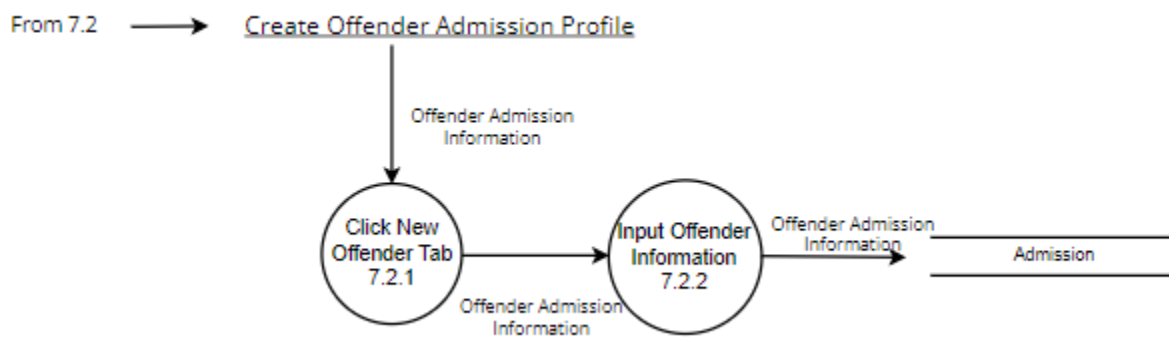


Figure 34 : Level 7.2 DFD

Level 7.3 DFD

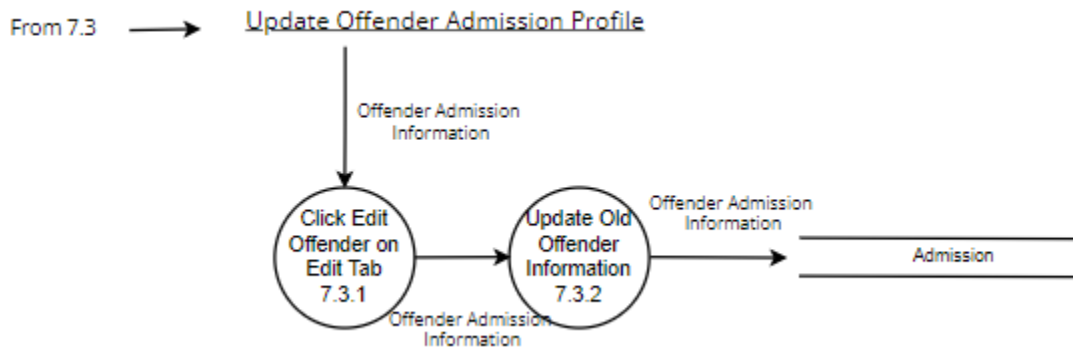


Figure 35 : Level 7.3 DFD

Level 8.0 DFD

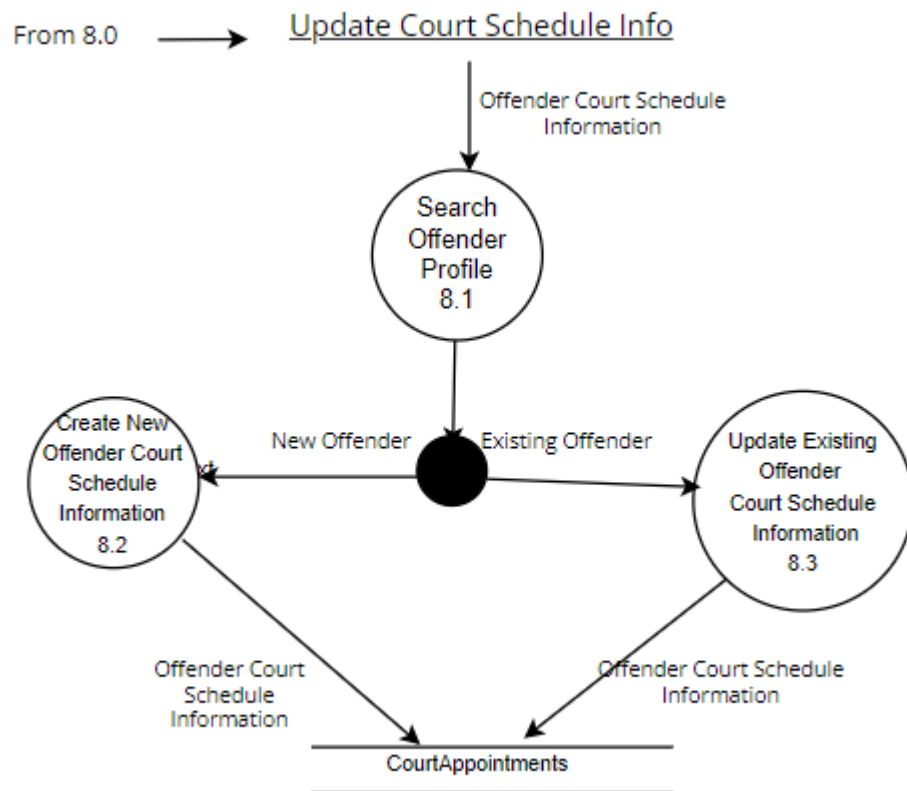


Figure 36 : Level 8.0 DFD

Level 8.1 DFD

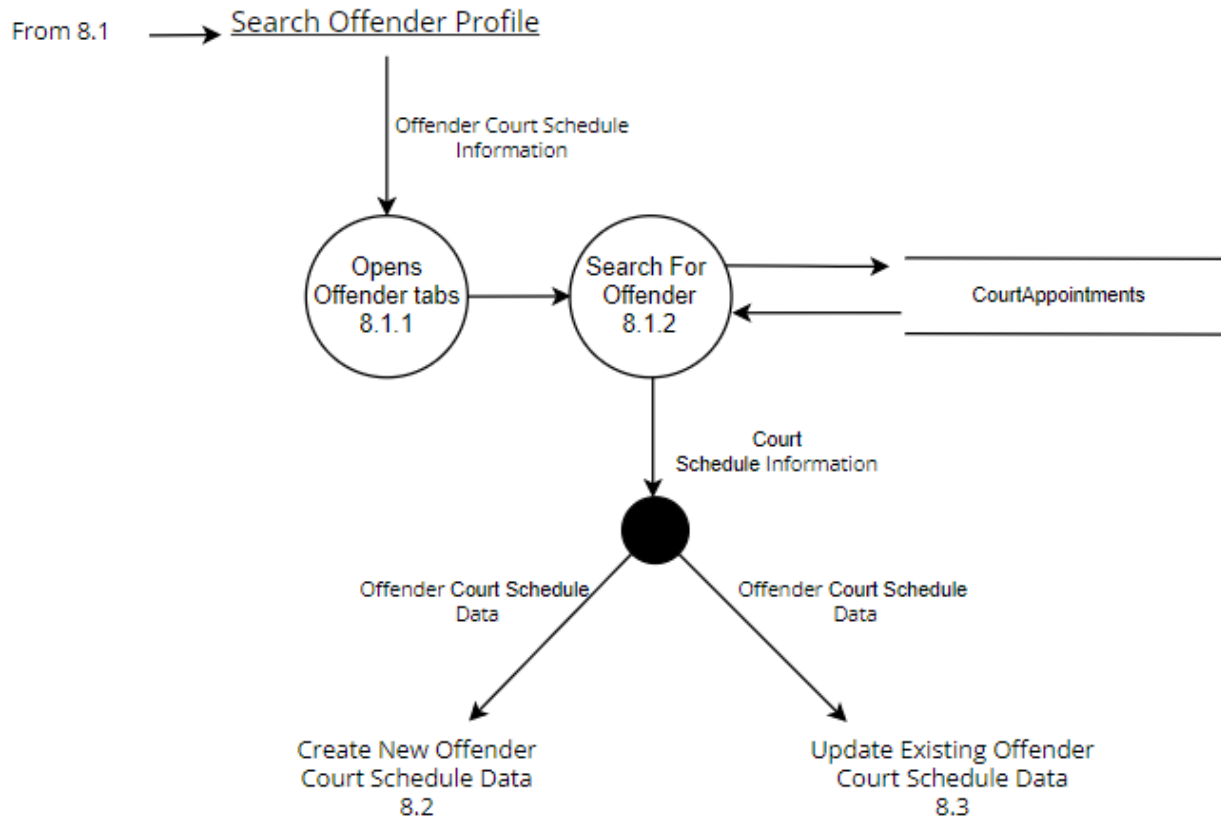


Figure 37 : Level 8.1 DFD

Level 8.2 DFD

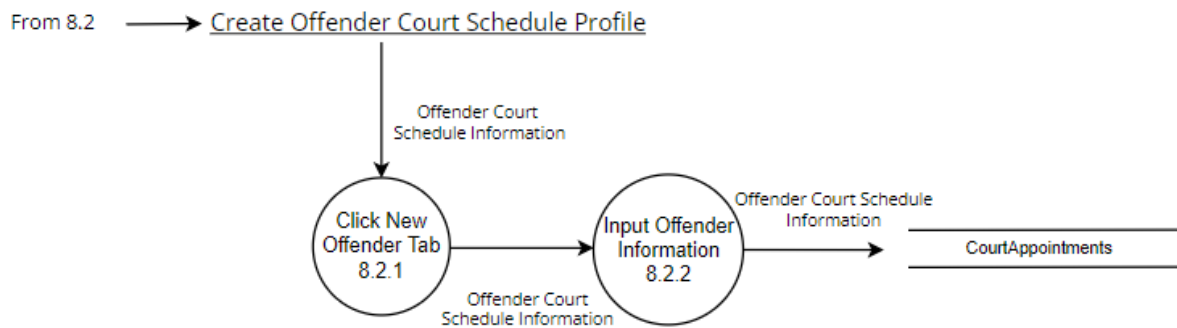


Figure 38 : Level 8.2 DFD

Level 8.3 DFD

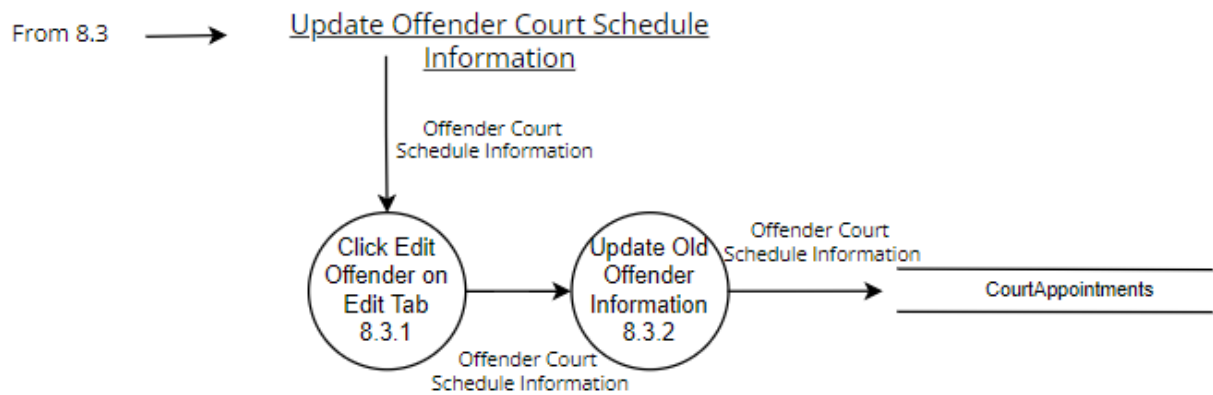


Figure 39 : Level 8.3 DFD

Level 9.0 DFD

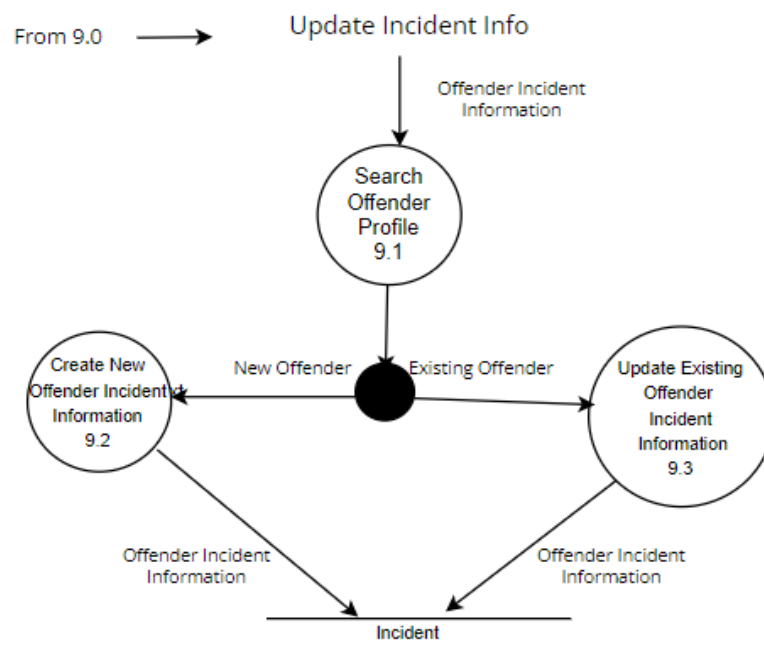


Figure 40 : Level 9.0 DFD

Level 9.1 DFD

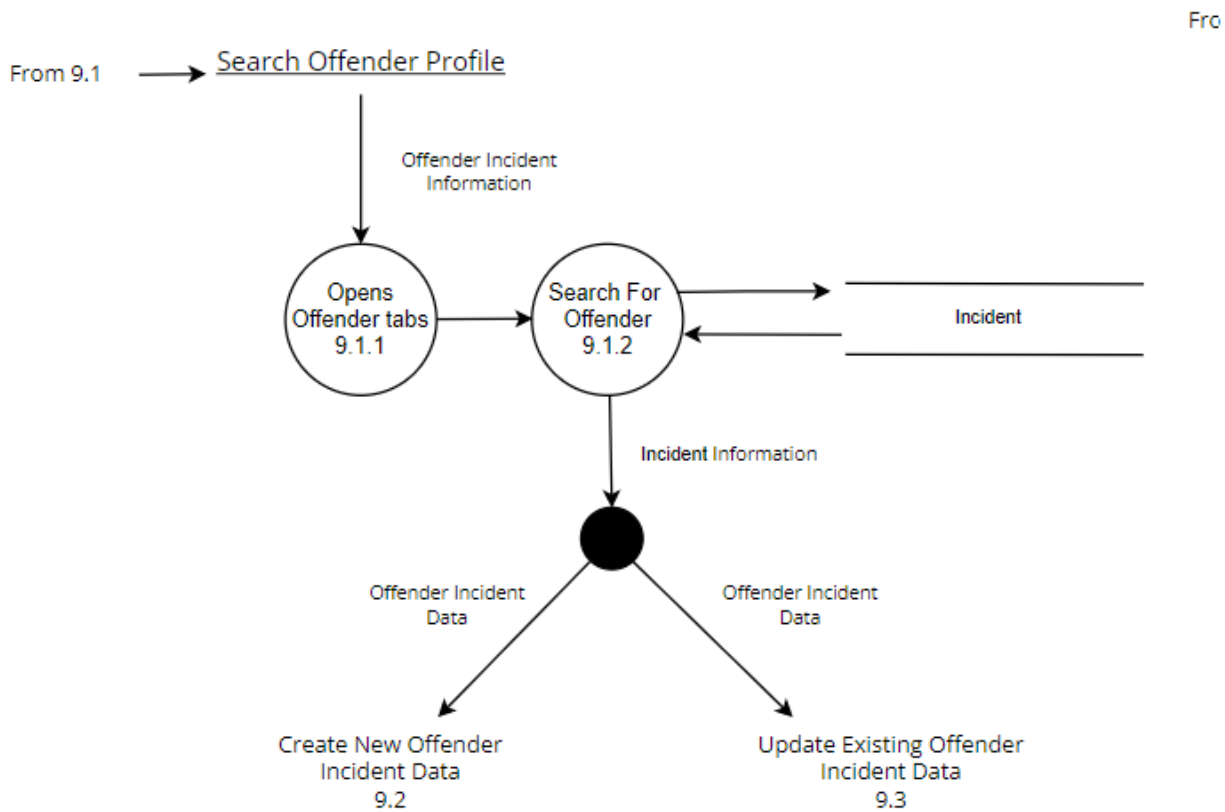


Figure 41 : Level 9.1 DFD

Level 9.2 DFD

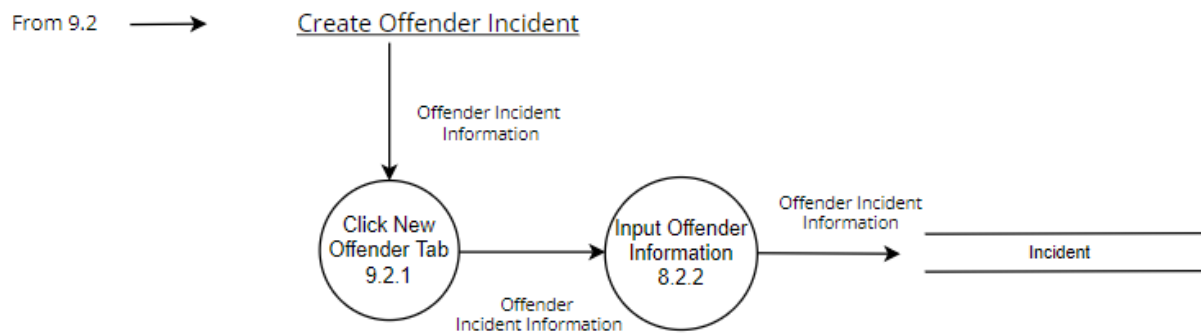


Figure 42 : Level 9.2 DFD

Level 9.3 DFD

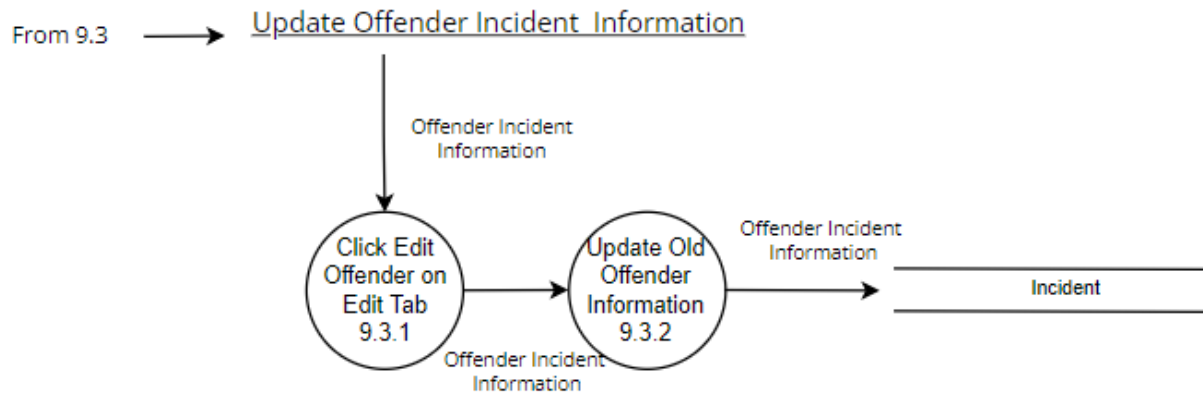


Figure 43 : Level 9.3 DFD

Level 10.0 DFD

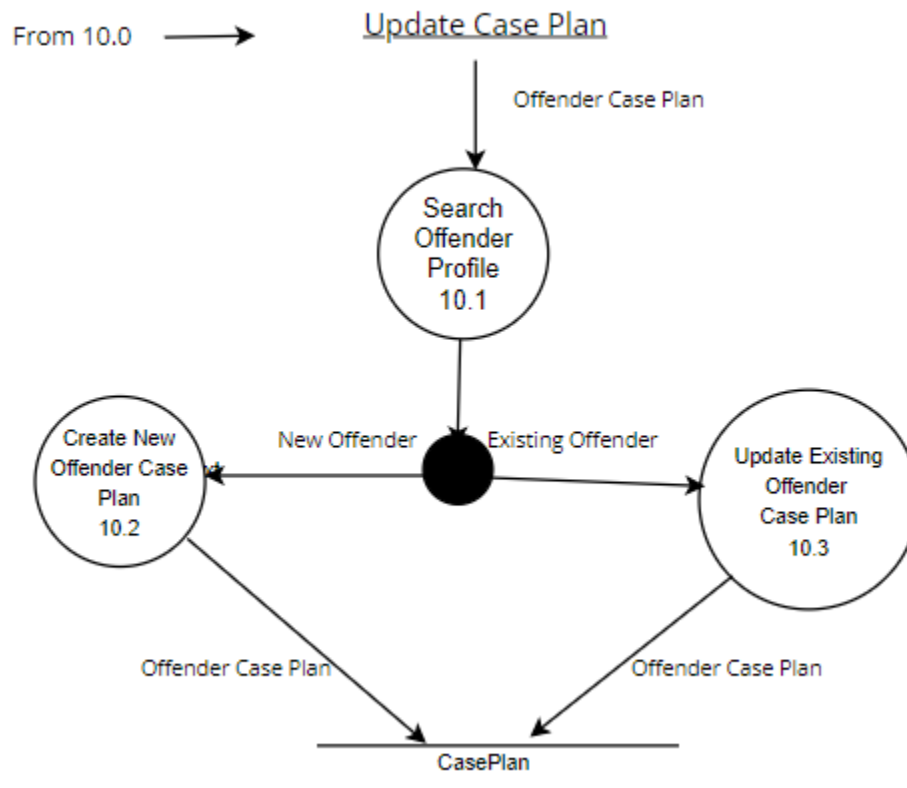


Figure 44 : Level 10.0 DFD

Level 10.1 DFD

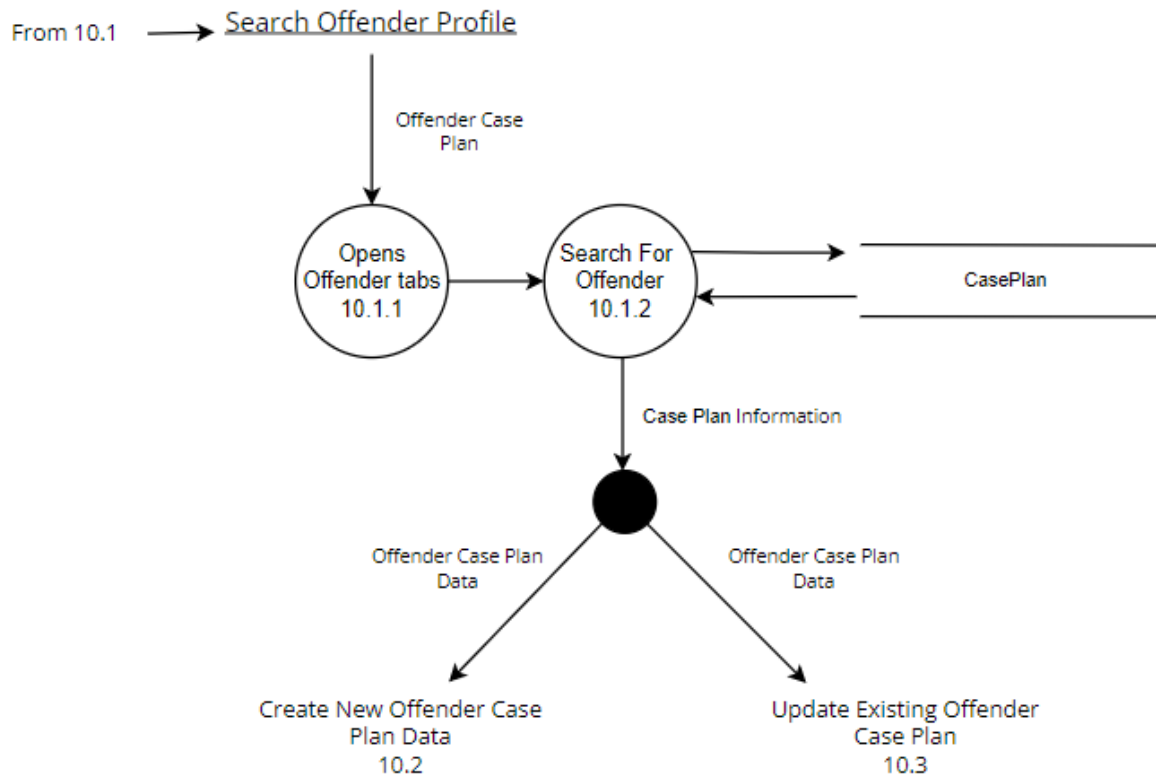


Figure 45 : Level 10.1 DFD

Level 10.2 DFD

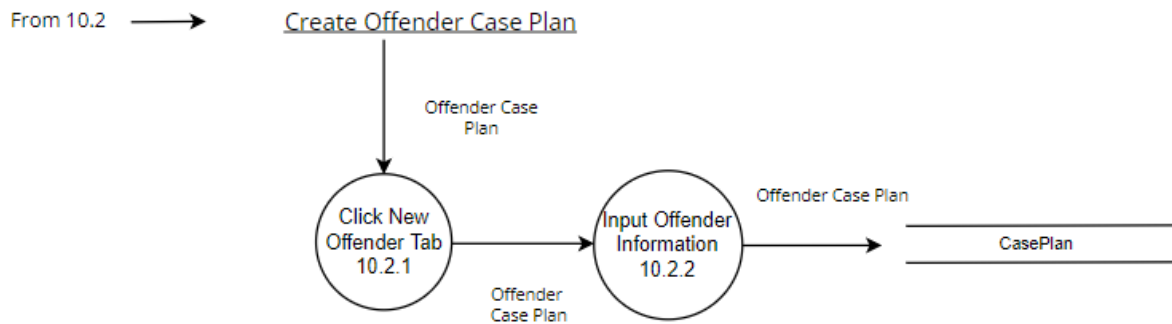


Figure 46 : Level 10.2 DFD

Level 10.3 DFD

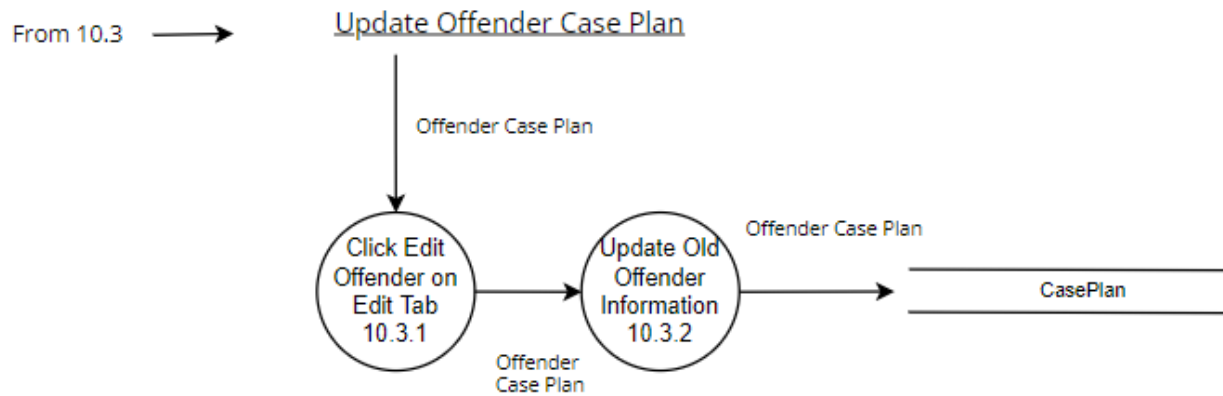


Figure 47 : Level 10.3 DFD

Level 11.0 DFD

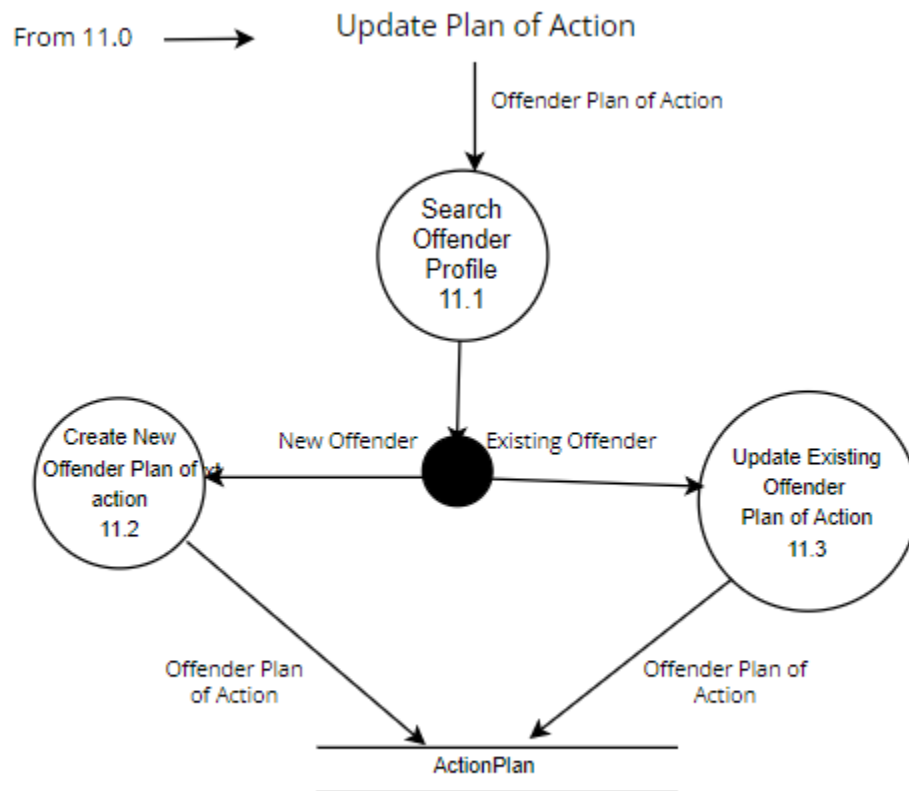


Figure 48 : Level 11.0 DFD

Level 11.1 DFD

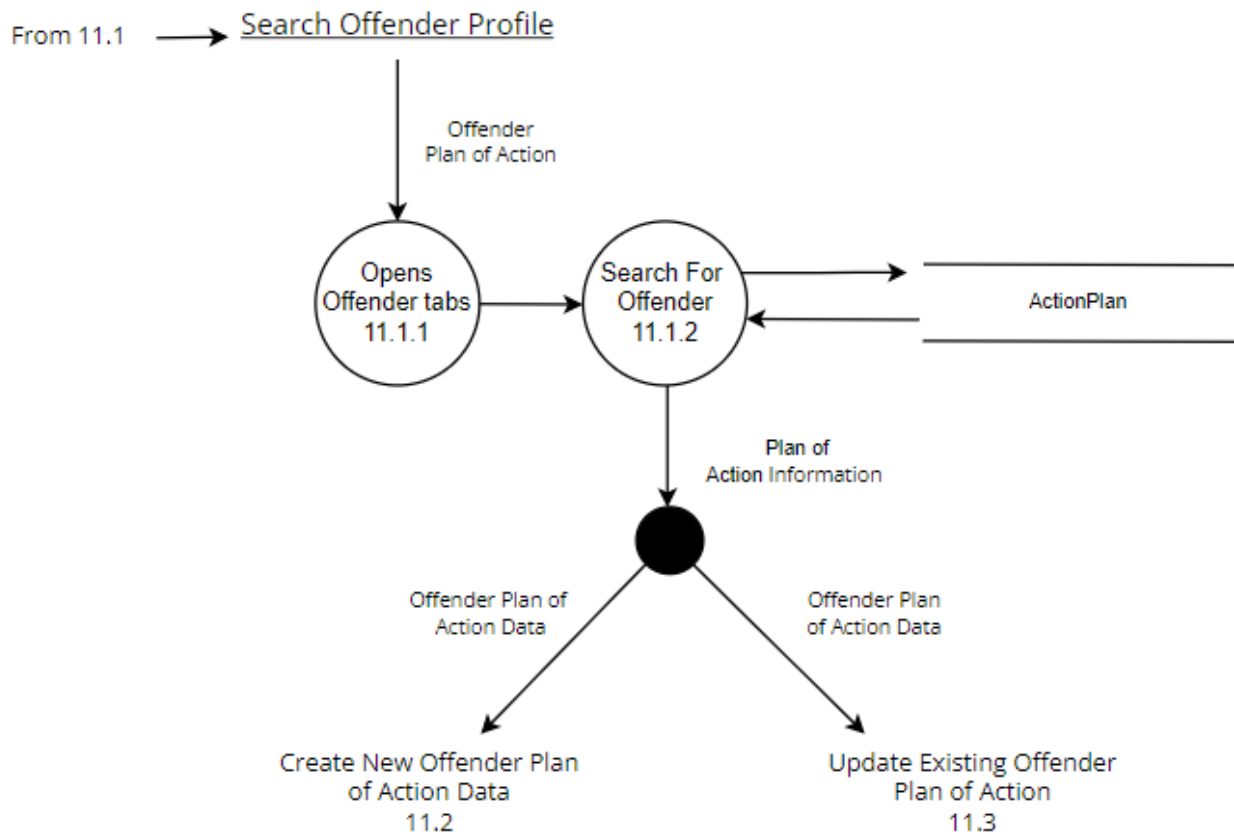


Figure 49 : Level 11.1 DFD

Level 11.1 DFD

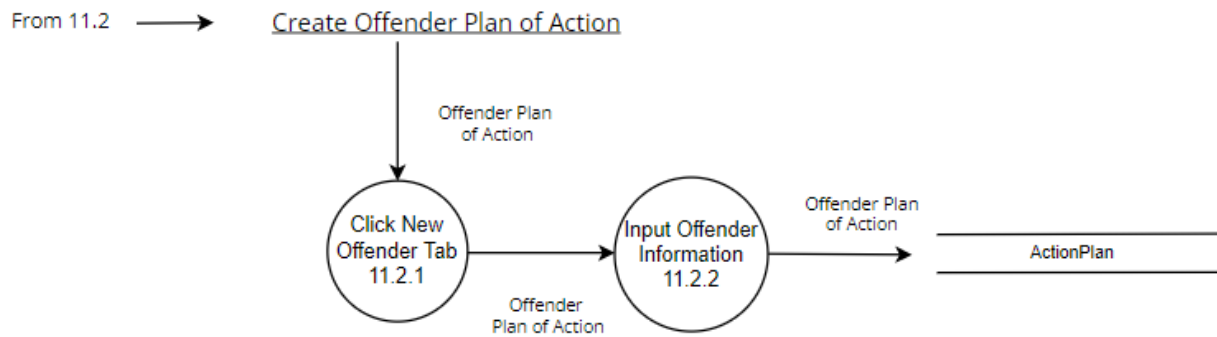


Figure 50 : Level 11.1 DFD

Level 11.2 DFD

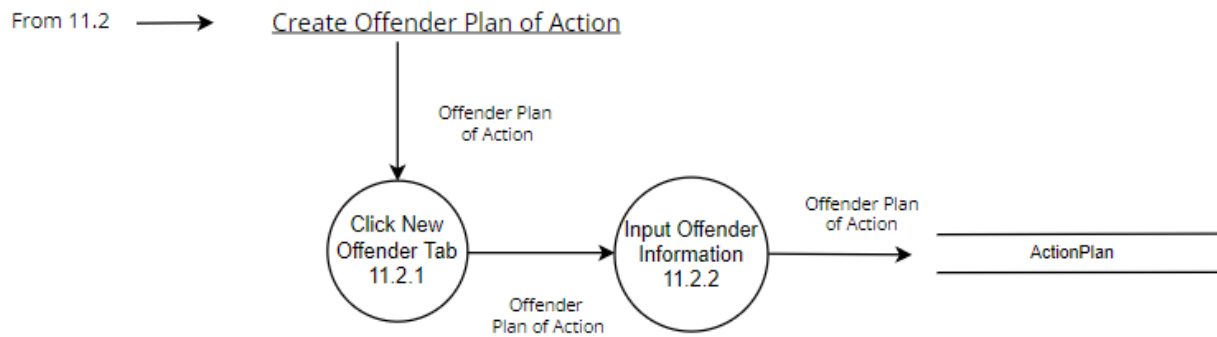


Figure 51: Level 11.2 DFD

Level 11.3 DFD

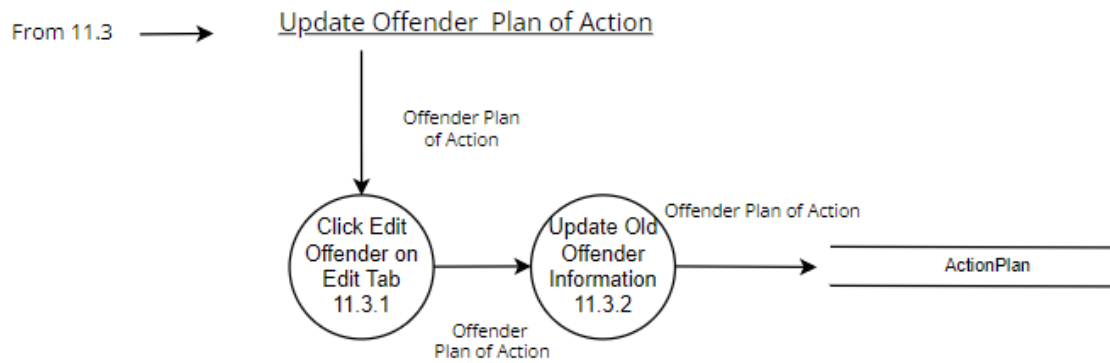


Figure 52 : Level 11.3 DFD

Level 12.0 DFD

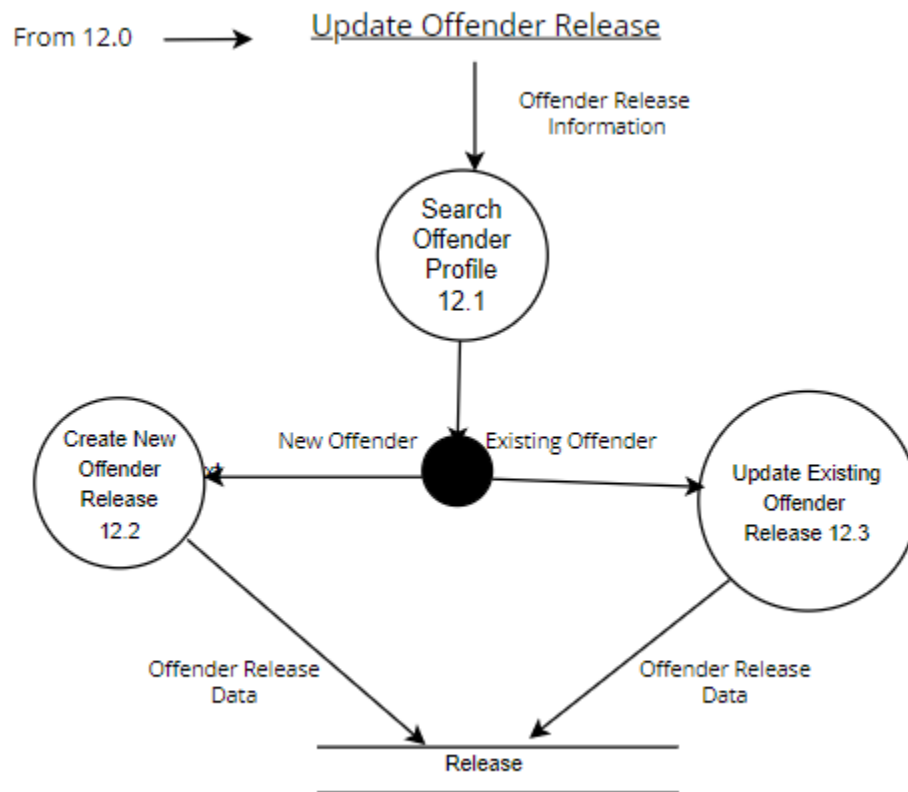


Figure 53 : Level 12.0 DFD

Level 12.1 DFD

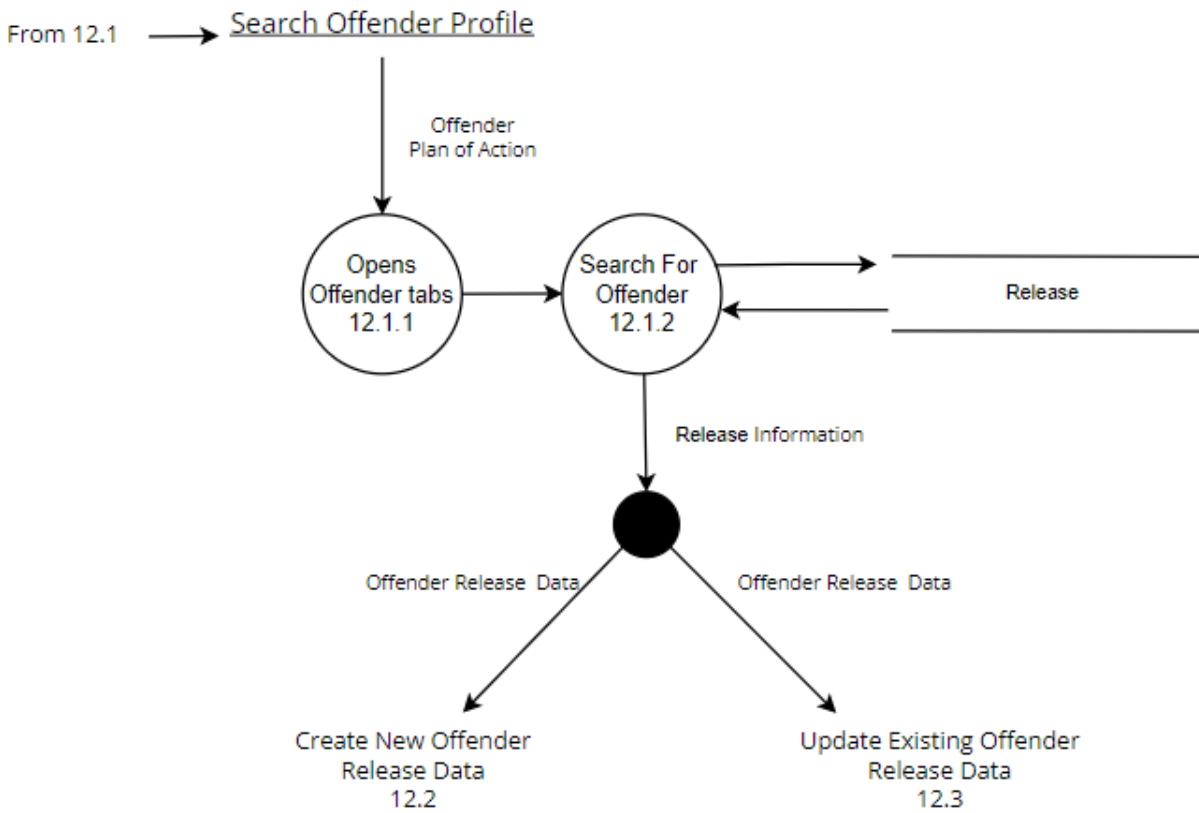


Figure 54 : Level 12.1 DFD

Level 12.2 DFD

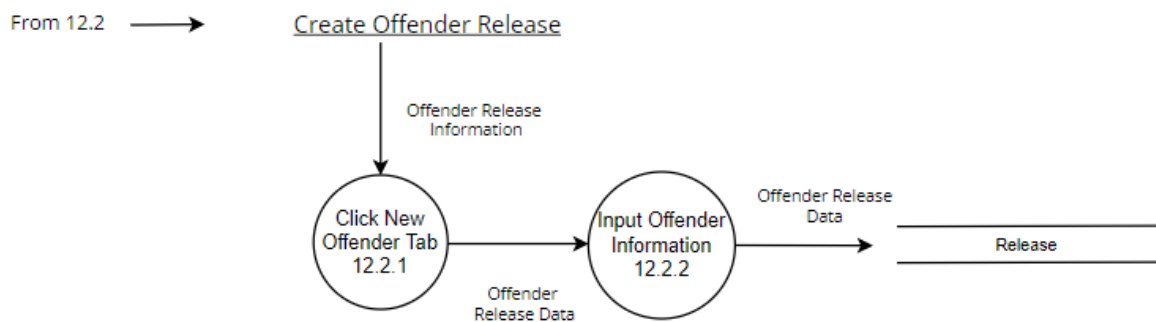


Figure 55 : Level 12.2 DFD

Level 12.3 DFD

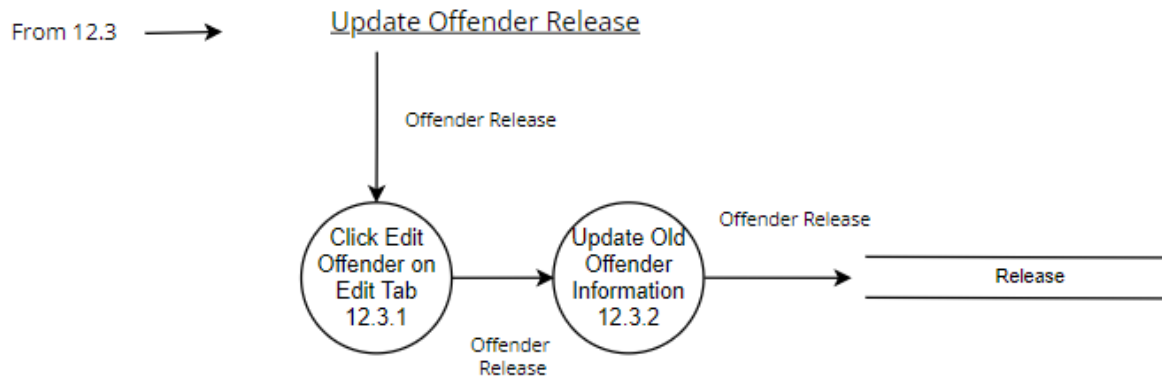


Figure 56 : Level 12.3 DFD

6.2.3 Software Structure chart showing module hierarchy

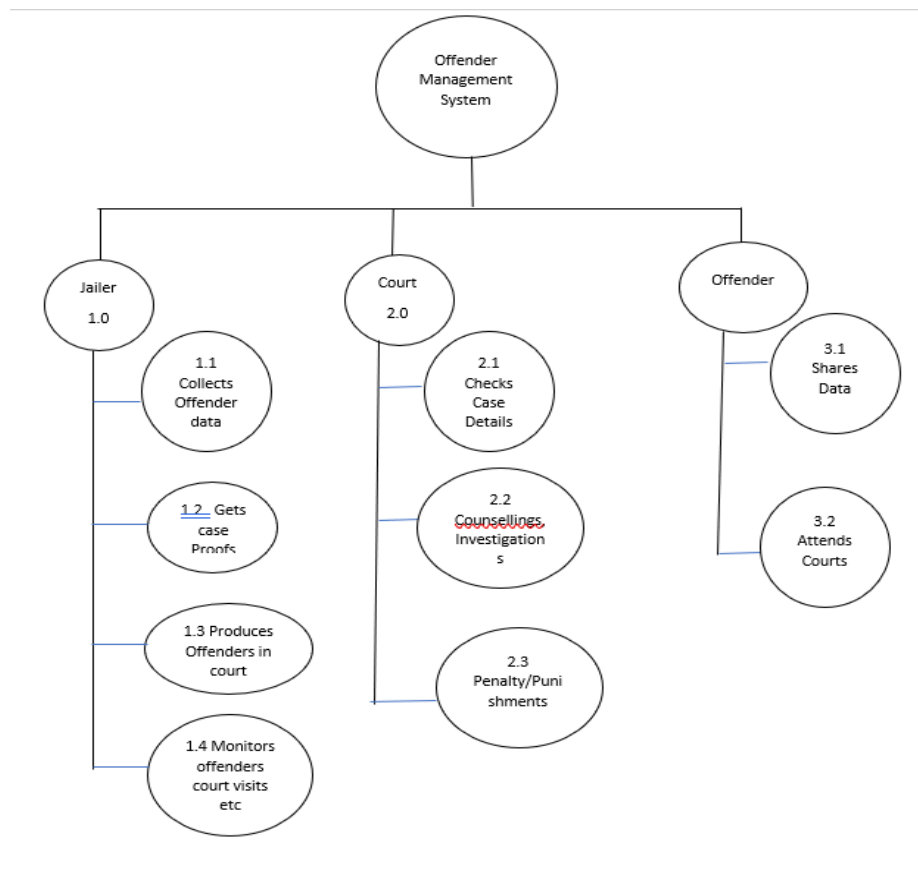


Figure 57 : Software Structure chart showing module hierarchy.

Fig: Software Structure chart showing module hierarchy

6.2.4. Use-Case model diagram

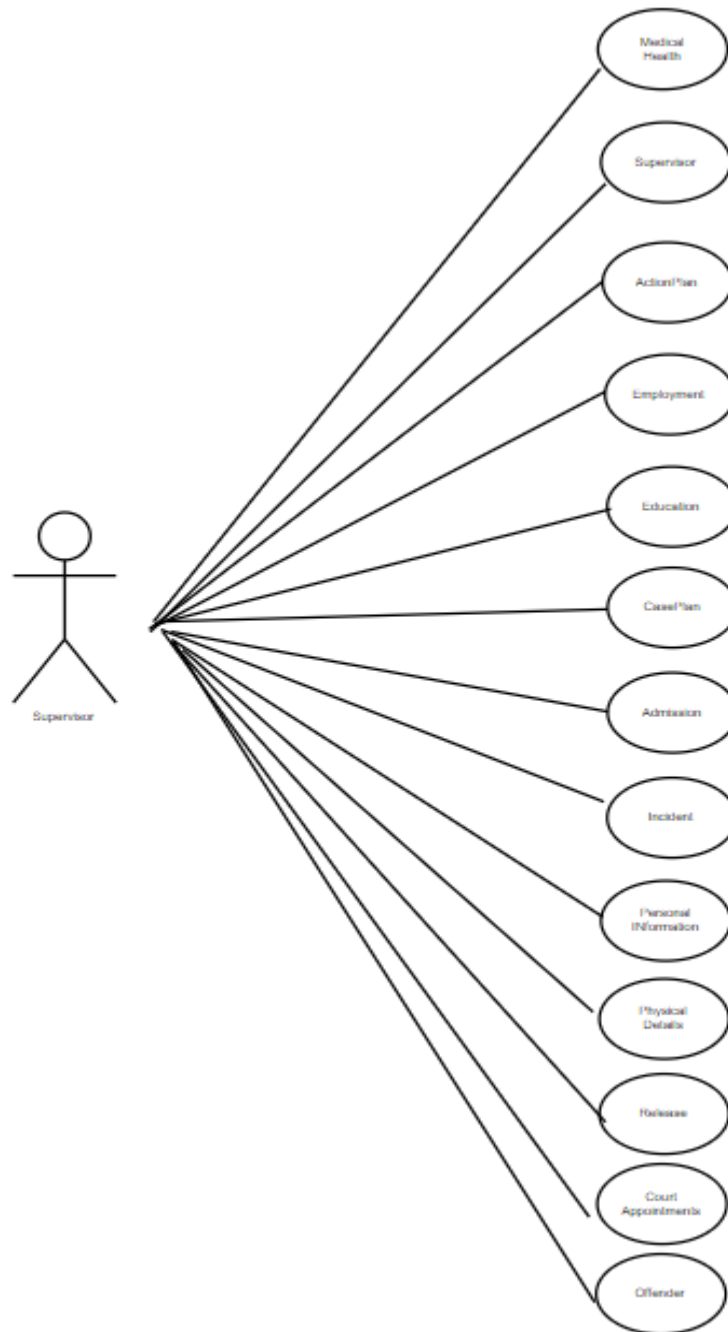


Figure 58 : Use-Case model diagram.

Fig: Use case Diagram

6.2.5. Use case Narrative for this application

Correctional Facility Management System (CFMS)		
Authors: Naresh, Vikram Singh, Jagadeshwar		Date:
		Version: 1.0
Use-Case Name:	Education	Use-Case Type Business Requirements:
Use-Case ID:	CFMS-1.00	
Priority:	High	
Source:	Offender	
Primary Business Actor:	Jailer	
Other Participating Actors:	Offender	
Other Interested Stakeholders:	Jailer	
Description:	This use-case describes the necessary steps for the jailer to add the education details of each of the offender	
Precondition:	The offender need to be confirmed by the court to be in that respective Jailer control	
Trigger:	This use case is initiated by the immediate need to add the educational details into the system.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Jailer needs to add the offender details	Step 2: If the provided details are respective to the offender, then they are stored into the database
Alternate Courses:	Alt-Step 1: Checks the reasons of rejection and comes up clearing them	
Conclusion:	This use case concludes whether or not the education details of offender added to the system	
Postcondition:	Offender educational details were added successfully into the system	
Business Rules:	N/A	
Implementation Constraints and Specifications:	Provider has full view privileges.	
Assumptions:	Credentials are valid.	
Open Issues:	N/A	

Table 15 : Use case Narrative for this application.

Correctional Facility Management System (CFMS)		
Authors: Naresh, Vikram Singh, Jagadeshwar		Date:
		Version: 1.0
Use-Case Name:	Employment	Use-Case Type Business Requirements:
Use-Case ID:	CFMS-2.00	
Priority:	High	
Source:	Offender	
Primary Business Actor:	Jailer	
Other Participating Actors:	Offenders	
Other Interested Stakeholders:	Offenders	
Description:	This use-case describes the necessary steps to add the employment details of the offender	
Precondition:	The offender need to be confirmed by the court to be in that respective Jailer control	
Trigger:	This use case is initiated by the immediate need to add the employment details into the system.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Jailer needs to add the offender employment details	Step 2: If the provided details are respective to the offender, then they are stored into the database
Alternate Courses:	Alt-Step 1: Checks the reasons of rejection and comes up clearing them	
Conclusion:	This use case concludes whether or not the employment details of offender added to the system	
Postcondition:	Offender employment details were added successfully into the system	
Business Rules:	N/A	
Implementation Constraints and Specifications:	Provider has full view privileges.	
Assumptions:	Credentials are valid.	
Open Issues:	N/A	

Table 16 : Use case 2 Narrative for this application

Correctional Facility Management System (CFMS)		
Authors: Naresh, Vikram Singh, Jagadeshwar		Date:
		Version: 1.0
Use-Case Name:	Admission	Use-Case Type Business Requirements:
Use-Case ID:	CFMS-3.00	
Priority:	High	
Source:	Offender	
Primary Business Actor:	Jailer	
Other Participating Actors:	Offender	
Other Interested Stakeholders:	Jailer	
Description:	This use-case describes the necessary steps to add the Offender personal details and case details to the management system.	
Precondition:	The jailer needs to register into the system to add the details of offender.	
Trigger:	This use case is initiated by the immediate need to register into the system.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Fills the offender details in the Registration Form.	Step 2: Registers successfully if valid details are provided , else rejects the user registration.
Alternate Courses:	Alt-Step 1: Checks the reasons of rejection and comes up clearing them	
Conclusion:	This use case concludes how to add the personal details and case details of the offender into the system.	
Postcondition:	Offender data successfully added into the system	
Business Rules:	N/A	
Implementation Constraints and Specifications:	Provider has full view privileges.	
Assumptions:	Credentials are valid.	
Open Issues:	N/A	

Table 17 : Use case 3 Narrative for this application

Correctional Facility Management System (CFMS)		
Authors: Naresh, Vikram Singh, Jagadeshwar		Date:
		Version: 1.0
Use-Case Name:	Adds Offender Case or incident Details	Use-Case Type Business Requirements:
Use-Case ID:	CFMS-4.00	
Priority:	High	
Source:	Offender	
Primary Business Actor:	Jailer	
Other Participating Actors:	Offender	
Other Interested Stakeholders:	Jailer	
Description:	This use-case describes the necessary steps to add the Offender case details to the offender management system.	
Precondition:	To add these details, the offender is to be added into the system	
Trigger:	This use case is initiated by the immediate need to register into the system.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Fills the offender’s case details in the Registration Form.	Step 2: Registers successfully if valid details are provided , else rejects the user registration. All the details are added to the database.
Alternate Courses:	Alt-Step 1: Checks the reasons of rejection and comes up clearing them	
Conclusion:	This use case concludes how to add the case details of each offender	
Postcondition:	Offender case data successfully added into the system	
Business Rules:	N/A	
Implementation Constraints and Specifications:	Provider has full view privileges.	
Assumptions:	Credentials are valid.	
Open Issues:	N/A	

Table 18 : Use case 4 Narrative for this application

Correctional Facility Management System (CFMS)		
Authors: Naresh, Vikram Singh, Jagadeshwar		Date:
		Version: 1.0
Use-Case Name:	Court Appointments	Use-Case Type Business Requirements:
Use-Case ID:	CFMS-5.00	
Priority:	High	
Source:	Offender	
Primary Business Actor:	Jailer	
Other Participating Actors:	Offender	
Other Interested Stakeholders:	Jailer	
Description:	This use-case describes the necessary steps to perform during action to schedule the offender’s court visits and regular counselling sessions.	
Precondition:	To add these details, the jailer must get the appointment from courts and counselling in charges briefing the possible dates of schedule	
Trigger:	This use case is initiated by the immediate need to add the offender court and counselling schedules	
Typical Course of Events:	Actor Action	System Response
	Step 1: Adds the offender’s court schedules and counselling schedules	Step 2: The system checks if the intended offender is in registered, If yes, it will store the schedule details into the system.
Alternate Courses:	Alt-Step 1: Checks the reasons of rejection and comes up clearing them	
Conclusion:	This use case concludes how to assign the court schedules for offender	
Postcondition:	Offender court visit and counselling schedules data successfully added into the system	
Business Rules:	N/A	
Implementation Constraints and Specifications:	Provider has full view privileges.	
Assumptions:	Credentials are valid.	
Open Issues:	N/A	

Table 19 : Use case 5 Narrative for this application

Correctional Facility Management System (CFMS)		
Authors: Naresh, Vikram Singh, Jagadeshwar		Date:
		Version: 1.0
Use-Case Name:	Physical Details	Use-Case Type Business Requirements:
Use-Case ID:	CFMS-6.00	
Priority:	High	
Source:	Offender	
Primary Business Actor:	Jailer	
Other Participating Actors:	Offender	
Other Interested Stakeholders:	Jailer	
Description:	This use-case describes the necessary steps to perform during action to add physical details of the offender	
Precondition:	To add these details, the jailer must get the details for the offender	
Trigger:	This use case is initiated by the immediate need to add the physical details of the offender	
Typical Course of Events:	Actor Action	System Response
	Step 1: Adds the physical details of the offender once the offender is assigned to the jailer	Step 2: The system checks if the intended offender is in registered, If yes, it will store the details into the system.
Alternate Courses:	Alt-Step 1: Checks the reasons of rejection and comes up clearing them	
Conclusion:	This use case concludes the addition of physical details of the offender	
Postcondition:	Physical details of the offender are added	
Business Rules:	N/A	
Implementation Constraints and Specifications:	Provider has full view privileges.	
Assumptions:	Credentials are valid.	
Open Issues:	N/A	

Table 20 : Use case 6 Narrative for this application

Correctional Facility Management System (CFMS)		
Authors: Naresh, Vikram Singh, Jagadeshwar		Date:
		Version: 1.0
Use-Case Name:	Medical Health	Use-Case Type Business Requirements:
Use-Case ID:	CFMS-7.00	
Priority:	High	
Source:	Offender	
Primary Business Actor:	Offender	
Other Participating Actors:	Jailer	
Other Interested Stakeholders:	Jailer	
Description:	This use-case describes the necessary steps to perform during action to add the medical health details of the offender.	
Precondition:	To add these details, the jailer must get the details like medical reports from the offender	
Trigger:	This use case is initiated by the immediate need to Update the offender’s medical report details.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Updates the offender’s health report details	Step 2: The system checks if the intended offender is in registered, If yes, it will store the added details into the system.
Alternate Courses:	Alt-Step 1: Checks the reasons of rejection and comes up clearing them	
Conclusion:	This use case concludes the offender’s health history	
Postcondition:	Offender health history gets added into the system	
Business Rules:	N/A	
Implementation Constraints and Specifications:	Provider has full view privileges.	
Assumptions:	Credentials are valid.	
Open Issues:	N/A	

Table 21: Use case 7 Narrative for this application

Correctional Facility Management System (CFMS)		
Authors: Naresh, Vikram Singh, Jagadeshwar		Date:
		Version: 1.0
Use-Case Name:	Physical Details	Use-Case Type Business Requirements:
Use-Case ID:	CFMS-6.00	
Priority:	High	
Source:	Offender	
Primary Business Actor:	Jailer	
Other Participating Actors:	Offender	
Other Interested Stakeholders:	Jailer	
Description:	This use-case describes the necessary steps to perform during action to add physical details of the offender	
Precondition:	To add these details, the jailer must get the details for the offender	
Trigger:	This use case is initiated by the immediate need to add the physical details of the offender	
Typical Course of Events:	Actor Action	System Response
	Step 1: Adds the physical details of the offender once the offender is assigned to the jailer	Step 2: The system checks if the intended offender is in registered, If yes, it will store the details into the system.
Alternate Courses:	Alt-Step 1: Checks the reasons of rejection and comes up clearing them	
Conclusion:	This use case concludes the addition of physical details of the offender	
Postcondition:	Physical details of the offender are added	
Business Rules:	N/A	
Implementation Constraints and Specifications:	Provider has full view privileges.	
Assumptions:	Credentials are valid.	
Open Issues:	N/A	

Table 22 : Use case 8 Narrative for this application

Correctional Facility Management System (CFMS)		
Authors: Naresh, Vikram Singh, Jagadeshwar		Date:
		Version: 1.0
Use-Case Name:	Incident	Use-Case Type Business Requirements:
Use-Case ID:	CFMS-7.00	
Priority:	High	
Source:	Offender	
Primary Business Actor:	Jailer	
Other Participating Actors:	Offender	
Other Interested Stakeholders:	Jailer	
Description:	This use-case describes the necessary steps to perform during action to add the incident details of the crime committed by offender	
Precondition:	To add these details, the jailer must get the details from the offender and the court	
Trigger:	This use case is initiated by the immediate need to add the case or incident details of the offender	
Typical Course of Events:	Actor Action	System Response
	Step 1: Adds the incident details of the offender once the offender is assigned to the jailer	Step 2: The system checks if the intended offender is in registered, If yes, it will store the details into the system.
Alternate Courses:	Alt-Step 1: Checks the reasons of rejection and comes up clearing them	
Conclusion:	This use case concludes the addition of incident details of the offender	
Postcondition:	Physical details of the offender are added	
Business Rules:	N/A	
Implementation Constraints and Specifications:	Provider has full view privileges.	
Assumptions:	Credentials are valid.	
Open Issues:	N/A	

Table 23 : Use case 9 Narrative for this application

Correctional Facility Management System (CFMS)		
Authors: Naresh, Vikram Singh, Jagadeshwar		Date:
		Version: 1.0
Use-Case Name:	Release	Use-Case Type Business Requirements:
Use-Case ID:	CFMS-8.00	
Priority:	High	
Source:	Court, Jailer	
Primary Business Actor:	Court	
Other Participating Actors:	Offender	
Other Interested Stakeholders:	Jailer	
Description:	This use-case describes the necessary steps to perform during action to add the release date and details of the respective offender	
Precondition:	To add these details, the jailer must get the details from the court	
Trigger:	This use case is initiated by the immediate need to add the release date and details of the offender	
Typical Course of Events:	Actor Action	System Response
	Step 1: Adds the release details of the offender once the court has confirmed on the case and the offender	Step 2: The system checks if the intended offender is in registered, If yes, it will store the details into the system.
Alternate Courses:	Alt-Step 1: Checks the reasons of rejection and comes up clearing them	
Conclusion:	This use case concludes the addition of release details of the offender	
Postcondition:	Release details of the offender are added	
Business Rules:	N/A	
Implementation Constraints and Specifications:	Provider has full view privileges.	
Assumptions:	Credentials are valid.	
Open Issues:	N/A	

Table 24 : Use case 10 Narrative for this application

Correctional Facility Management System (CFMS)		
Authors: Naresh, Vikram Singh, Jagadeshwar		Date:
		Version: 1.0
Use-Case Name:	Action Plan	Use-Case Type Business Requirements:
Use-Case ID:	CFMS-9.00	
Priority:	High	
Source:	Officials who are handling the case	
Primary Business Actor:	Officials who are handling the case	
Other Participating Actors:	Offender	
Other Interested Stakeholders:	Jailer	
Description:	This use-case describes the necessary steps to get the action plan details from the officials who are allotted to deal with this case.	
Precondition:	To add these details, the jailer must get the details from the hired officials	
Trigger:	This use case is initiated by the immediate need to add the action plan details of the offender’s case	
Typical Course of Events:	Actor Action	System Response
	Step 1: Adds the Action plan details of the offender’s case once the hired official details about it	Step 2: The system checks if the intended offender is in registered, If yes, it will store the details into the system.
Alternate Courses:	Alt-Step 1: Checks the reasons of rejection and comes up clearing them	
Conclusion:	This use case concludes the addition of Action Plan details of the offender	
Postcondition:	Action Plan details of the offender are added	
Business Rules:	N/A	
Implementation Constraints and Specifications:	Provider has full view privileges.	
Assumptions:	Credentials are valid.	
Open Issues:	N/A	

Table 25 : Use case 11 Narrative for this application

Correctional Facility Management System (CFMS)		
Authors: Naresh, Vikram Singh, Jagadeshwar		Date:
		Version: 1.0
Use-Case Name:	Case Plan	Use-Case Type Business Requirements:
Use-Case ID:	CFMS-10.00	
Priority:	High	
Source:	Officials who are handling the case	
Primary Business Actor:	Officials who are handling the case	
Other Participating Actors:	Offender	
Other Interested Stakeholders:	Jailer	
Description:	This use-case describes the necessary steps to get the case plan details from the officials who are allotted to deal with this case.	
Precondition:	To add these details, the jailer must get the details from the hired officials	
Trigger:	This use case is initiated by the immediate need to add the case plan details of the offender’s case	
Typical Course of Events:	Actor Action	System Response
	Step 1: Adds the case plan details of the offender’s case once the hired official details about it	Step 2: The system checks if the intended offender is in registered, If yes, it will store the details into the system.
Alternate Courses:	This use case is initiated by the immediate need to add the case plan details of the offender’s case	
Conclusion:	Actor Action	System Response
Postcondition:	Step 1: Adds the case plan details of the offender’s case once the hired official details about it	Step 2: The system checks if the intended offender is in registered, If yes, it will store the details into the system.
Business Rules:	Alt-Step 1: Checks the reasons of rejection and comes up clearing them	
Implementation Constraints and Specifications:	This use case concludes the addition of case Plan details of the offender	
Assumptions:	Action Plan details of the offender are added	
Open Issues:	N/A	

Table 26 : Use case 12 Narrative for this application

6.2.6. Systems Sequence diagram for use case

1. Sequence-Login

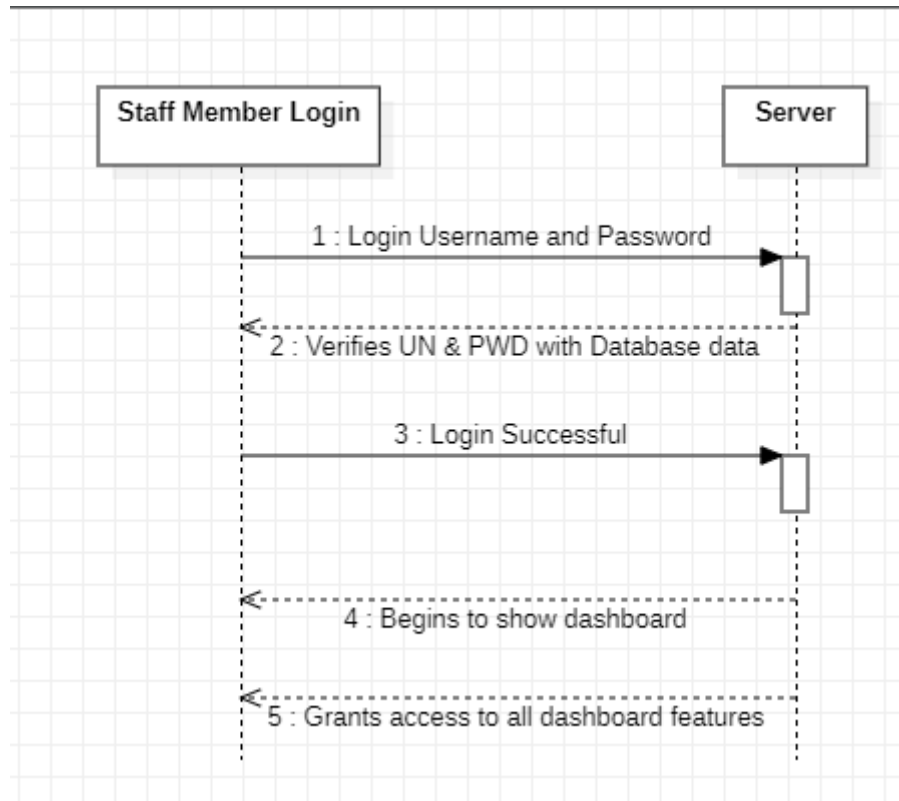


Figure 59 : Systems Sequence diagram for use case

2. Sequence Staff Member Role

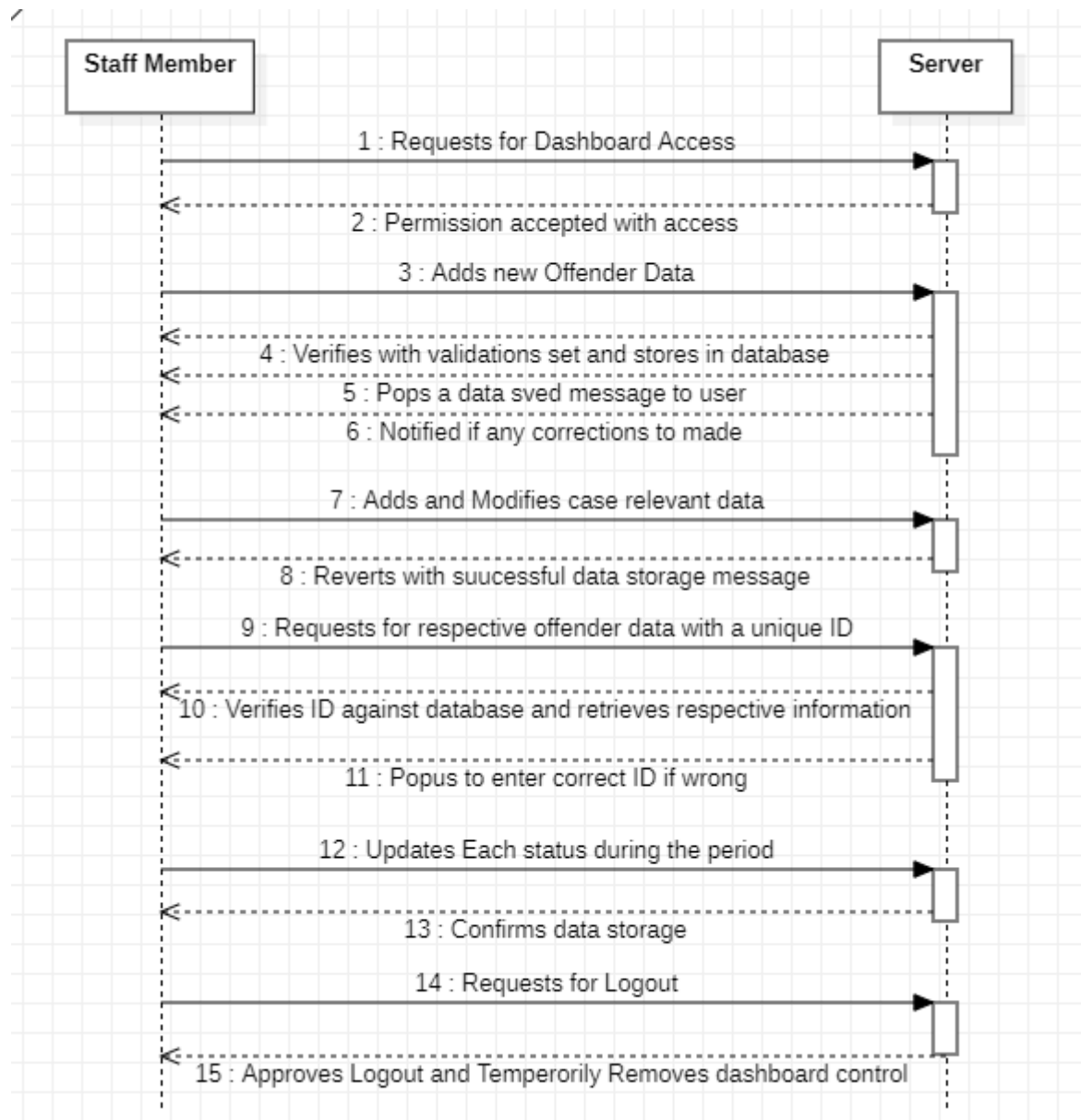


Figure 60 : Sequence Staff Member Role

6.2.7 Activity Diagram

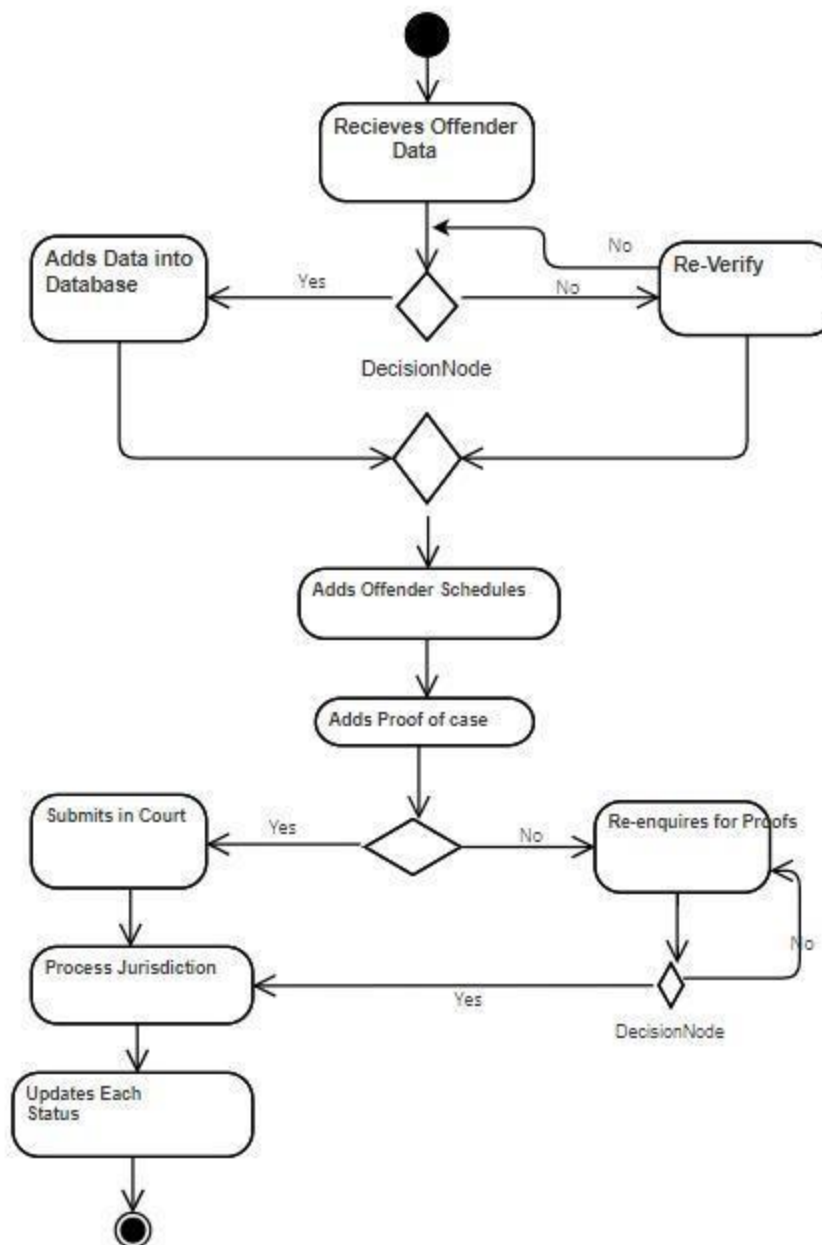


Figure 61 : Activity Diagram

Fig: Activity Diagram

6.3 Design the System Database

6.3.1 Detailed Entity Relationship (ER) Diagram

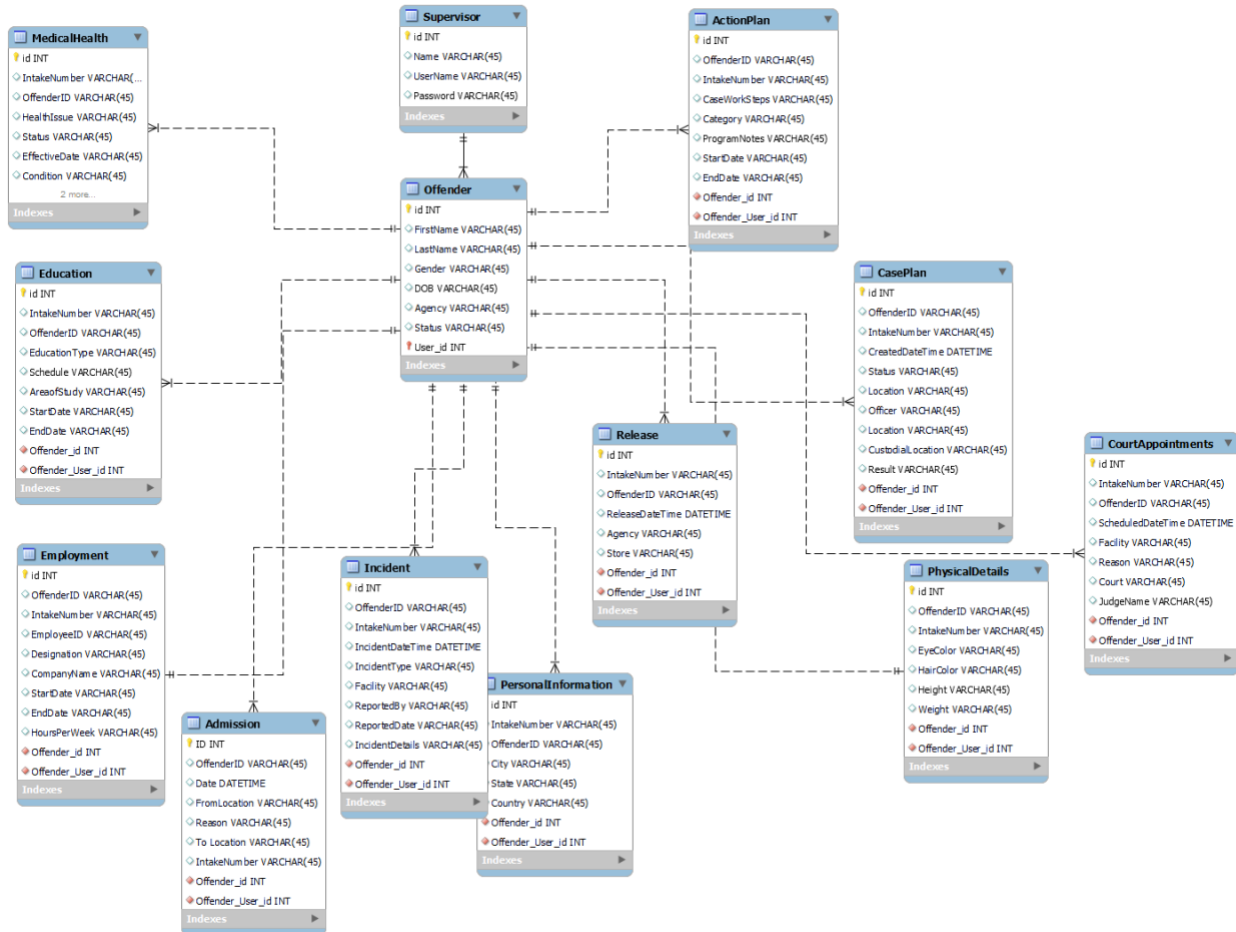


Figure 62 : Detailed Entity Relationship (ER) Diagram.

6.3.2 Perform the 3 normalization forms, listing dependencies

Third Normal Form

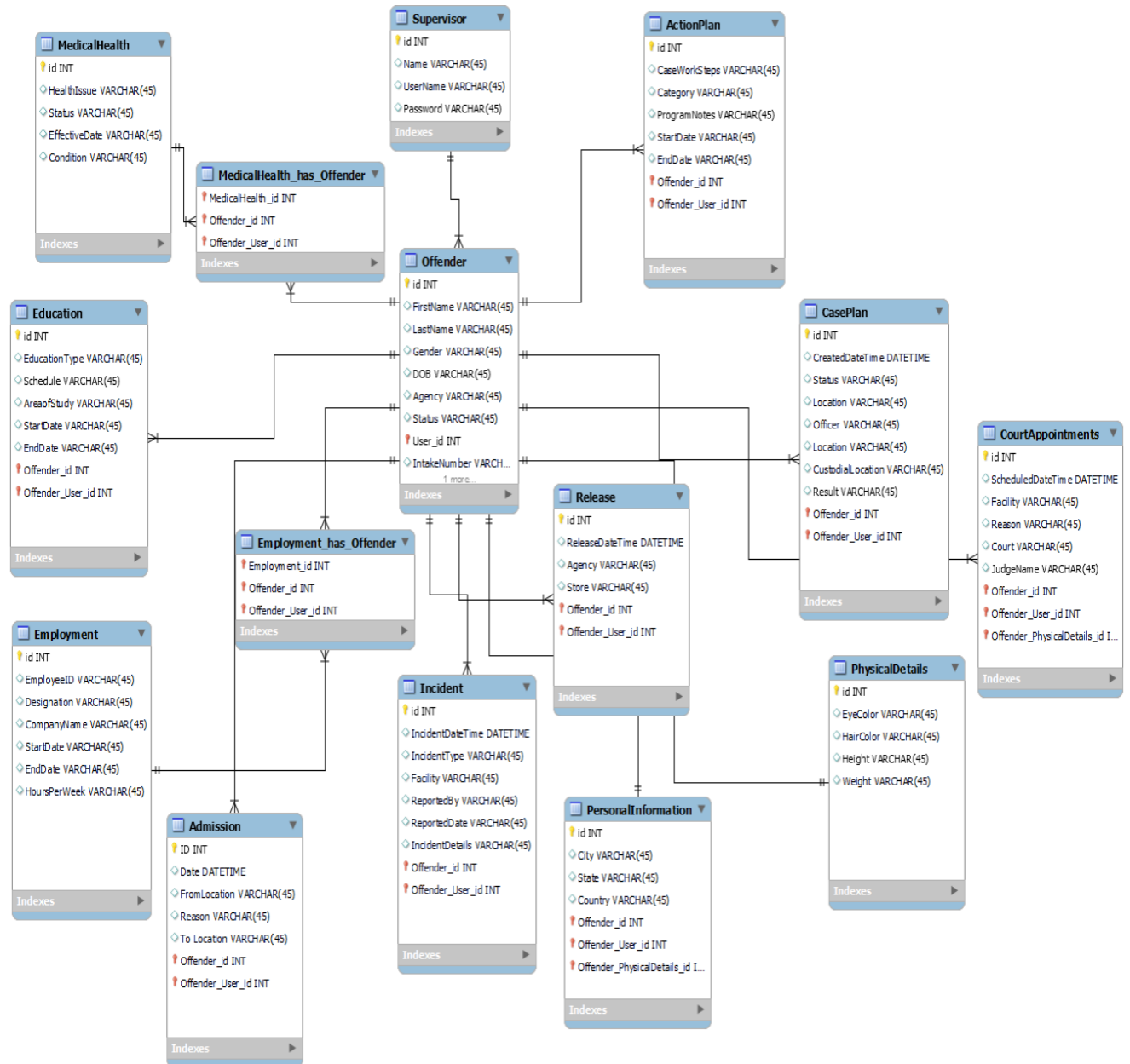


Figure 63 : Perform the 3 normalization forms, listing dependencies.

6.3.3 Chart Entities, Attributes, Domain, Primary keys and Foreign keys

These items are included in the data dictionary.

6.3.4 Data Dictionary of all the attributes

Table Name: Staff Member/Jailer Registration

ID	Field Name	Data Type	Constraint	Description
0	Id	Int	Primary Key	Auto Increment
1	Name	varchar(10)	Primary Key	Auto Increment
2	Email	varchar(50)	Primary Key	Auto Increment
3	Contact	Bigint	Primary Key	Auto Increment
4	Password	varchar(10)	Primary Key	Auto Increment
5	Address	varchar(50)	Primary Key	Auto Increment
6	City/Branch	varchar(50)	Primary Key	Auto Increment

Table 27 : Staff Member/Jailer Registration.

Table Name: Offender Registration

ID	Field Name	Data Type	Constraint	Description
0	Id	Int	Primary Key	Auto Increment
1	Name	varchar(10)	Primary Key	Auto Increment
2	Date of Birth	varchar(10)	Primary Key	Auto Increment
3	Village/City	varchar(10)	Primary Key	Auto Increment
4	Email	varchar(50)	Primary Key	Auto Increment
5	Contact	Bigint	Primary Key	Auto Increment
6	Password	varchar(10)	Primary Key	Auto Increment
7	Address	varchar(50)	Primary Key	Auto Increment
8	City/Branch	varchar(50)	Primary Key	Auto Increment
9	Criminal Activity	varchar(100)	Primary Key	Auto Increment
10	Allegations	varchar(100)	Primary Key	Auto Increment
11	Investigations	varchar(100)	Primary Key	Auto Increment
12	Proceedings	varchar(100)	Primary Key	Auto Increment
13	Counselling	varchar(100)	Primary Key	Auto Increment
14	Other	varchar(100)	Primary Key	Auto Increment

Table 28 : Offender Registration

6.4 System Interface Design

6.4.1 Staff Member Dashboard.


Model Name	Correctional Facility Management System Login
Parameters Passed & Meaning	Email: Use your personal/professional email. Password: Creating a password containing: capital letters, lowercase letters, numbers, and special characters.
Description of Module Function	The page will ask for the administrators username & password and will allow the staff to move on to the next page.
Input	Click: Login/Enter
Output	Access to CFMS. However, if password/username is incorrect, a notification will appear asking the user to re-input.
Called Modules	Dashboard.
Report/Screen Layout	 [LOGIN SCREENSHOT]
Story	After entering username & password to login page. The staff member will be granted access to the following page.
Error Message	If invalid credentials, prompt the user of invalid credentials.

Table 29 : Staff Member Dashboard.

6.4.2 Change Password

Model Name	Correctional Facility Management System – Change Password
Parameters Passed & Meaning	Confirm email address, old password and then enter new password
Description of Module Function	This module allows the user to change their password.
Input	Click: “Change Password” or “Cancel”.
Output	Return to the home page where the user can access the rest of the software.
Called Modules	Change Password
Report/ Screen Layout	[CHANGE PASSWORD SCREEN]
Story	Whenever the user would like to change their current password, they can simply find the change password tab in their student account. The change password will seek for the old password and the ask for an input of the new password. This will then confirm and verify the changes that have been made.
Error Msg.	If any fields are empty, prompt the user of the missing information.

Table 30 : Change Password.

6.4.3 Forgot Password.

Model Name	Correctional Facility Management System –Forgot Password
Parameters Passed & Meaning	Email address given while registering
Description of Module Function	This module allows the user to reset their password
Input	Click: “Send Mail” or “Back to log in”.
Output	Return to the home page where the user can access the rest of the software.
Called Modules	Forgot Password
Report/ Screen Layout	[FORGOT PASSWORD SCREEN]
Story	If the user forgets their account password. They will simply click on “Forgot password” and type in the email associated with the account. They will receive an email prompting the user how to reset their password. Once the password has been reset, the user will log in as normal.
Error Msg.	If any fields are empty, prompt the user of the missing information.

Table 31 : Forgot Password.

6.4.4 Add Offender

Model Name	Correctional Facility Management System- Adding Offender Page
Parameters Passed & Meaning	Full Name, Date of Birth, Village/City, Email, Contact, Password, Address, City/Branch, Criminal Activity, Allegations, Investigations, Proceedings, Counselling's, Other
Description of Module Function	The page will ask for the landlord to input the information of the student they want to add
Input	Click: "Add offender"
Output	Ask user if they would like to add another student or go back to home page
Called Modules	Add Offender
Report/ Screen Layout	<div style="border: 1px solid black; padding: 10px; width: fit-content;"> [OFFENDER ADD SCREEN] </div>
Story	The user will add the student information into the system. The user must then enter the student's personal information and click add to enter the student into the system. This will then verify the student officially.
Error Message	If information is entered incorrectly or cannot be read, a message will prompt "System could not understand the input. Please try again"

Table 32 : Add Offender.

7.0 Construction Phase

In this Correctional Facility Management System, we have started to overcome the problems detected from the problem analysis phase. In the proposed system, every data associated to offender and the case are stored in a cloud database secured with necessary access controls.

The control over the system is handed to a specific person here in called Staff Member. Staff Member will add if any new offender data and updates the case details and offender schedules for case visits, proofs etc. This system is proposed to avoid the case proofs compromising from the system. This system also enables data storage and maintenance at ease and user-friendly manner.

In this system, for every offender a unique ID is generated, and all his related information is stored in it. Once the case details, proofs were submitted, al were uploaded into the system and produced only when they are required by the court.

In the construction phase, we first started with the user interface design for staff member to login and have his dashboard access. Later on we developed the interface for adding offenders and his data. Followed by modifying/updating the offender data whenever required. Finally, we fulfilled the proposed system features.

7.1. Build and Test Networks (intranet and/or internet), if necessary

Testing is the vital phase in any project development and implementation. No system is designed and developed perfect at the first attempt. Each phase of the system is to be tested to review and analyze the improvements or the future demands needed. To fully furnish all these, the developed system is to be tested on different networks and interfaces.

After each module implementation, it is to be functionally tested applying different test cases to check the possible success results and failure results. Based on these outcomes, further development process is to be carried out to make the project a fully furnished one.

7.2. Build and Test Databases

7.2.1. Complete the database Alpha Testing

Databases are the crucial components for storing and managing the system data. Building a user-friendly database structure makes the system easily accessible by any of the user personnel. Each database component is to be tested to find the errors if any.

For the databases, conduct the complete Alpha Testing, also known as acceptance testing. Be spoke systems are developed with respect to the single users. This alpha testing is performed until the system developer and the customer agreed to the developed features are as per the acceptable implementation of the system.

7.3. Install and Test New Software Packages (if necessary)

Correctional Facility management system requires python as a language and Django a framework of python language which also contains its own server. Apart from the backend we'll be requiring a database to store the data.

Steps for Installing Python:

1. Download the executable file from the python.org site. Python version above 3.6 is preferred.
2. Run the installer. Make sure to select both the checkboxes at the bottom, install Launcher, add python to path , and then click Install Now.
3. On clicking the Install Now, the installation process starts.
4. The installation process will take a few minutes to complete and once the installation is successful, the following screen is displayed.

Steps for testing Python Installed or not:

To ensure if Python is successfully installed on your system. Follow the given steps

- Open the command prompt.
- Type 'python' and press enter.
- The version of the python which you have installed will be displayed if the python is successfully installed on your windows.

7.4. Write and Test new Programs. P.688

7.4.1. Complete the Alpha Testing

For the systems built, conduct the complete Alpha Testing, also known as acceptance testing. Be spoke systems are developed with respect to the single users. This alpha testing is performed until the system developer and the customer agreed to the developed features are as per the acceptable implementation of the system.

7.5. Schedule - use the start and end dates of this Academic year

	Month1	Month2	Month3
Requirements Gathering			
Analysis			
Design			
Coding			
Testing			
Implementation			

Table 33 : Schedule - use the start and end dates of this Academic year.

7.6. Staffing - show owners responsibility

7.6.1. Staffing hierarchy chart

1. **Naresh Rayapati:** Naresh Rayapati has taken the responsibility of field surveying regarding the blood banks that are available nearby and collected the information about how they work from receiving the blood from donor and delivering it to the required person. Apart from Field Survey he also learnt the UI technologies that are required for the project.
2. **Vikram Singh Navsinde:** Vikram Singh Navsinde took the responsibility of clearly documenting the data that was gathered by the team members from the field survey.
3. **Jagadeshwar Alladi:** Jagadeshwar Alladi has taken the responsibility of gathering the fields that are required to store the Offender required data into the database and took the part in developing the Database backend.

7.7. Acceptance testing

7.7.1. Plan

Software project plan can be viewed as the following.

Within the organization: Within the organizational level, assess the following before you start the project implementation.

1. How the project is to be implemented?
2. What will be the constraints?
3. What are the different market trends and what should be our strategies?

With respect to the Customer: Every product we develop is intended to address one or more problems or insufficiencies of the customers or the end users. So ensure that the project is going to address all its intended usage. If possible, schedule timely meetings with the customers and examine to take the feedback. This helps in immediate modifications than roll backing all the changes for a single modification/suggestion.

7.7.2. Schedule

	Month1	Month2	Month3
Requirements Gathering			
Analysis			
Design			
Coding			
Testing			
Implementation			

Table 34 : Schedule.

7.7.3. Acceptance Criteria

7.7.3.1. Complete the Beta Testing

Beta testing must be conducted if the system is to be launched as a complete software product. During this process of beta system, the product is delivered among multiple potential users who agree to use the system. The customers then start using the system and reports if they face any difficulty in using or if any improvisations needed. This helps in building a complete system acceptable anywhere.

7.8. Tools to be used in the proposed system

7.8.1. Data Base

A database management system (or DBMS) is essentially nothing more than **a computerized data-keeping system**. Users of the system are given facilities to perform several kinds of operations on such a system for either manipulation of the data in the database or the management of the database structure itself.

To store the data of the Offender management system that involves donors blood data , and the requests raised by the users for blood we will be making use of the MYSQL database.

7.8.2. Screen generators

There are no screen generators used in this system.

7.8.3. Hardware - servers, etc.

The hardware that is required for executing of project involves the following:

1. Laptop / Desktop
 - a. 4GB DDR3 / DDR4 Ram
 - b. i5 Processor of any generation
 - c. Minimum 128GB of Hard Disk space

The servers that we are going to use are by default provided by Django, a web framework of python. So There is no requirement of specially installing any servers like nginx, Apache etc.

7.9. Maintenance documentation

7.9.1. List of modules

Donor Module

- The donor Module covers all the activities of the donor with respect to the system.
- The donor can register into the system and log in to the dashboard.
- The donor will post the blood donation request by adding the blood sample name and units available.
- If the blood added is in the specific time duration given, the blood banks will accept the donation request. Otherwise, the donation request gets rejected.

Receiver Module

The receiver registers into the system and logs in for the dashboard.

In the dashboard, the receiver can see the different blood banks and the available blood samples in it. If he found the required blood sample in the right amount, the receiver puts the request to blood banks. The blood banks check the receiver request, verifies and accepts the request. If the Offender denies the request, the notification will be sent to the receiver dashboard. So, he can resend the request for blood.

Blood Bank

Offender manager is the admin who controls all the system. He is the one who verifies the blood received from the donors and posts it for donation.

7.9.2. Auditing procedures

Maintain a full-length document to record the requirements received and the requirements that are fulfilled throughout the project. Note down the progress in all areas of the project with respect to the feedback received and the further implementations made. This helps in tracking the day-to-day complete project flow.

7.9.3. Backup and restore procedures

Maintaining Backups of the database is really an important case for preventing the data loss that occurs due to some technical glitches or even disasters. So in order to avoid these losses we need to maintain the backups of the database at particular intervals of time.

These backups are so helpful in restoring the data to the database.

MySQL provides backup and restore functionality, in GUI based also, we can simply backup our entire database into a single file and restore it whenever and wherever it is required although version should be matched with the restoring MySQL server.

7.9.4. List of maintenance contacts with email and phone numbers

Name	Email	Phone	Maintenance
Naresh Rayapati	rayapatin@mail.sacredheart.edu	475-332-5803	
Vikram Singh Navsinde	navsindev@mail.sacredheart.edu	203-220-1519	
Jagadeshwar Alladi	Alladij2@mail.sacredheart.edu	203-410-3051	

Table 35 : List of maintenance contacts with email and phone numbers.

7.9.5. List of vendors, consultants and external contact`

Name	Email	Phone
Naresh Rayapati	rayapatin@mail.sacredheart.edu	475-332-5803
Vikram Singh Navsinde	navsindev@mail.sacredheart.edu	203-220-1519
Jagadeshwar Alladi	Alladij2@mail.sacredheart.edu	203-410-3051

Table 36 : List of vendors, consultants and external contact.

8. Implementation Phase

8.1. Conduct System test

8.1.1. Schedule

System testing is conducted to check the functionality of each of the modules involved in the entire software. This is to find the functional errors and bugs to early rectify them and resolve them. Early detection of bugs is highly recommended to complete the project on scheduled time to complete it in the budget allocated.

8.1.2. Acceptance Criteria

Overtime bugs and errors lead to revamping few of the modules. This requires the manpower, time and the budget. So, to avoid these system tests are scheduled at regular intervals. Whenever the test case meets the expected outcome, then we can consider it as an Acceptance.

8.2. Prepare Conversion Plan

A conversion plan is something designed to provide a detailed schedule of designed modules and their involving activities. This simply states the beginning and end dates of the task.

8.3. Install Databases

8.3.1. Populate the new system's databases

MySQL is one of the most relational databases used across almost all major projects. Checkout these prerequisites and start installing the MySQL server.

Prerequisites

- MySQL Setup Software
- Microsoft .NET Framework 4.5.2
- Microsoft Visual C++ Redistributable for Visual Studio 2019
- RAM 4 GB (6 GB recommended)

Download SQL

1. Go to my SQL official website <https://dev.mysql.com/downloads/installer/> and download the community server edition software.
2. Download the setup choosing MySQL-installer-web-community.

Installing MySQL

1. After setup got downloaded, unzip the MSI installer .exe file. Next choose Setup type depending on your MySQL product and features. Select the full option shown and click Next.

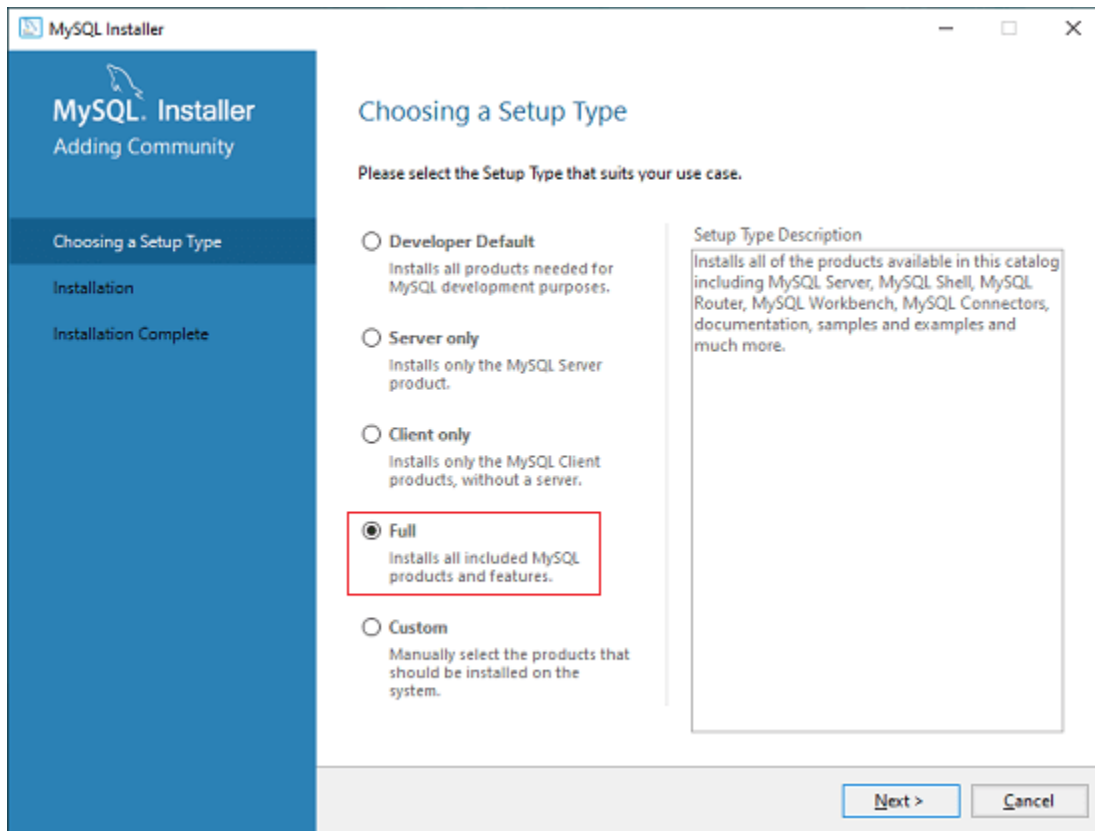


Figure 64: Step 1 Installing MySQL.

2. It may give some features which may fail to install on system. Resolve them clicking execute. Then click on next.

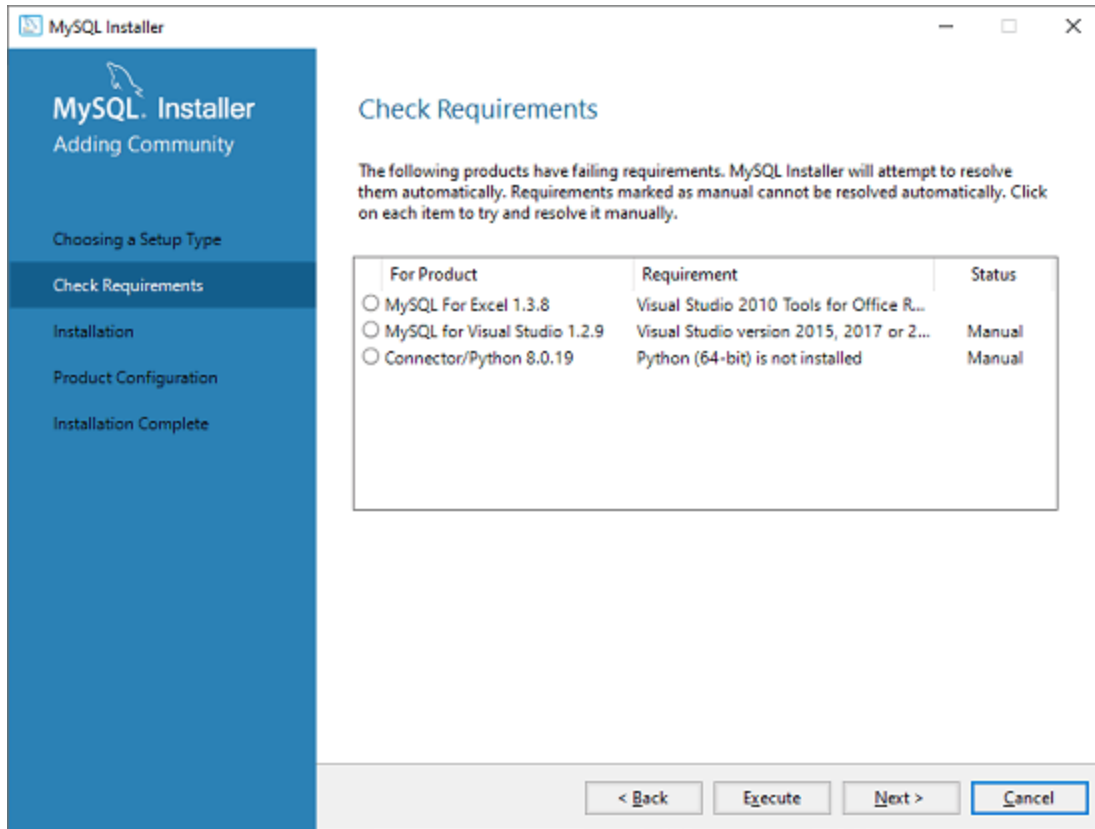


Figure 65 : Step 2 Installing MySQL.

3. Then you will see a dialogue box asking to confirm. Click on Yes and your product gets installed in a few minutes. You will now see the list of products that are going to be installed.

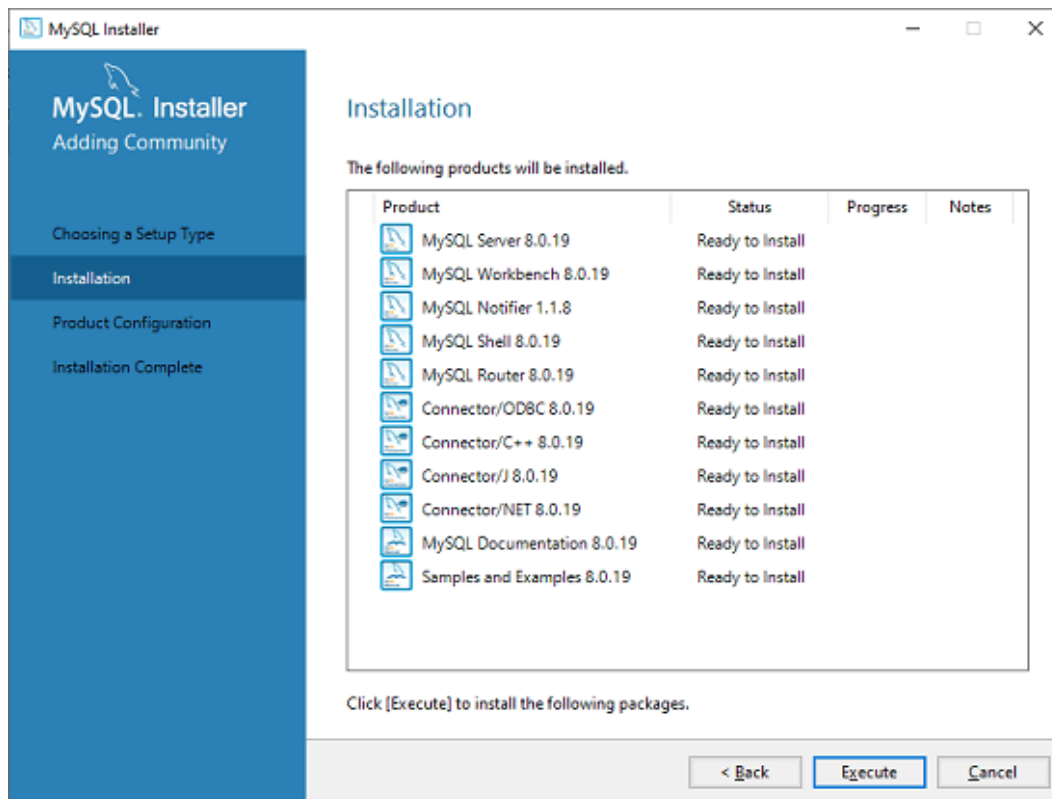


Figure 66: Step 3 Installing MySQL.

4. Click on Execute and it will install all the products. After installation, click on next.
5. Set the MySQL root password.

In this way, follow all the steps your screen suggests.

8.3.2. Hardware – servers, etc.

The hardware that is required for executing of project involves the following:

2. Laptop / Desktop
 - a. 4GB DDR3 / DDR4 Ram
 - b. i5 Processor of any generation
 - c. Minimum 128GB of Hard Disk space

The servers that we are going to use are by default provided by Django , a web framework of python. So There is no requirement of specially installing any servers like nginx, Apache etc.!

8.4. Training

8.4.1. Plan how to Schedule Training manuals, User guides

Once the system is fully developed, a complete training will be provided for all the intended Offender admins, general users. In this training process, we ensure to educate the benefits of using the system. The system is developed on user-friendly terms and hence is easy to use by anyone with minimum of knowledge.

8.5 Convert to a new system.

8.5.1. Stakeholders discuss experiences

Stakeholder experience is a key element of firms' success mantra. The foundation of the service-profit chain theory is the significance of both internal and external stakeholders, particularly customers and employees, and the relationship between these two important stakeholders and profitability and growth. The relationship between employee satisfaction and overall income and growth is quite obvious:

In addition to providing exceptional service and improving client experiences, motivated, devoted, and productive staff make the finest brand ambassadors. Thus, the basic tenet of the service profit chain notion is that satisfied employees equal satisfied clients. Therefore, utilising the connection between the two stimulates and accelerates revenue and growth.

8.5.2. Agree on future enhancements

The stakeholders agreed on the future enhancements proposed.

8.5 3. What did we learned – Openly documented

Through this project, we understood how a project is proposed and requirements are formulated. We have completely involved in this project right from gathering the requirement to designing the project plan.

We designed the project plan considering the proposed budget for each of the modules involved. As we closely worked with the finance team of the project, we could now easily estimate the cost of a project and analyse the investment required for the coming years also. We analyzed the profits expecting after keeping the revenue with respect to the project budget spent.

9.0 Glossary

Offender: The person who is accused for committing or involving in any crime.

Jailer: The in charge or the authorized person who takes care of all the offenders in the jail.

Case: The crime or the offend charged on the offender is called as the case.

Case Proofs: The details like photographs or the persons who witnessed the crime are termed as case proofs.

Counselling: For some cases, it is just required the offender to be counselled on the positive note of life and helping them to come from the trauma of committing unwanted mistakes. This process of assessing the offender by the psychologists is called the counselling sessions.

10. Bibliography

1. Trotter, C. and Evans, P. (2012) 'An Analysis of Supervision Skills in Youth Probation.'
2. Australian and New Zealand Journal of Criminology 45(2): 255-273
3. Ward, T. and Brown, M. (2004) 'The Good Lives Model and Conceptual Issues in Offender
4. Rehabilitation', Psychology, Crime and Law 10 (3): 243-257
5. Partridge, S. (2004) Examining Case Management Models for Community Sentences, Home
6. Office Online Report 17/04, London: Home Office.