1. Introduction

1.1 Existing System

The traditional, non-digital system for managing legal cases has several significant drawbacks:

- **Physical Document Risks:** There is a constant risk of fire, theft, or loss of physical documents.
- **Inefficient Retrieval:** Finding old case files is a time-consuming process.
- **Slow Workflow:** Heavy workloads often lead to a slower overall process.
- **Manual Information Gathering:** Searching for and collecting information physically is a cumbersome task.
- **Poor Data Security:** The system offers less data security compared to digital alternatives.
- **Accessibility Issues:** Accessing necessary information from different locations is a problem.
- **Record-Keeping Challenges:** Maintaining comprehensive records for every file is challenging.
- **Risk of Missing Deadlines:** There is a high level of stress associated with potentially missing important hearing dates, reports, and evidence deadlines.

1.2 Need for the New System

The new system is needed to address the inefficiencies of the existing manual processes. Its purpose is to:

- Save time in legal case management.
- Accelerate legal processes.
- Eliminate extensive paperwork.
- Remove the need for physical storage of documents.
- Provide timely notifications for hearing dates and case updates.
- Manage cases more smoothly and achieve faster results.
- Enable faster searching and access to information from anywhere at any time.

1.3 Objective of the New System

The primary objectives for developing the eCourt Manager system are to:

- **Reduce Manual Work:** Minimize the manual effort required in case management.
- **Increase Efficiency:** Reduce the total time required for various tasks.
- **Optimize Manpower:** Lower the amount of manpower needed to manage cases.
- **Ensure Accuracy:** Provide a higher degree of accuracy in data and record-keeping.
- **Centralize Data:** Create a proper, centralized database for all case-related information.

- **Generate Reports:** Enable the easy generation of reports.
- **Be User-Friendly:** Offer a user-friendly interface that is easy to access and navigate.
- **Maintain Reliability:** Ensure the system is reliable, accurate, and provides non-redundant information.

1.4 Problem Definition

Lawyers often face significant challenges in managing a large volume of cases, clients, and related documentation through traditional, paper-based methods. This manual system is inefficient, prone to errors, and poses security risks such as the loss or theft of critical documents. Key problems include the time-consuming nature of searching for old files, the difficulty in accessing case information remotely, and the high risk of missing important dates for hearings and evidence submission. This overall inefficiency creates a stressful and slow-moving workflow, making it difficult to manage a legal practice effectively. The eCourt Manager project aims to solve these problems by creating a centralized, secure, and accessible digital platform for complete case management.

1.5 Advantage

The implementation of the eCourt Manager system offers several key advantages:

- **Efficiency:** The online system reduces the inefficiency of slower manual processes.
- **Document Security:** It mitigates the risk of damage to case information and prevents documents from being stolen or lost.
- **Reduced Manual Labor:** The system reduces the amount of manual work and the need for extensive manpower.
- **Accessibility:** Lawyers can easily access information about current or past cases from any location.
- **Task Delegation:** Junior lawyers or staff can be assigned tasks like setting the next court hearing date.
- **Client Access:** Clients can view court hearing dates and other case-related information.
- **Appointment Scheduling:** Clients have the ability to schedule appointments to meet with their lawyer.

2. Requirement Analysis (Literature Review and Survey)

User Requirements

- Lawyers can register, manage client details, manage case documents, and view case notifications from anywhere at any time.
- Clients can create their own accounts, request lawyer changes, view case progress, and provide feedback.
- Staff supports lawyers with case management and timely reminders about hearings.
- Admin manages user registrations, staff allocations, and system-level controls.

Functional Requirements

- Centralized login/authentication ensuring secure access for all users.
- Add, update, and delete cases in the system.
- Store and manage case documents (FIRs, evidence, etc.).
- Manage hearing dates and send notifications about hearings.
- Enable clients to request lawyer changes through the platform.
- Generate various reports for users and admin.
- User and staff management features for the admin role.

Non-Functional Requirements

- **Usability:** The user interface is intuitive, simple, and designed for effective interaction without training.
- **Reliability:** The system must have low failure rates and resolve errors quickly. It should prevent unauthorized access.
- **Performance:** Multiple users can simultaneously use the application (lawyers, clients, staff) without system lag.
- **Security:** Login modules prevent unauthorized access; strong password requirements for all users.
- **Accuracy**: Information is correctly processed and displayed, with minimal errors.
- **Supportability**: The system must be compatible with all major operating systems and browsers.

Stakeholder Survey :

- **Lawyers:** Need an efficient online platform to manage many clients, cases, documents, and remotely access all critical details. Expressed difficulty with manual paperwork and tracking hearing schedules.
- **Clients:** Want easy ways to view case status, request lawyer changes if dissatisfied, and provide feedback or ratings. Prefer a transparent, user-friendly interface with notifications about case progression.
- **Staff:** Require access to client and case data to coordinate supporting activities like document uploads and hearing reminders. Need tools to assist lawyers and keep all case records organized.
- **Admin:** Responsible for overseeing the system, managing user registrations, allocating staff, handling security, and generating reports. Require robust controls for maintaining data integrity and system-wide configuration.

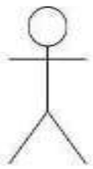
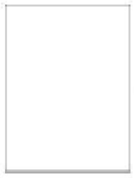
Key Survey Findings:

- **Centralized Management:** All parties want a single, streamlined system for managing cases, hearing dates, and notifications instead of fragmented or paper-based workflows.
- **Accessibility:** Stakeholders emphasize access from any location at any time and compatibility with multiple platforms (desktop, mobile).
- **Automation:** Notifications, reminders for hearings, and streamlined lawyer/client communication are highly desired.
- **Data Security:** Secure logins and the ability to control and audit system access are needed to protect sensitive information.
- **Reporting:** Quick report generation for case statuses, hearing outcomes, and user feedback is essential for admins and lawyers.

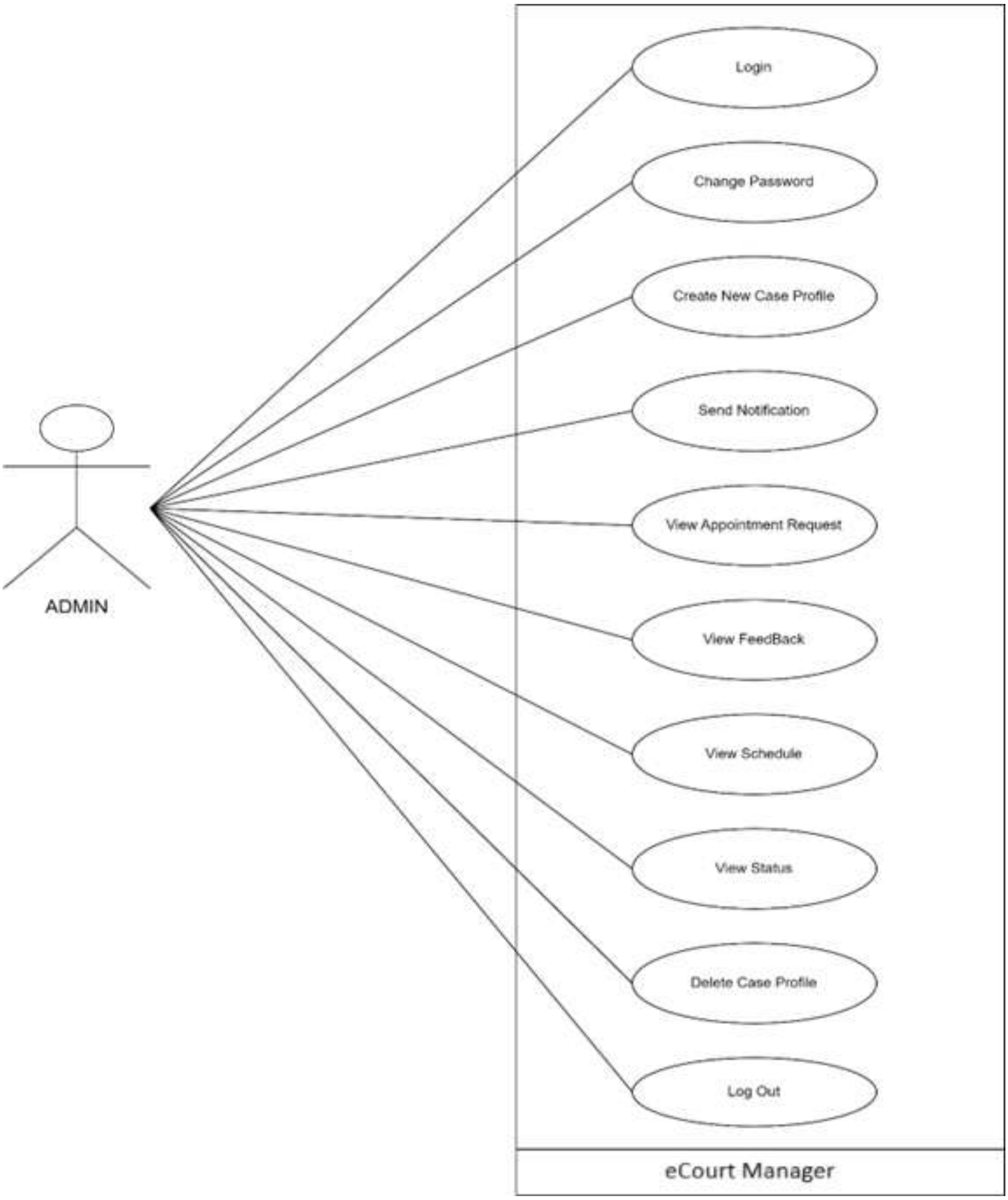
3. System Design

3.1 Use Case Diagram

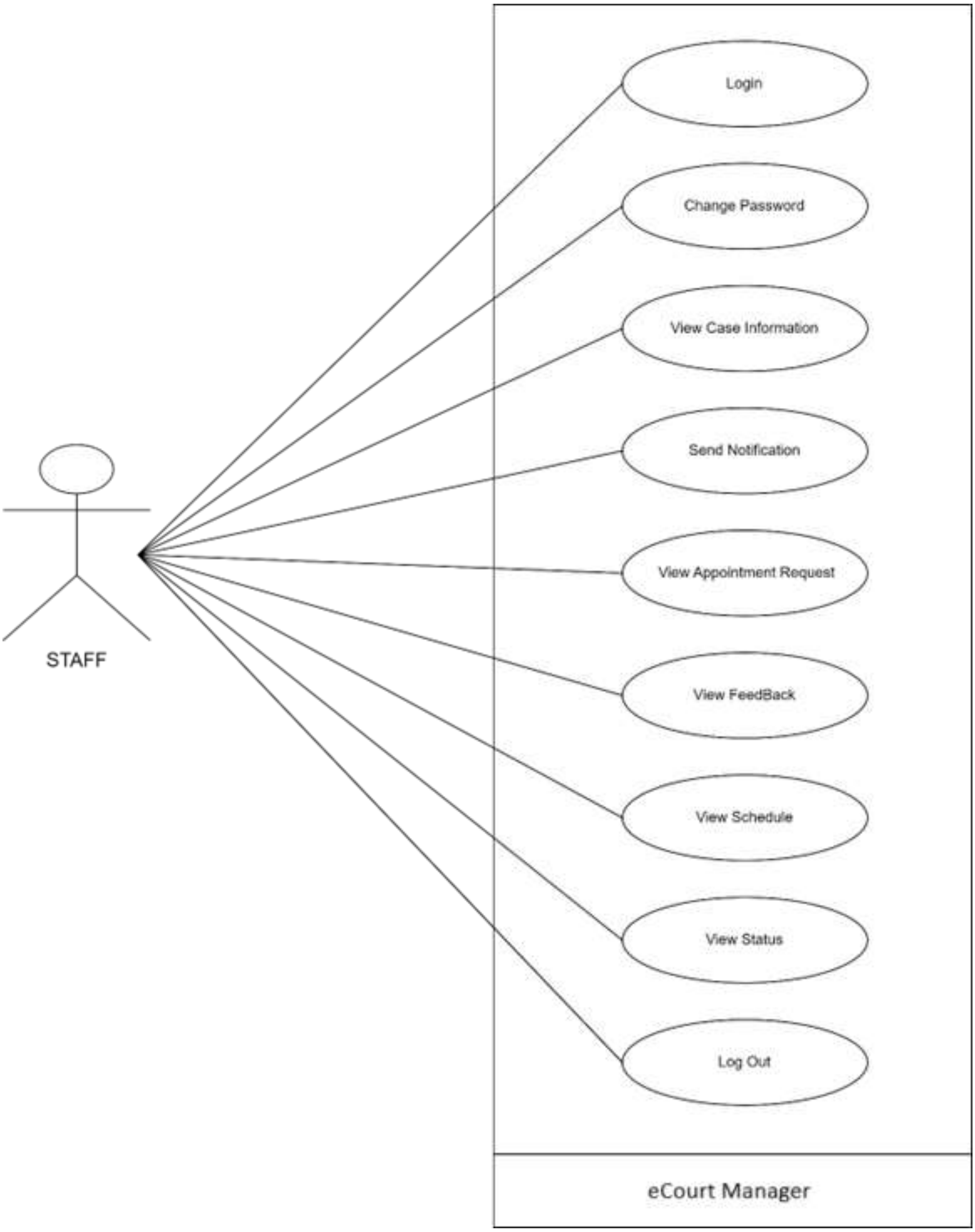
A Use Case Diagram displays the relationship between users (actors) and the various functions (use cases) they can perform within a system. It provides an external view of the system, representing actions a user might take to complete a task.

Symbol	Name	Description
	Actor	Actor in a use case diagram is any entity that performs a role in one givensystem. This could be a person, organization or an external system andusually drawn like skeleton shown below.
	Use case	A use case represents a function or anaction within the system. It’s drawn asan oval and named with the function.
	System Boundary	The system is used to define the scopeof the use case and drawn as a rectangle. This an optional element butuseful when you’re visualizing large systems. For example, you can create all the use cases and then use the system object to define the scope covered by your project. Or you can even use it to show the different areascovered in different releases.
———	Association	Association represents communication between actor anduse case.

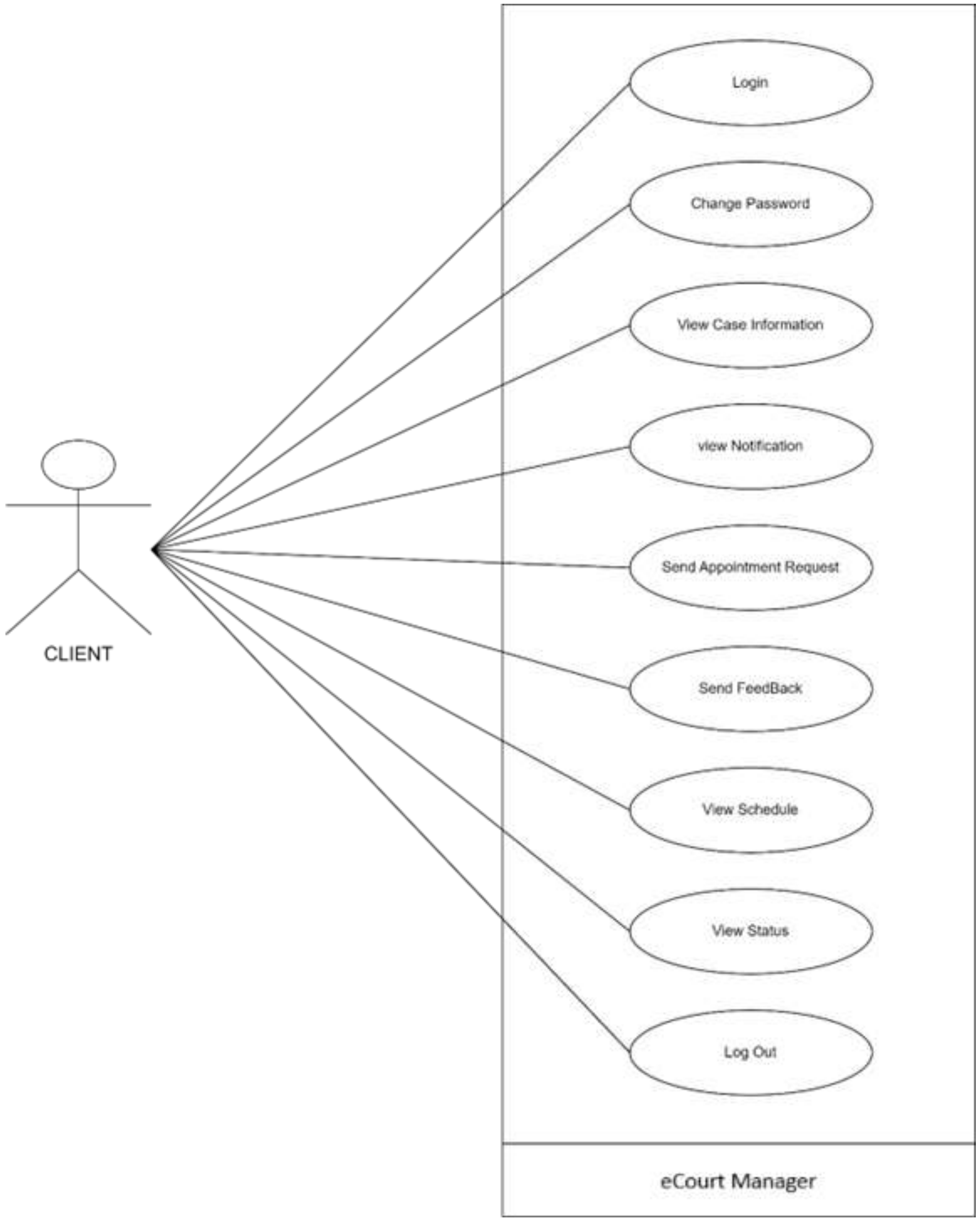
Admin:


Login
Change Password
Create New Case Profile
Send Notification
View Appointment Request
View FeedBack
View Schedule
View Status
Delete Case Profile
Log Out
ADMIN
eCourt Manager

Staff:


Login
Change Password
View Case Information
Send Notification
View Appointment Request
View FeedBack
View Schedule
View Status
Log Out
STAFF
eCourt Manager

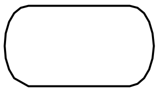
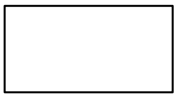
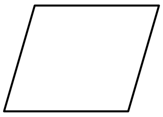
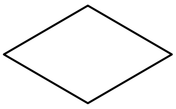
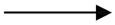
Client:


Login
Change Password
View Case Information
view Notification
Send Appointment Request
Send FeedBack
View Schedule
View Status
Log Out
CLIENT
eCourt Manager

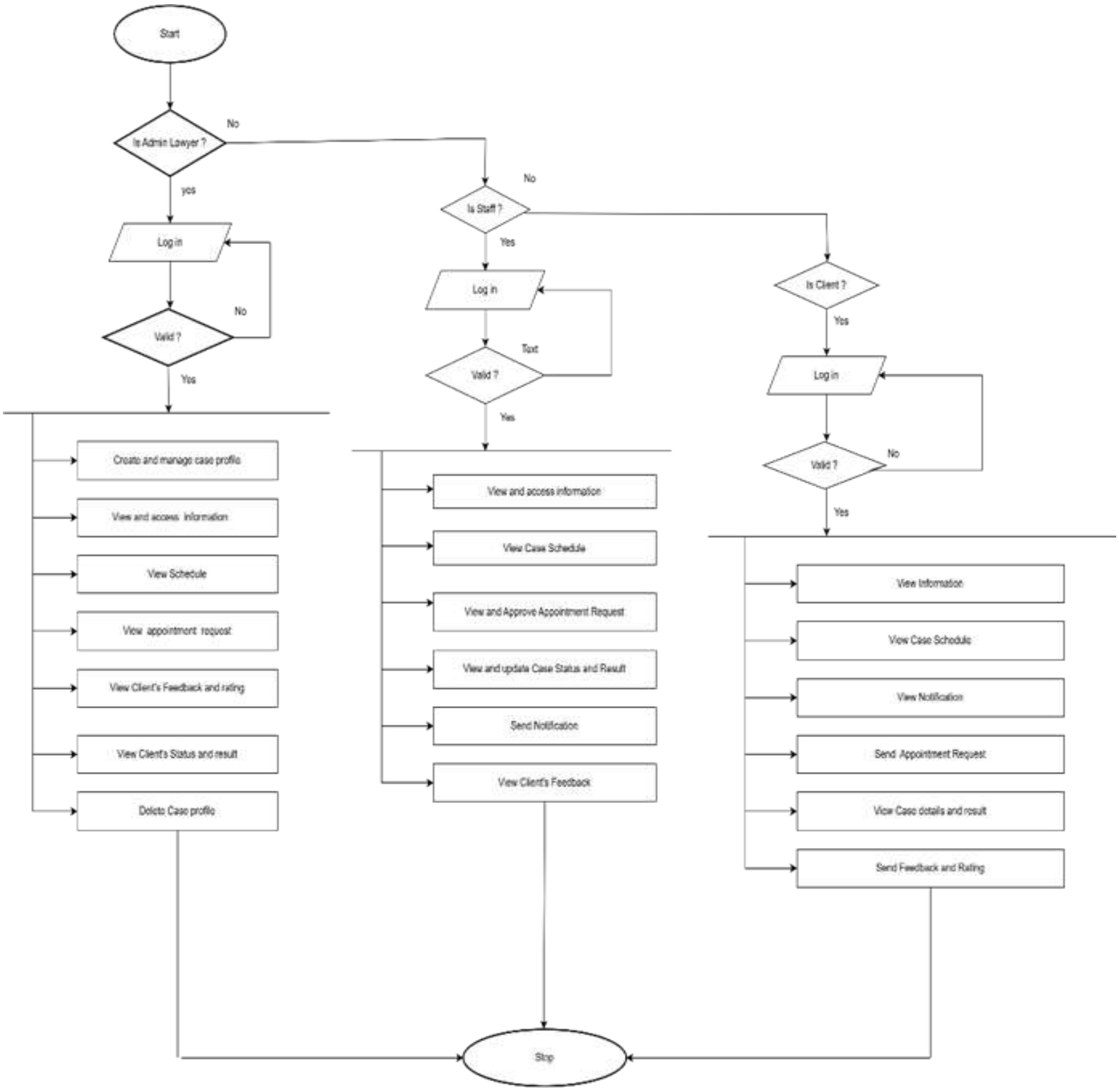
3.2 Flow Diagram

A System Flow Diagram, or flowchart, visually represents how data moves through a system and how decisions are made to control events. It is an effective way to visualize a complex process for a team.

SYMBOL OF SYSTEM FLOW DIAGRAM :-

Symbol	Name	Description
	Terminator	The terminal or terminator represents the start or end points of a flowchart process.
	Process	The process symbol is the most common component of a flowchart and indicates a step in the process.
	Input/ Output	The input/output symbol represents the process of in- or outputting external data.
	Decision	This symbol represents a decision you or your team need to make to get to the next step of the process. Typically, it's a true or false decision or a yes or no question that you need to answer.
	Flowline	The flowline shows the process's direction by connecting two blocks with one another.

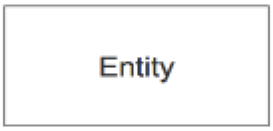
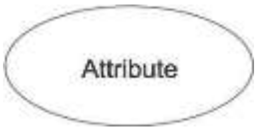
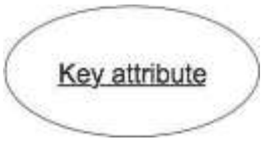
Flow Diagram:

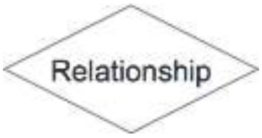
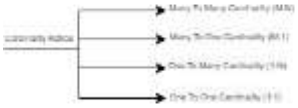


3.3 E-R Diagram

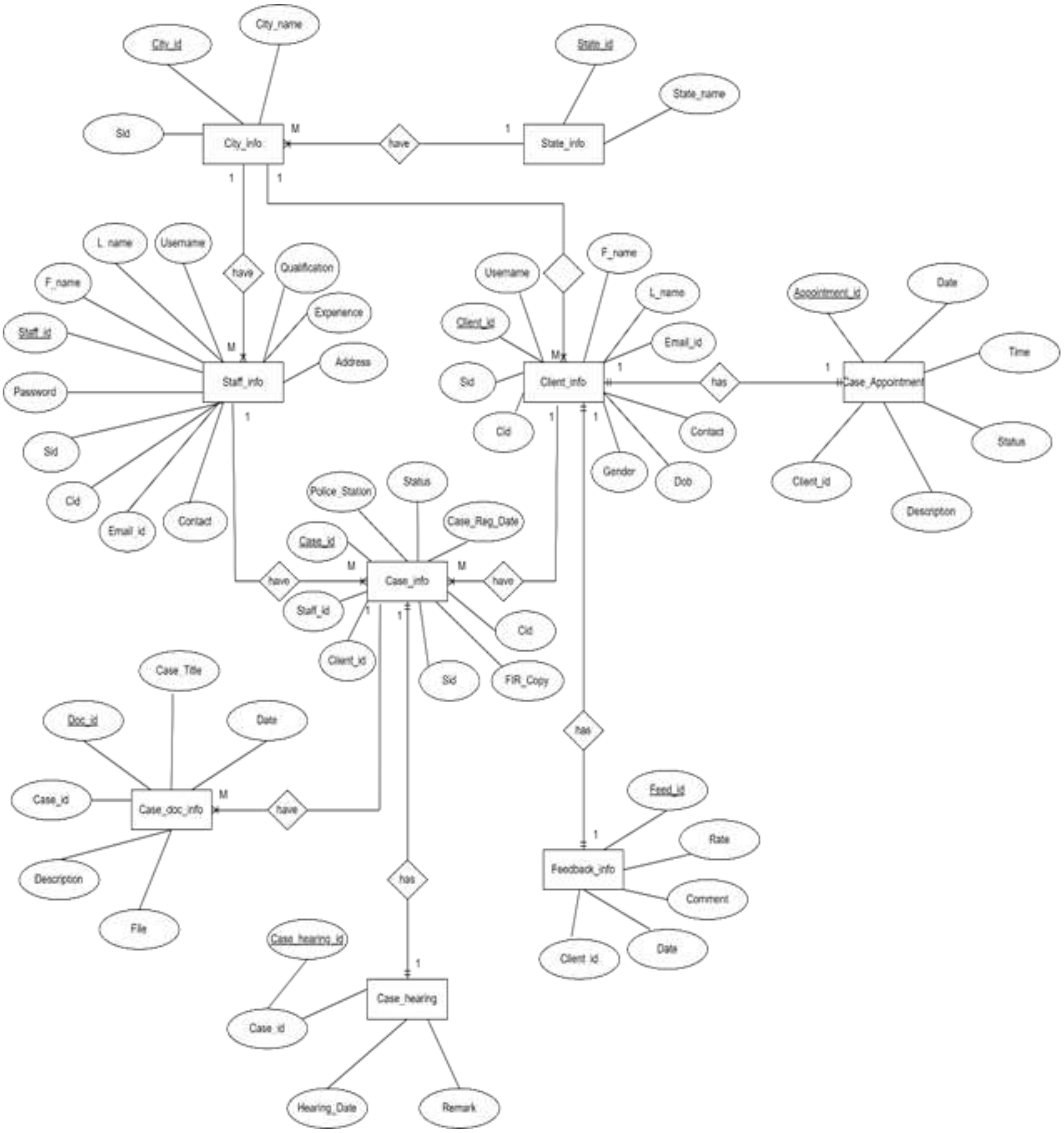
An Entity-Relationship (E-R) Diagram is a type of flowchart used in database design that illustrates how entities like people, objects, or concepts relate to each other within a system.

SYMBOL OF ENTITY ER DIAGRAM:-

Symbol	Name	Description
	Entities	Represents real-world objects or concepts in the database. ExamplesSupplier, Manufacturer, Distributor,Retailer, and Customer.
	Attributes	Attributes provide additional information about entities. Examples product attributes (e.g., name, price), customer attributes (e.g., name, address), etc.
	Primary Key	Denotes the attribute within an entitythat uniquely identify instances of theentity. Typically underlines an attribute to indicate its importance.

	Strong Relationships	A relationship where entity is existence-independent of other entities, and PK of Child doesn’t contain PK component of ParentEntity. A strong relationship is represented by a single rhombus Manufacturing.
 Cardinality Ratios Many To Many Cardinality (M:N) Many To One Cardinality (M:1) One To Many Cardinality (1:N) One To One Cardinality (1:1)	Cardinality	Specifies the number of instances orrelationships between entities. Common notations include "1" (one), "M" (many), "0" (zero), and "N" (avariable number).

ER Diagram:


City_name
City_id
State_id
State_name
Sid
City_info
M
have
1
State_info
1
1
L_name
Username
have
Qualification
F_name
Experience
Staff_id
M
Staff_info
Address
Password
1
Sid
Cid
Email_id
Contact
F_name
Username
L_name
Client_id
Email_id
Sid
M
Client_info
1
has
1
Case_Appointment
Appointment_id
Date
Time
Status
Client_id
Description
Cid
1
1
Contact
Gender
Dob
Police_Station
Status
Case_Reg_Date
Case_id
M
have
Case_info
M
have
Staff_id
1
1
Cid
Client_id
Sid
FIR_Copy
Case_Title
Doc_id
Date
Case_id
Case_doc_info
M
have
Description
File
has
Feed_id
1
Rate
Feedback_info
Comment
Client_id
Date
has
Case_hearing_id
1
Case_id
Case_hearing
Hearing_Date
Remark

3.4 Class Diagram

A Class Diagram maps out the static structure of a system by modelling its classes, attributes, operations, and the relationships between objects. Since classes are the building blocks of objects, these diagrams are fundamental components of UML.

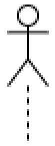
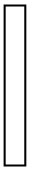
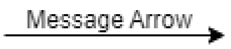
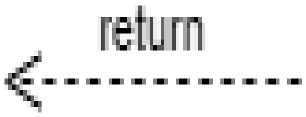
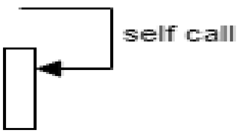
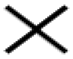
3.5 State Diagram

A State Diagram is a visual model that shows the different conditions, or **states**, an object can be in throughout its lifecycle. It illustrates how the object **transitions** from one state to another in response to specific events or triggers. Essentially, it provides a clear map of an object's dynamic behaviour from start to finish, making it ideal for designing event-driven systems.

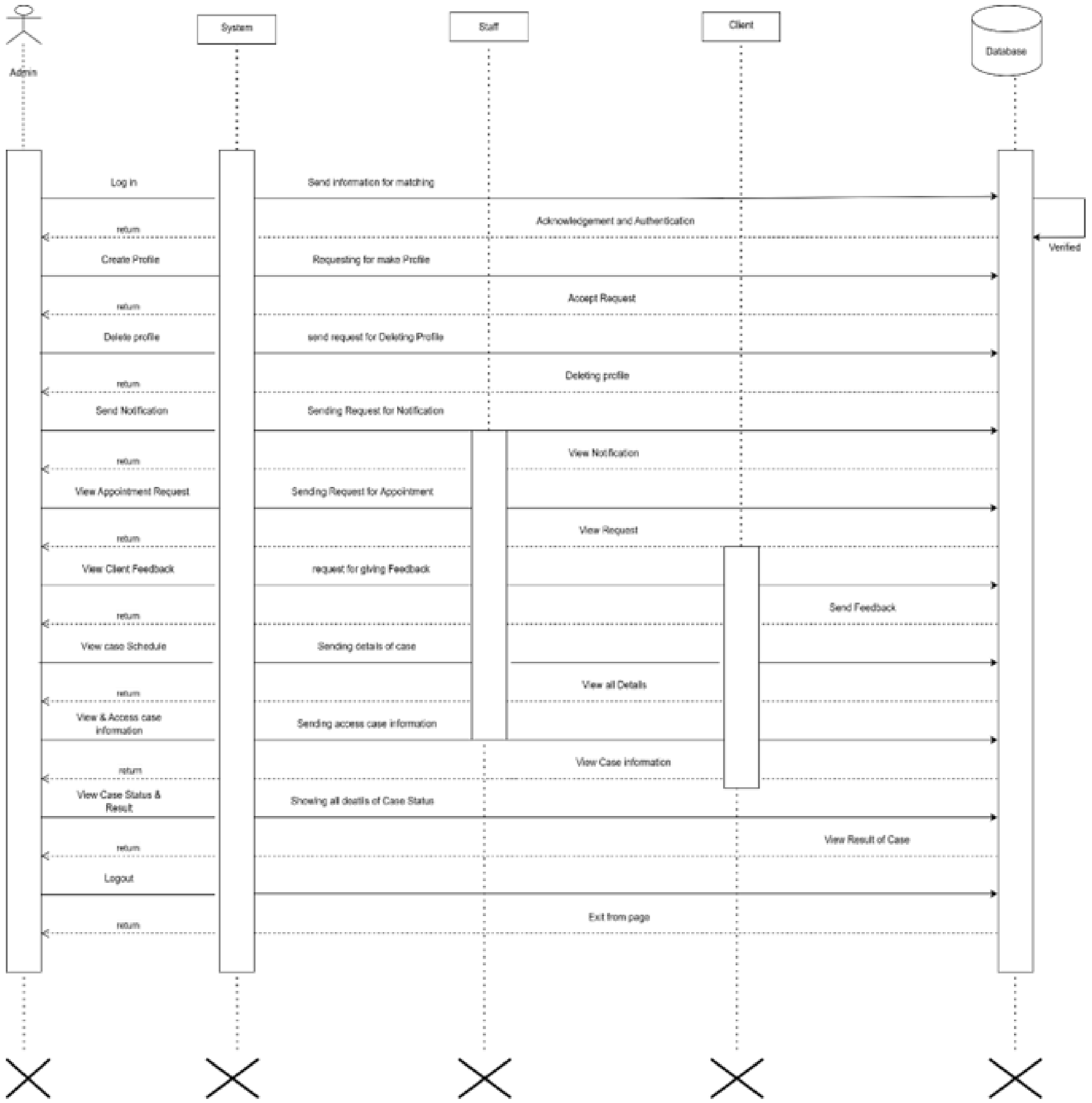
3.6 Sequence Diagram

A Sequence Diagram represents the details of a UML use case by illustrating how objects interact with each other in a sequential order over time. These diagrams can be useful references for businesses to understand interaction flows.

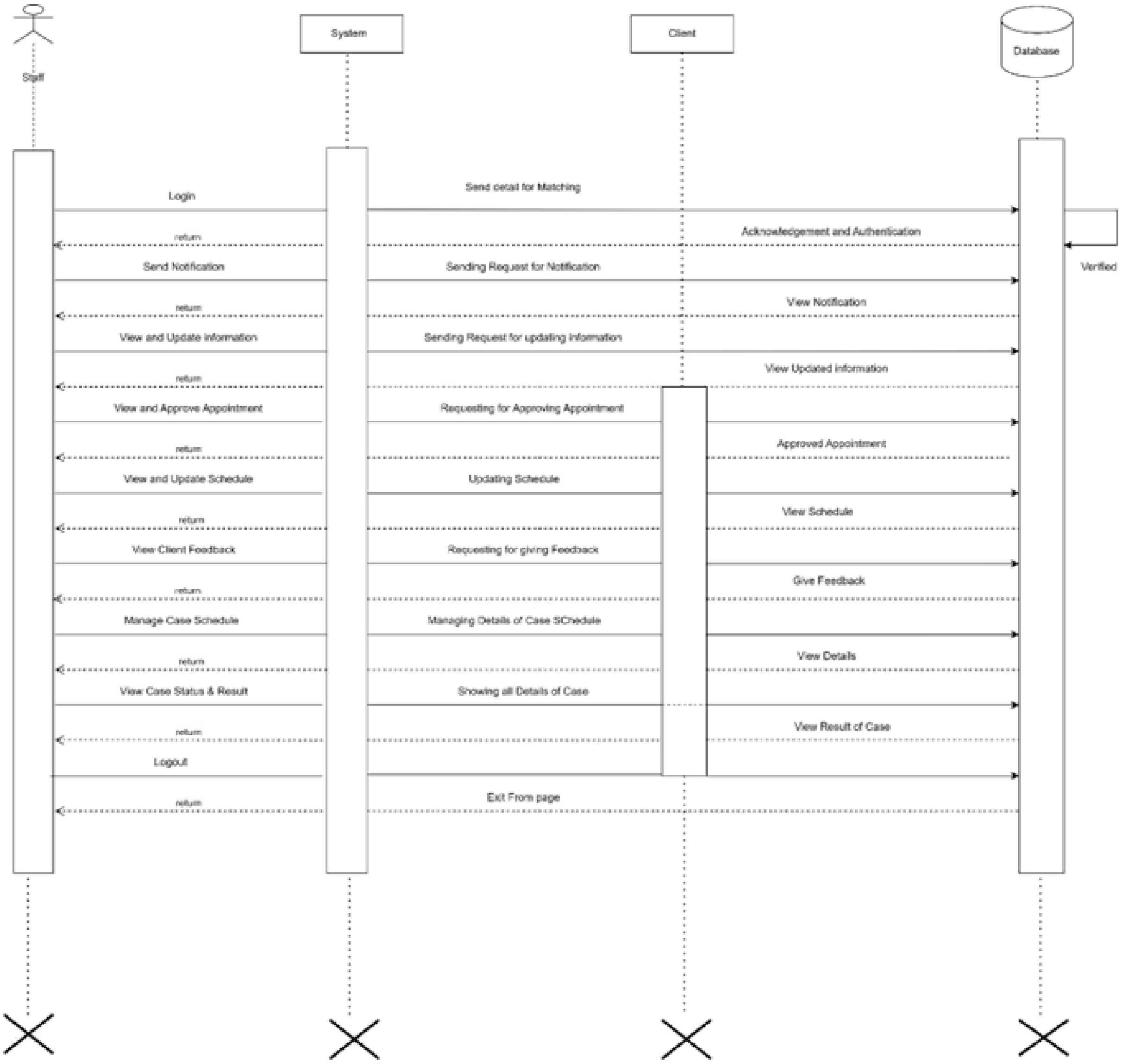
SYMBOL OF SEQUENCE DIAGRAM :-

Symbol	Name	Description
	Actor Lifeline	Represents an external entity or actor interacting with the system. It typically symbolizes a user or an external system
	Activation Bar	It indicating the period during which a participant is actively involved in the interaction.
	Message Arrow	Arrow connecting two participants, indicating the flow of a message or communication between them.
	Return Message	Represents the return of control or a response message from the receiver back to the sender, typically indicated by a dashed line with a filled arrowhead.
	Self-Message	A message sent from a participant to itself, often used to represent internal processing or method calls within the same object or entity.
	Object Destruction	An "X" or similar notation at the end of a lifeline, indicating the termination or destruction of an object or participant.

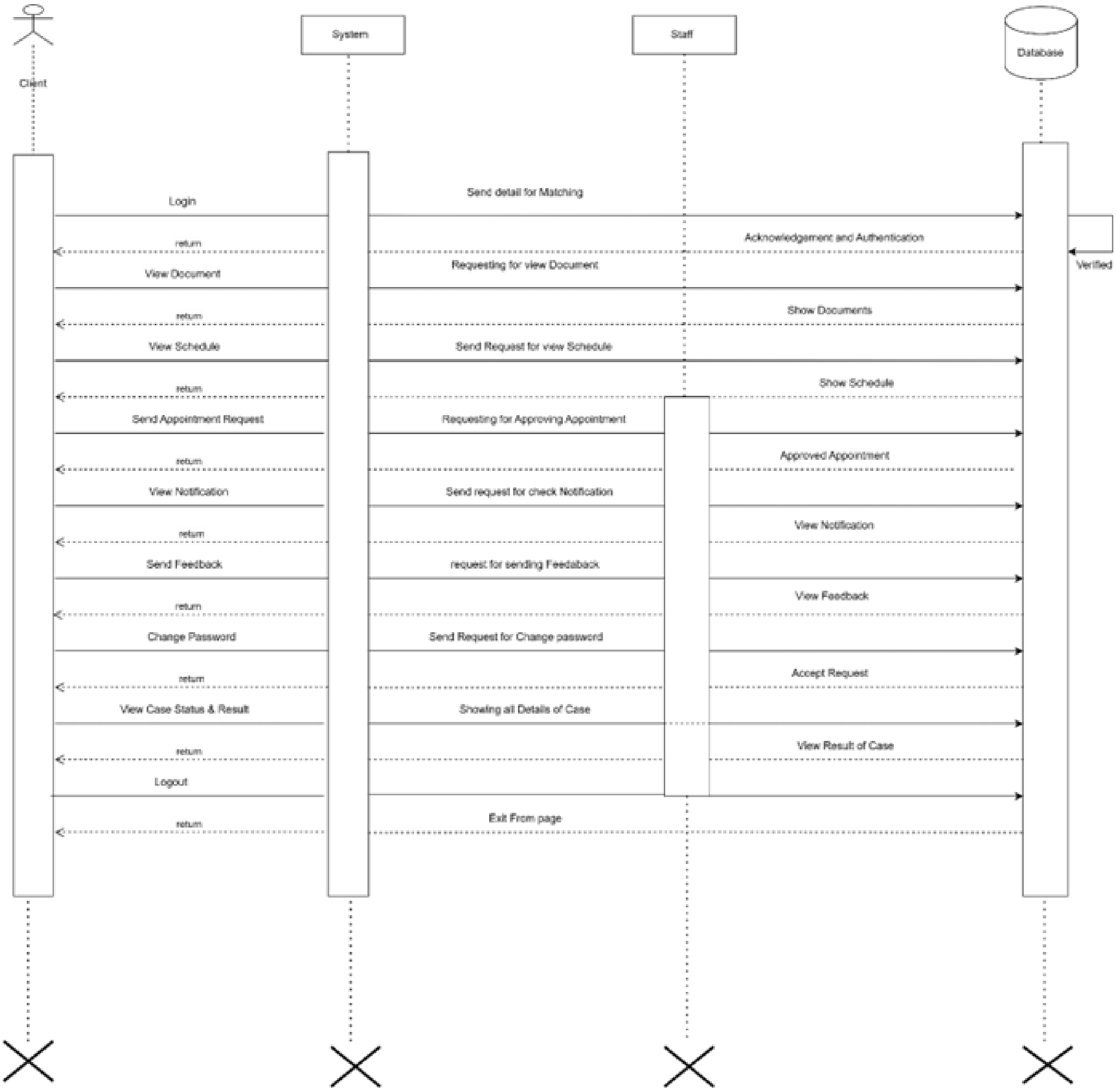
Admin:


Admin
System
Staff
Client
Database
Log in
Send information for matching
return
Acknowledgement and Authentication
Verified
Create Profile
Requesting for make Profile
return
Accept Request
Delete profile
send request for Deleting Profile
return
Deleting profile
Send Notification
Sending Request for Notification
return
View Notification
View Appointment Request
Sending Request for Appointment
return
View Request
View Client Feedback
request for giving Feedback
return
Send Feedback
View case Schedule
Sending details of case
return
View all Details
View & Access case information
Sending access case information
return
View Case information
View Case Status & Result
Showing all deatils of Case Status
return
View Result of Case
Logout
return
Exit from page

Staff:


Staff
System
Client
Database
Login
Send detail for Matching
return
Acknowledgement and Authentication
Verified
Send Notification
Sending Request for Notification
return
View Notification
View and Update information
Sending Request for updating information
return
View Updated information
View and Approve Appointment
Requesting for Approving Appointment
return
Approved Appointment
View and Update Schedule
Updating Schedule
return
View Schedule
View Client Feedback
Requesting for giving Feedback
return
Give Feedback
Manage Case Schedule
Managing Details of Case SChedule
return
View Details
View Case Status & Result
Showing all Details of Case
return
View Result of Case
Logout
return
Exit From page

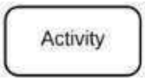
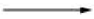
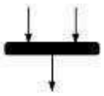
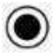
Client:


Client
System
Staff
Database
Login
Send detail for Matching
return
Acknowledgement and Authentication
Verified
View Document
Requesting for view Document
return
Show Documents
View Schedule
Send Request for view Schedule
return
Show Schedule
Send Appointment Request
Requesting for Approving Appointment
return
Approved Appointment
View Notification
Send request for check Notification
return
View Notification
Send Feedback
request for sending Feedaback
return
View Feedback
Change Password
Send Request for Change password
return
Accept Request
View Case Status & Result
Showing all Details of Case
return
View Result of Case
Logout
return
Exit From page

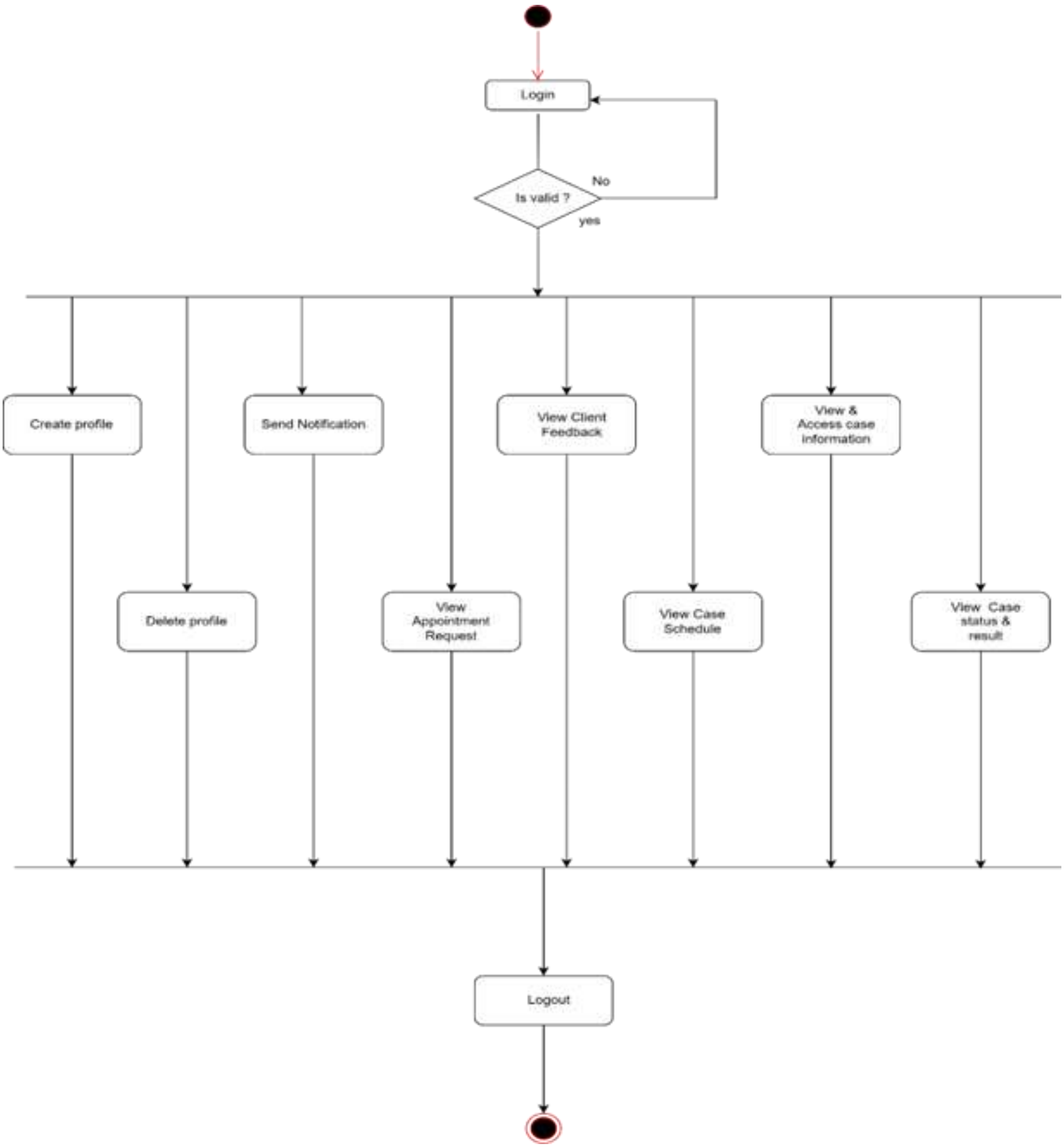
3.7 Activity Diagram

An Activity Diagram is essentially a flowchart that provides a graphical representation of a system's step-by-step activities and actions. It supports choices, iterations, and concurrent processes within a workflow.

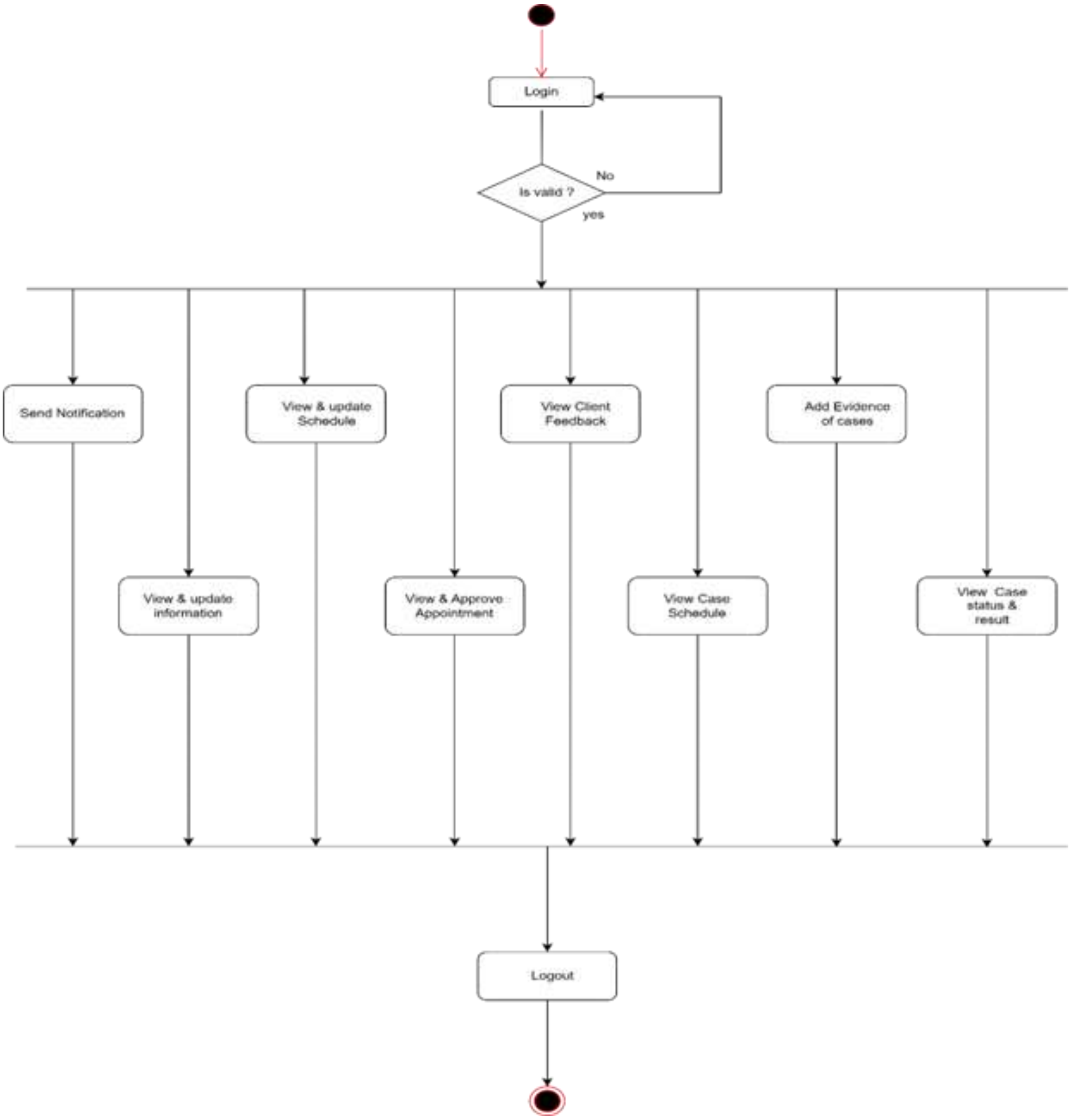
SYMBOL OF ACTIVITY DIAGRAM :-

Symbol	Name	Description
	Start symbol	Represents the beginning of a process or workflow in anactivity diagram. It can be used by itself or with a note symbol that explains the starting point.
	Activity symbol	Indicates the activities that make up a modeled process.These symbols, which include short descriptions within the shape, are the main building blocks of an activity diagram
	Connector symbol	Shows the directional flow, or control flow, of the activity. An incoming arrow starts a step of an activity, once the step is completed, the flow continues with theoutgoing arrow
	Joint symbol/ Synchronization bar	Combines two concurrent activities and reintroduces them to a flow where only one activity occurs at a time. Represented with a thick vertical or horizontal line.
	Decision symbol	Represents a decision and always has at least two paths branching out with condition text to allow users to view options. This symbol represents the branching or mergingof various flows with the symbol acting as a frame or container
	End symbol	Marks the end state of an activity and represents thecompletion of all flows of a process.

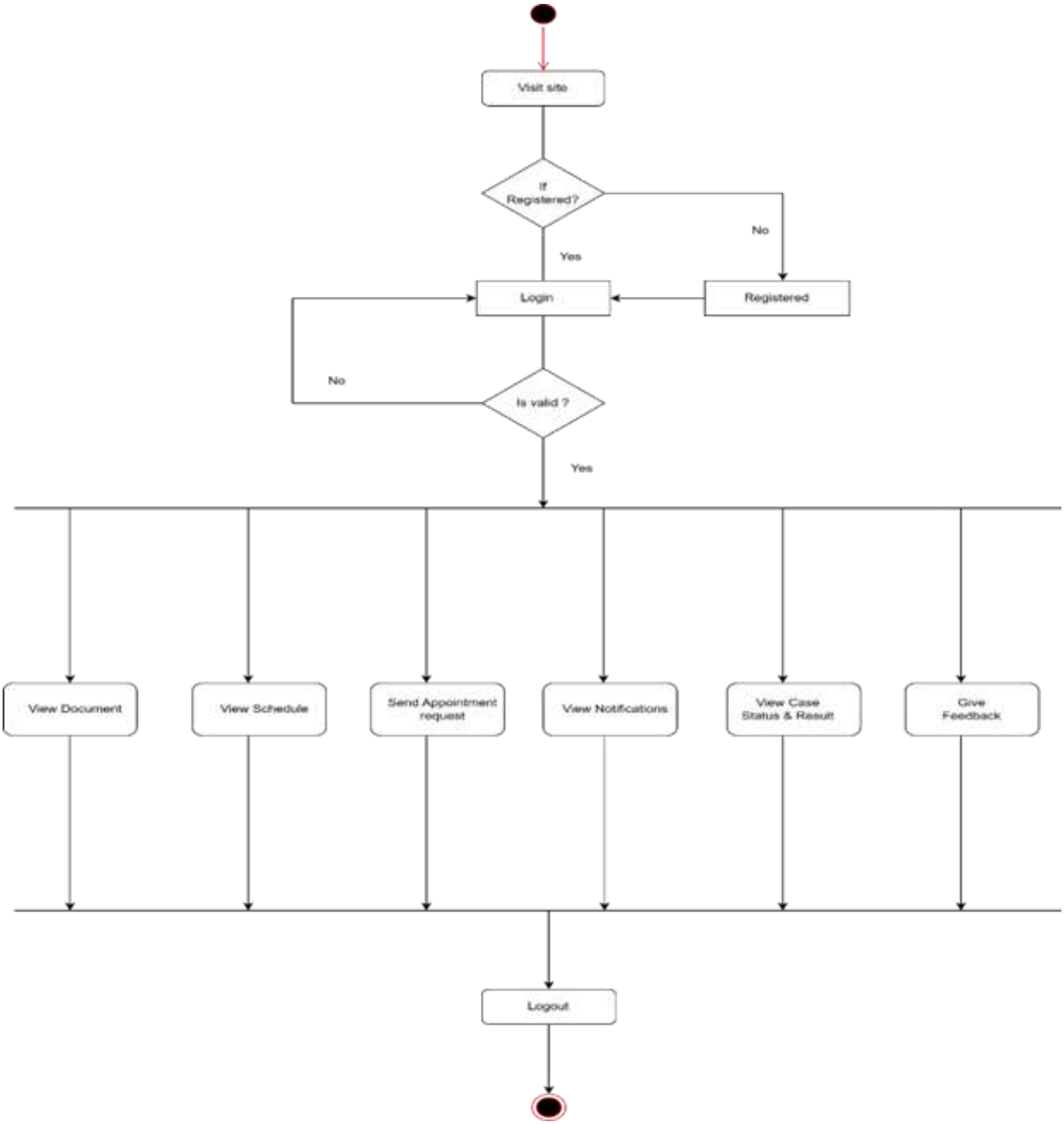
Admin:


Login
Is valid ?
No
yes
Create profile
Send Notification
View Client Feedback
View & Access case information
Delete profile
View Appointment Request
View Case Schedule
View Case status & result
Logout

Staff:


Login
Is valid ?
No
yes
Send Notification
View & update information
View & update Schedule
View & Approve Appointment
View Client Feedback
View Case Schedule
Add Evidence of cases
View Case status & result
Logout

Client:


Visit site
If Registered?
No
Yes
Login
Registered
No
Is valid ?
Yes
View Document
View Schedule
Send Appointment request
View Notifications
View Case Status & Result
Give Feedback
Logout

3.8 Data Dictionary

A Data Dictionary is a centralized repository of information about data, such as tables and their fields. It defines the data elements, including their names, data types, constraints, and descriptions, providing a blueprint for the database.

State info:

No	Field_name	Datatype	constraint	Description
1	State_id	Int(11)	Primary key	State id
2	State_name	Varchar(30)	Not null	State name

City info:

No	Field_name	datatype	constraint	Description
1	City_id	Int(11)	Primary key	City id
2	City_name	Varchar(30)	Not null	City name
3	sid	int(11)	Foreign key	Foreign key from state table

Staf f info:

No	Field_Name	DataType	Constraint	Description
1	Staff_id	Int(11)	Primary key	Staff id
2	Username	Varchar(30)	Not null	Username
3	Password	Varchar(30)	Not null	Password
4	F_name	Varchar(20)	Not null	First name
5	L_name	Varchar(20)	Not null	Last name
6	Email	Varchar(30)	Not null	Email
7	Contact	bigInt	Unique	Contact number
8	Qualification	Varchar(20)	Not null	Qualificatio n
9	Experience	Varchar(20)	Not null	Experience
10	Date_of_reg	Date	Not null	Registeratio n Date
11	Address	Text	Not null	Address
12	Sid	Int(11)	Foreign Key	Foreign key from state table
13	Cid	Int(11)	Foreign Key	Foreign key from city table
14	Gender	Varchar(10)	Not null	Admin gender
15	Photo	Varchar(255)	Not null	Photo
16	Age	Int(3)	Not null	Admin age
17	Dob	Date	Not null	Birth date

Client info:

No	Field_Name	DataType	Constraint	Description
1	Client_id	Int(11)	Primary key	client id
2	Username	Varchar(30)	Not null	Username
3	Password	Varchar(30)	Not null	Password
4	F_name	Varchar(20)	Not null	First name
5	L_name	Varchar(20)	Not null	Last name
6	Email	Varchar(30)	Not null	Email
7	Contact	bigInt	Unique	Contact number
8	Qualification	Varchar(20)	Not null	Qualification
9	Experience	Varchar(20)	Not null	Experience
10	Date_of_reg	Date	Not null	Registeration Date
11	Address	Text	Not null	Address
12	Sid	Int(11)	Foreign Key	Foreign key from state table
13	Cid	Int(11)	Foreign Key	Foreign key from city table
14	Gender	Varchar(10)	Not null	Admin gender
15	Photo	Varchar(255)	Not null	Photo
16	Age	Int(3)	Not null	Admin age
17	Dob	Date	Not null	Birth date

Case info:

No	Field_name	Datatype	Constraint	Description
1	Case_id	Int(11)	Primary key	Case id
2	Case_title	Varchar(30)	Not null	Case title
3	description	Text	Not nulll	Case description
4	FIR_copy	Varchar(100)	Not null	Fir copy
5	Police station	Varchar(100)	Not null	Police station
6	Staff_id	Int(11)	Foreign key	Staff id
7	Client_id	Int(11)	Foreign key	Client id
8	Case_type	Varchar(50)	Not null	Case type
9	Case_reg_date	Date	Not null	Case Register date
10	Status	Varchar(10)	Not null	Status
11	Result	Varchar(10)	Not null	Result
12	Sid	Int(11)	Foreign key	Foreign key from state table
13	Cid	Int(11)	Foreign key	Foreign key from city table

Case Documents info:

No.	Field_Name	Data Type	Constraint	Description
1	Doc_id	Int(11)	Primary Key	Doc id
2	Case_id	Int(11)	Foreign Key	Case id
3	Case_title	Varchar(30)	Not Null	Title of case
4	Description	Text	Not Null	Description of case
5	File	Varchar(255)	Not Null	Files
6	date	date	Not null	Date

Case Hearing info:

No	Field_name	Data_types	Constraint	Description
1	Case_hearing_id	Int(11)	Primary key	Case_hearing_id
2	Case_id	Int(11)	Foreign key	Case_id
3	Hearing_date	date	Not null	Hearing date
4	Remarks	Varchar(30)	Not null	Remarks

Case Appointment info:

No	Field_name	Data_types	Constraint	Description
1	Appointment_ id	Int(11)	Primary key	Appointmen t id
2	Date	Date	Not Null	Date
3	Time	Time	Not Null	Time
4	Client_id	Int(11)	Foreign key	Client id
5	Status	Varchar(50)	Not null	Status
6	Description	Text(50)	Not Null	Description

Inquiry info:

No	Field_name	Data_types	Constraint	Description
1	Inquiry_id	Int(11)	Primary key	Inquiry id
2	Inquiry_name	Varchar(30)	Not null	Inquiry name
3	Email_id	Varchar(30)	Not null	Email id
4	Contact	Bigint	Not null	Contact
5	Message	Text	Not null	Message
6	date	date	Not null	date

Feedback info:

No	Field_name	Data_type	Contraint	Description
1	Feed_id	Int(11)	Primary Key	Feedback id
2	Rating	Int(11)	Not Null	Rating
3	Date	Date	Not Null	Date
4	comment	Varchar(30)	Not Null	Comment
5	Client_id	Int(11)	Foreign Key	Client id