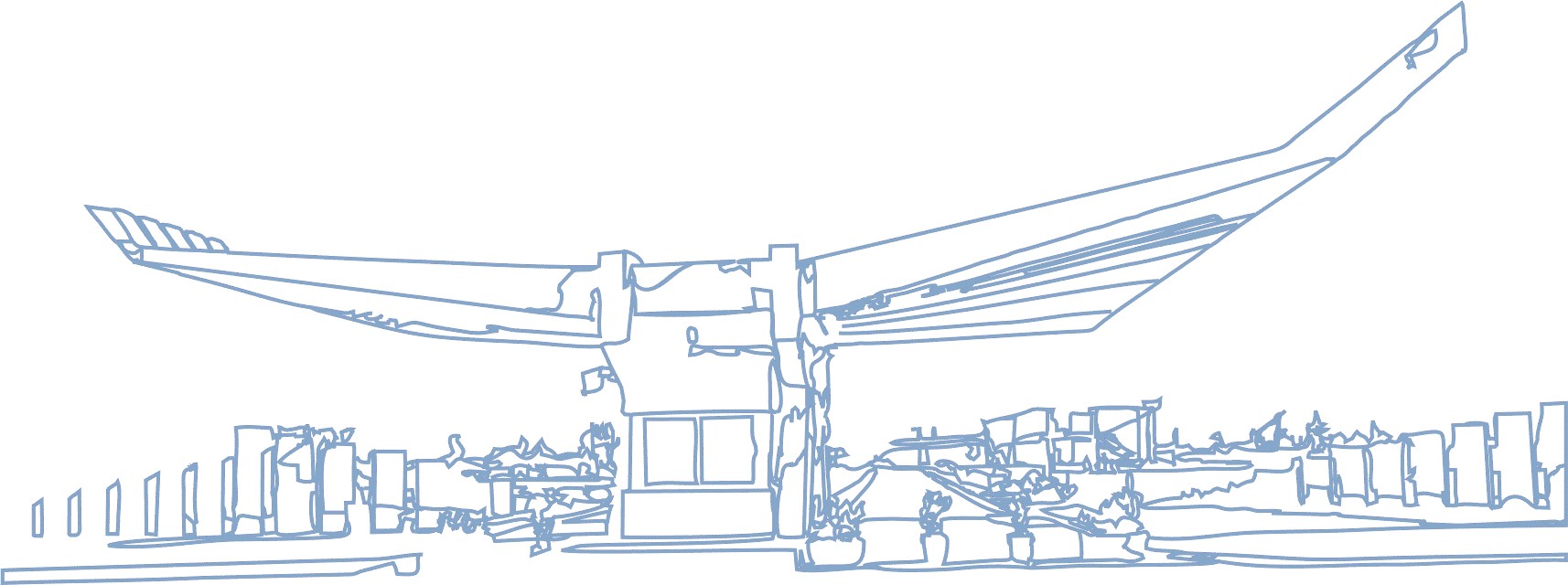
**SOFTWARE MODELING AND DESIGN**

**[Bailiff’s Office Software System]**



**Team Members:**

**Greidi Gjoka [1]**

**Holti Kryemadhi [2]**

**Rei Kurti [3]**

Table of Contents

1. EXECUTIVE SUMMARY………………………………………………………
   1. PROJECT OVERVIEW……………………………………………………………………......
   2. PURPOSE AND SCOPE OF THE SPECIFICATION………………………………………...
2. PRODUCT/SERVICE DESCRIPTION………………………………………….
   1. PRODUCT CONTEXT………………………………………………………………………...
   2. USER CHARACTERISTICS………………………………………………………………….
   3. ASSUMPTIONS………………………………………………………………………………..
   4. CONSTRAINTS………………………………………………………………………………..
   5. DEPENDENCIES………………………………………………………………………………
3. REQUIREMENTS……………………………………………………………………………...
   1. FUNCTIONAL REQUIREMENTS……………………………………………………………
   2. NON-FUNCTIONAL REQUIREMENTS……………………………………………………..
      1. PRODUCT REQUIREMENTS……………………………………………….......
      2. ORGANIZATIONAL REQUIREMENTS………………………………………..
      3. EXTERNAL REQUIREMENTS…………………………………………………
4. SOFTWARE DESIGN…………………………………………………………..
   1. USER SCENARIOS……………………………………………………………………………
   2. USE CASES……………………………………………………………………………………
   3. DATA FLOW DIAGRAMS……………………………………………………………………
   4. STATE DIAGRAMS…………………………………………………………………………..
   5. COLLABORATION DIAGRAMS……………………………………………………………
   6. SEQUENCE DIAGRAMS……………………………………………………………………..
   7. CLASS DIAGRAMS…………………………………………………………………………...
   8. ER DIAGRAMS………………………………………………………………………………..
   9. OBJECT DIAGRAMS………………………………………………………………………….
   10. COMPONENT DIAGRAMS………………………………………………………………….
   11. DEPLOYMENT DIAGRAMS……………………………………………………………..
5. 5. APPENDIX
6. EXECUTIVE SUMMARY

Product Overview

Bailiff is a manager, overseer or custodian; a legal officer to whom some degree of authority or jurisdiction is given. They keep track of given verdicts in court. In civil court proceedings, the court may find one party (the creditor) is entitled to a remedy or judgment from another party (the debtor). The judgment often involves the payment of money for debt, damages or other causes. Creditors can either be individuals, business entities, organizations or the government. The Court Order Enforcement Act sets out a number of options for creditors to collect the judgment. When the debtor does not pay the judgment, the creditor may seek a court order (or writ) to have the judgment enforced by a court bailiff. The most common court orders include a writ of possession, writ of seizure and sale, and order of seizure and sale.

Until now, these offices do not possess a proper means of keeping information and documentation organized, and instead use printed paper only to store said documentation. Storing client information and organizing these cases by said clients proves to be difficult. Instead, a name and a phone number is all that is collected. Employees have to search through documents in order to get previous cases of a specific client. Accountant of the firm has to manually check each document, in order to present a “Revenue and Loss statement”.

For the reasons mentioned above, our group has decided to create a Bailiff Office System, which will not only improve performance, it will increase efficiency of documentation and information gathering. This will also provide easier way of communication between employees and prevent possible errors and inaccuracies.

Purpose and Scope of Specification

The main purpose of the Bailiff Office System Software would be digitalizing the process of information gathering and organizing. Every process needed would be digitized in a system which is structured to allow communication, easy access to information through using simple keywords, or identifying numbers in order to get a specific case. For every change that is committed into the cases, the system will display a log where all changes are listed, allowing other employees to see for mistakes, or to be updated regarding that case. This system also allows the accountant ease of access to the financial department of the firm. The user is based on four levels of access: the administrator (bailiff/owner of the firm), co-bailiff, two lawyers and an accountant.

2. PRODUCT/SERVICE DESCRIPTION

Bailiff-s-Office-Software-System is a web application software that aims to create an efficient way to organize cases. Through this software, bailiffs can add new cases to the system and update current ones in a more systematic way. They can manage and observe their employees better. This system can be used by all members of the firm, with a special level of access for each of them.

PRODUCT CONTEXT

This product is intended to be used by all members of the firm. The employees will be able to access the information much easier and communicate with each other.

USER CHARACTERISTICS

All members of the firm will use the system, which means that there will be different levels of access.

Administrator (highest level):

Administrator of the software will be the owner of the firm, the main bailiff. As the admin, the owner of the firm has the highest level of access. These include: adding new employees, removing employees, managing employee details. They also are in charge of adding new cases, updating/removing adding information regarding the case, and changing the status of the case. Admin can approve/revert changes

Bailiff (acting admin):

The second user with the highest level, the bailiff also has access to adding cases, updating them and changing their status. They can also manage employee information but cannot remove or add employees.

They also approve/revert changes appearing in the log.

Lawyer (user):

Lawyers can edit and update case information, client information and change status of cases. Compared to admin and bailiff, they have lower level of access and less privileges

ASSUMPTIONS

- The system is expected to have access to all data entered by users.

- The information in the system is expected to be accessed based on privileges (access level).

- It is expected for the software to be used in Windows, Mac OS, Linux operating systems, and Android or iOS for mobile use of the software.

- It is assumed that all users are connected to the internet when using the software.

- It is expected for the software to be used in all devices.

- It is expected that all changes to the case documentation and status are to be shown in the log screen in the main page.

CONSTRAINTS

- All members of the firm must be logged in to access the web application.

- All members of the firm must be connected to the internet in order to access the software.

- All members of the firm must have basic knowledge of the web application.

DEPENDENCIES

The admin account cannot change the priority over existing cases if there are less than 2 ongoing cases

• The admin account cannot view revenue and loss statement if it is not uploaded by the accountant

• The admin account cannot manage members if there are no registered members (i.e., the list of members is empty)

• Non-admin employee accounts cannot login to the system until given a password by the admin

• Accountant cannot export data if a financial report has not been made

• Accountant cannot make a financial report and revenue/loss statement if no case fees have been received

1. REQUIREMENTS

FUNCTIONAL REQUIREMENTS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Requirement # | Requirement | Comment | Priority | Date Reviewed | Reviewed/Approved |
| RQ\_01 | Software will have a different levels of access. The owner of the firm will have the highest level of access, followed by a fellow bailiff, lawyers, and accountant who has the lowest access level | Users will have different levels of privileges | 1 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_02 | Every member of the firm will login into the system by a username/ID and a password. | Users will use username or ID to login to the web application | 1 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_03 | Owner of the firm can manage other members, register new members working for the firm or delete users who do not work in the firm anymore. | Owner can manage members, add new ones easier. | 1 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_04 | Owner can add basic details such as: first name, last name, date of birth, position. | Owner adds details such as name, surname and their date of birth | 2 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_05 | Admin provides firm members with a temporary password, which members can change | In order to login, members receive a one-time password to login | 2 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_06 | Admin can change password of their account at "Account" tab. | Admin can change password when they want to | 2 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_07 | Bailiff can change existing password at "Account" tab. | Bailiff can change password when they want to | 3 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_08 | Lawyers can change existing password at "Account" tab. | Lawyer can change password when they want to | 3 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_09 | All employees of the firm, including the owner can upload documents, files of different kind, etc. | This is used for all members to upload new documentation. | 2 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_10 | Bailiffs and lawyers can change the direction where the files and documentation are stored. | Bailiff and lawyer can change status of cases and their documentation | 2 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_11 | Owner of firm can choose priority over different cases and upload the documentation provided by court in the main page for all members of the firm to see. | Owner chooses which case has higher importance and must be done first | 1 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_12 | Admin can add new cases in Main page. | Admin adds new cases | 1 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_13 | Admin can update information regarding case. | Admin can add new details and documents regarding the case | 2 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_14 | Bailiff can update information regarding case. | Bailiff can change and modify the information of the cases | 2 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_15 | Lawyers can update information regarding case. | Lawyer can change and modify the information of the cases | 1 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_16 | Admin can change status of case from "Ongoing" to "Closed". | Status of case is changed to show if it is completed or not | 2 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_17 | Bailiff can change status of case from "Ongoing" to "Closed". | Status of case is changed to show if it is completed or not | 2 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_18 | Lawyers can change status of case from "Ongoing" to "Closed". | Status of case is changed to show if it is completed or not | 2 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_19 | Accountant will be able to get a report of all cases fees by using "Get List". | Accountant uses Get List to get a list of all fees for cases | 2 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_20 | Accountant will have the option to export data from report in Excel file. | Accountant has the option to export the list as an Excel file | 3 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_21 | Accountant can upload a revenue and loss statement through "Upload Statement". | Account will upload a financial report for all expenses, and earnings | 2 | 29/03/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_22 | Admin can see statement uploaded by accountant. | Admin can check on what the accountant has uploaded | 2 | 29/05/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_23 | User activity is posted in a log in the main page. | All work that has been done, is posted in a log where changes and updates are done | 2 | 29/05/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_24 | Admin can be redirected to case which has been changed. | Admin can see what is being changed and be redirected towards the changes. | 2 | 29/05/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_25 | Admin can revert changes made to case if needed. | If admin sees a need to change details, he can be redirected. | 2 | 29/05/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_26 | Admin can add new clients in the "Clients" tab. | New clients are added to “Clients” to get list. | 1 | 29/05/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_27 | Admin can modify client details and add/remove cases in "Clients" tab. | Admin can add, remove or modify client information | 2 | 29/05/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_28 | Each client has a unique client id number. | All clients that are added into the software get a unique id number | 2 | 29/05/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_29 | Bailiff can access "Client" tab and modify client details. | Bailiff can only edit client details, but not add or remove | 2 | 29/05/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_30 | Lawyer can access "Client" tab and modify client details. | Bailiff can only edit client details, but not add or remove | 1 | 29/05/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_31 | Bailiff/Lawyer can add/remove cases from specific clients. | If a client drops a case, case is removed from the client. | 2 | 29/05/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_32 | Admin can add/remove cases from specific clients. | The same thing with the lawyer and bailiff, admin can add and remove cases from specific clients | 1 | 29/05/2022 | Holti Kryemadhi/Rei Kurti |
| RQ\_33 | Admin can add/remove clients from "Clients" tab. | Admin can add and remove clients | 1 | 29/05/2022 | Holti Kryemadhi/Rei Kurti |

NON-FUNCTIONAL REQUIREMENTS

Product Requirements

User Interface Requirements

• As a web-based app, it must support all modern browsers such as Google Chrome, Mozilla, and Microsoft Edge etc.

• Interface must be user-friendly and easy to navigate, even for users who may not be tech-savvy.

• Login page requiring the user to enter their username/ID and password.

• Main page that shows a dashboard and an overview of recent activity.

• Management page, where users that have the permission can manage other lower-ranking users.

• Cases page, which shows a list of cases that can be sorted (e.g., by deadline, or whether they are finished or not.) Also, cases can be favorite for easier access or given higher priority.,

Usability

• Web app must be able to be accessed by users with only an internet connection.

• Must be easy to use, navigate, and easy to remember even for users not familiar with similar apps.

• Must allow for easy and quick completion of tasks by the users.

Efficiency:

Performance Requirements

Pages must load fast (although it may depend on users’ internet connection, it would be preferable to have a loading time of less than 2 seconds).

Requires a powerful server machine with good internet connection. Web at the same time and on the operating system that is installed in the server.

Space Requirement

Must handle 30 users simultaneously (the software must be able to handle enough members even though the firm has 5-6 members)

Dependability  
 The app must have a low failure rate (less than 2%)

In case of a crash, it must display standard error message to user

Availability

The web-app must be accessible by users at all times (with the exception of maintenance)

Maintenance must take a short time (a maximum of several hours)

Security

Crucial information such as user data must be protected from any potential malware or cyber attacks

ORGANIZATIONAL REQUIREMENTS

Environmental Requirements

Server machine that will be stored inside the bailiff’s office, and will occupy approximately a 1x1 meter space. Power required will be the power necessary for running the server, which should be optimized to be as efficient as possible.

Operational Requirements

The Web App will have different access levels depending on the user’s role within the organization.

• Owner – can manage other members, change priority of cases.

• Employees – can upload documents, files, etc.

• Higher ranks will also have all the access that lower ranks have.

Development Requirements

1. Client Side Programming (Front-end)

Technologies to be used in client-side web development, that involves everything users see on their screens, will be:

• Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS).

• JavaScript (JS), to make web pages interactive. We will be using jQuery library.

1. Server-Side Programming(Back-end)

• Programming language: PHP

• To store the data: MySQL database

• Server to handle requests from clients’ computers. Will be physically stored inside the bailiff’s office

EXTERNAL REQUIREMENT

Regulatory Requirements

Information collection and usage must be done fully in compliance with Albanian laws for protection of personal data.

Ethical Requirements

Only the necessary information will be collected from external entities such as clients

The office may only share such information if required do do so by a court of law or Albanian legislation

Legislative Requirements

Based on Law No.9887, dated 10.03.2008, “On the Protection of Personal Data”, all personal user data is to remain private, and sensitive data such as passwords are to be secured via encryption.

DOMAIN REQUIREMENTS

1. SOFTWARE DESIGNS

User Scenarios

Case 1. User successful login attempt

A user of the firm enters the correct credentials for username/email and password.

The user is logged in and redirected to the main screen.

Case 2. User failed login attempt

User enters the incorrect credentials for username/email and password.

User cannot log in and is prevented from moving further.

User will be displayed a message where they are required to enter the correct credentials.

Case 3. Bailiff posts new case

Bailiff logs in to the web app.

Bailiff is redirected to the main page-

User clicks on the “Add New Case” button.

Necessary documentation and information are added.

The case is posted on the main page, where other members can access its information.

Case 4. Firm members view new case

User logs in to the web app.

User is redirected to the main page.

User clicks on the new case that is added to the main page.

User can access the information posted by the bailiff.

Sorting documents

Case 5. User logs in to the web app.

User goes to the “Cases” button, where they are redirected to the page where cases are stored.

User selects between two categories, ”Ongoing Cases” and “Closed Cases”.

User chooses the sort option to sort documents by specified criteria.

Case 6. Updating Case status

User logs in to the web app.

User goes to the “Cases button, where they are redirected to the page where cases are stored.

User selects between two categories, “Ongoing Cases” and “Closed Cases”.

User changes the status of the case from “Ongoing” to “Closed”, and vice-versa.

Case 7. Accountant generates revenue and loss statement

Accountant logs in to the web app.

User goes to the “Cases” button, where they are redirected to the page where cases are stored.

User selects the “Closed Cases” category.

User gets a list of all fees recorded for closed cases with “Get List”.

User uploads an income statement sheet, for the admin to see.

Case 8. User logout

User is logged in.

User clicks on logout button.

User is successfully redirected to login screen.

Case 9. User failed logout attempt

User is logged in.

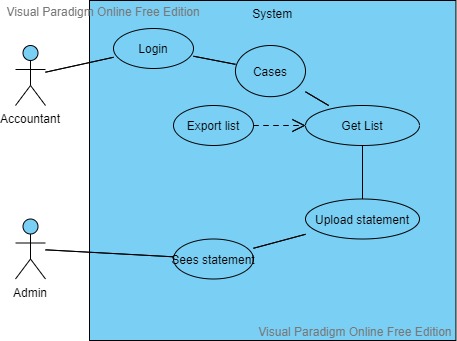
User click on logout button.

User in not redirected to login screen.

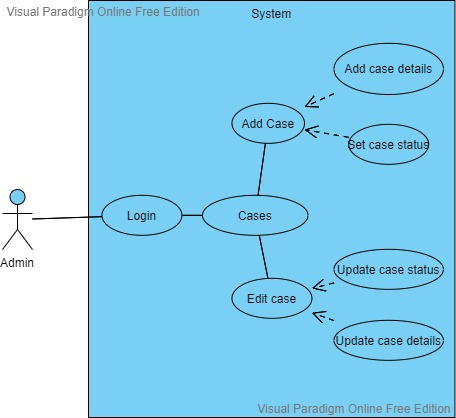
User is displayed message that shows “Error! Something went wrong!”.

Use Cases

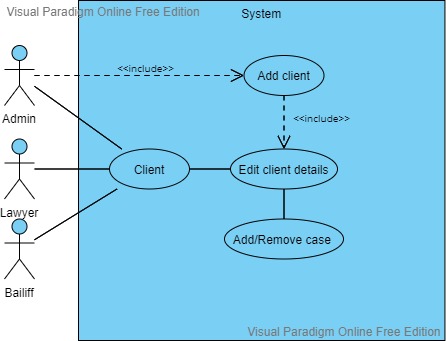
Use Case 1: Accountant Report



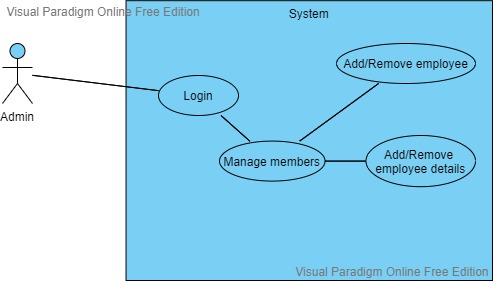
Use Case 2: Admin Case Management



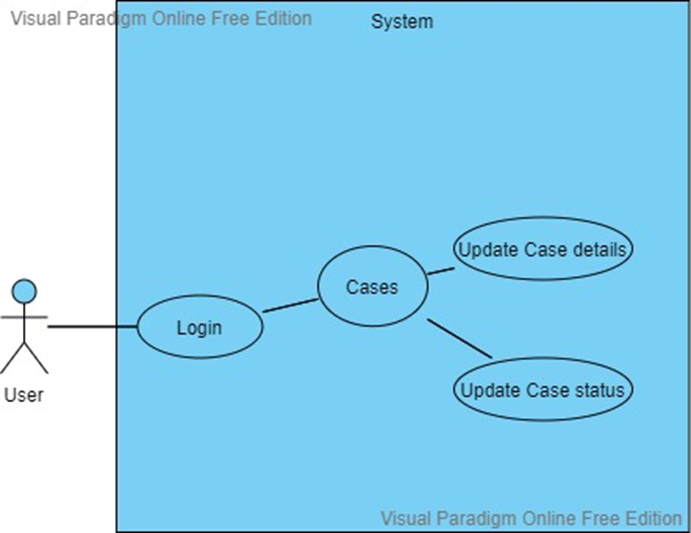
Use Case 3: Client Management



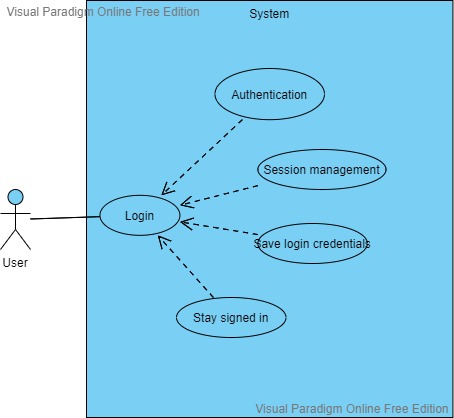
Use Case 4: Edit Members



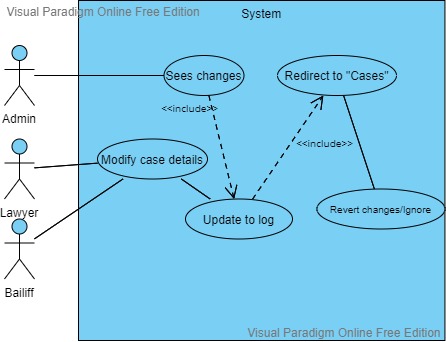
Use Case 5: Update Case



Use Case 6: Use Login

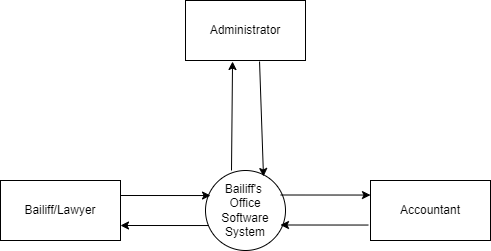
****

Use Case 7: User Activity

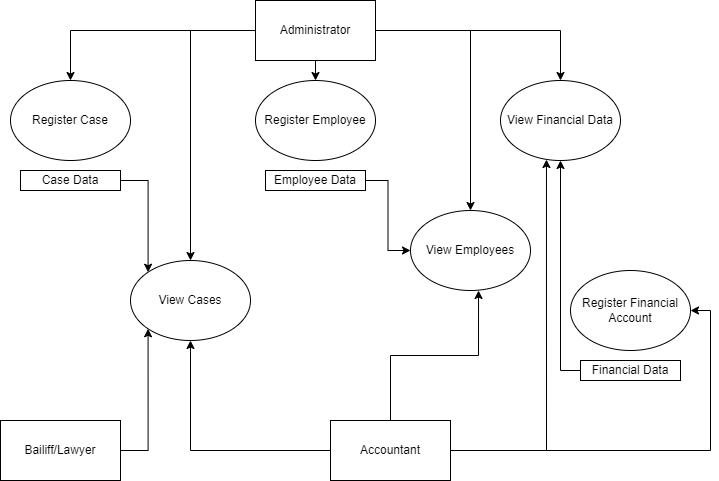
****

DATA FLOW DIARGAMS

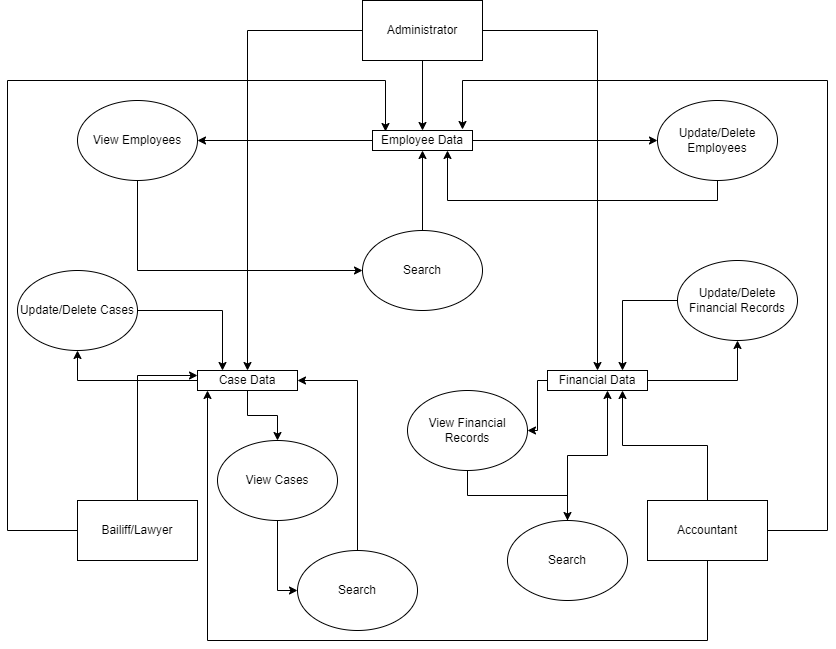
1. DFD: Level 0

****

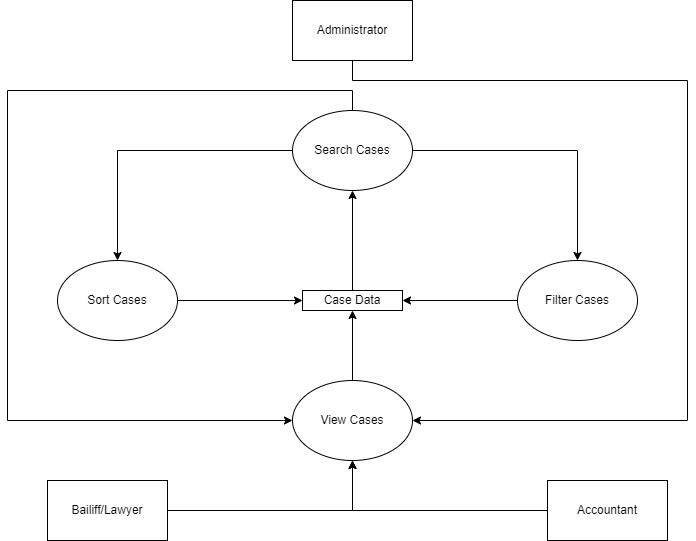
1. DFD: Level 1



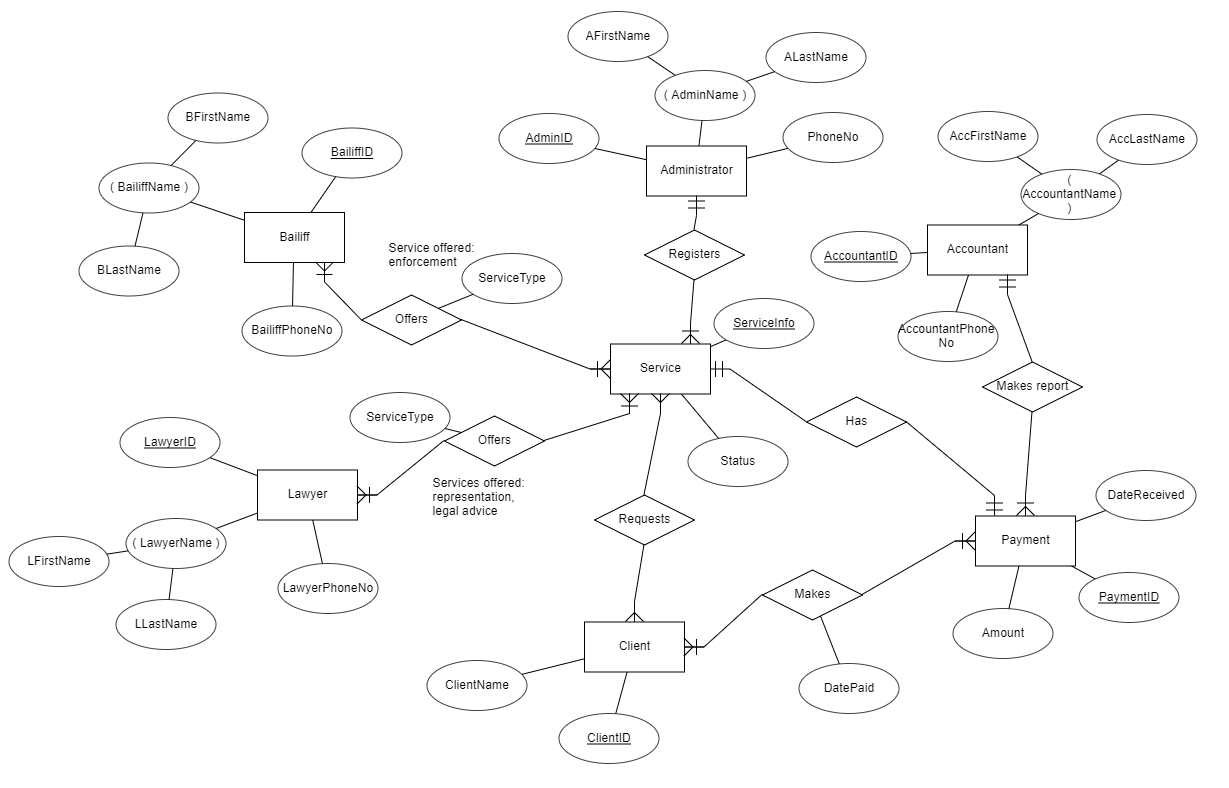
1. DFD: Level 2



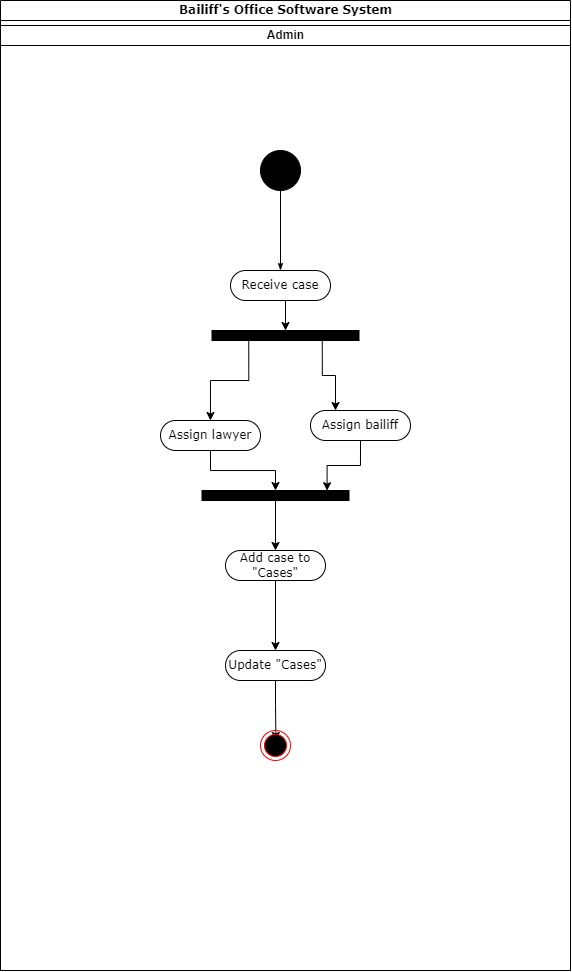
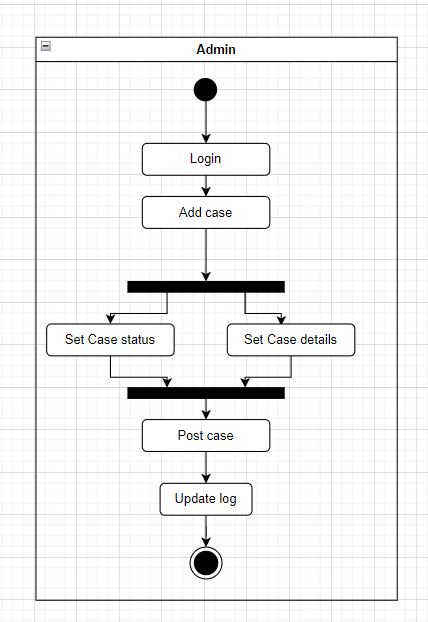
1. DFD: Level 3

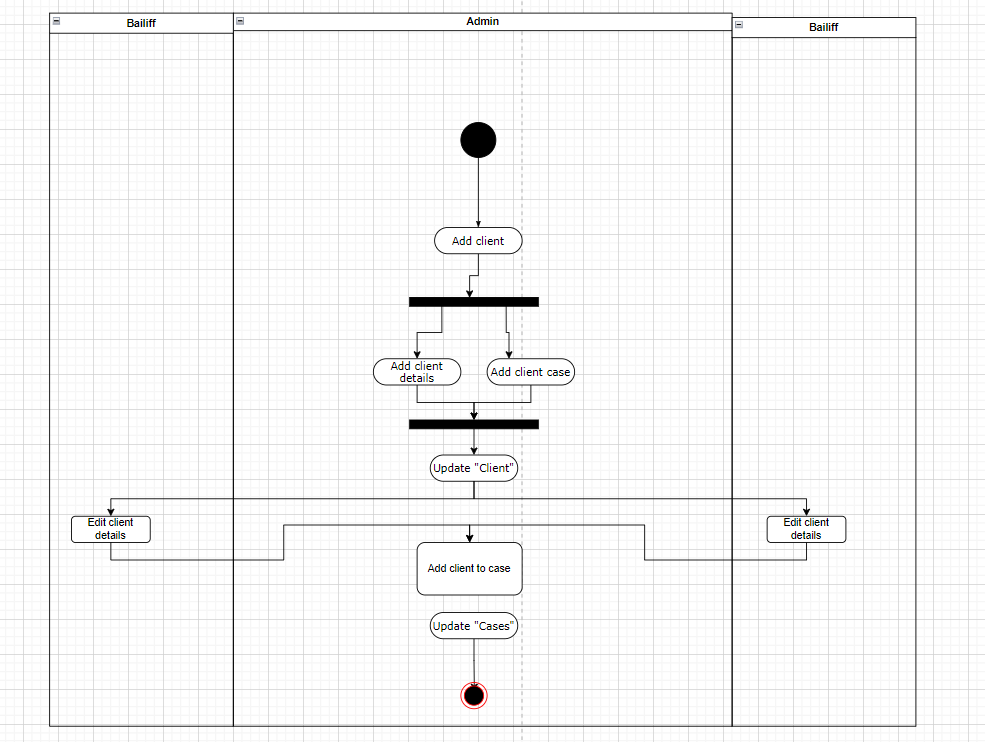
****

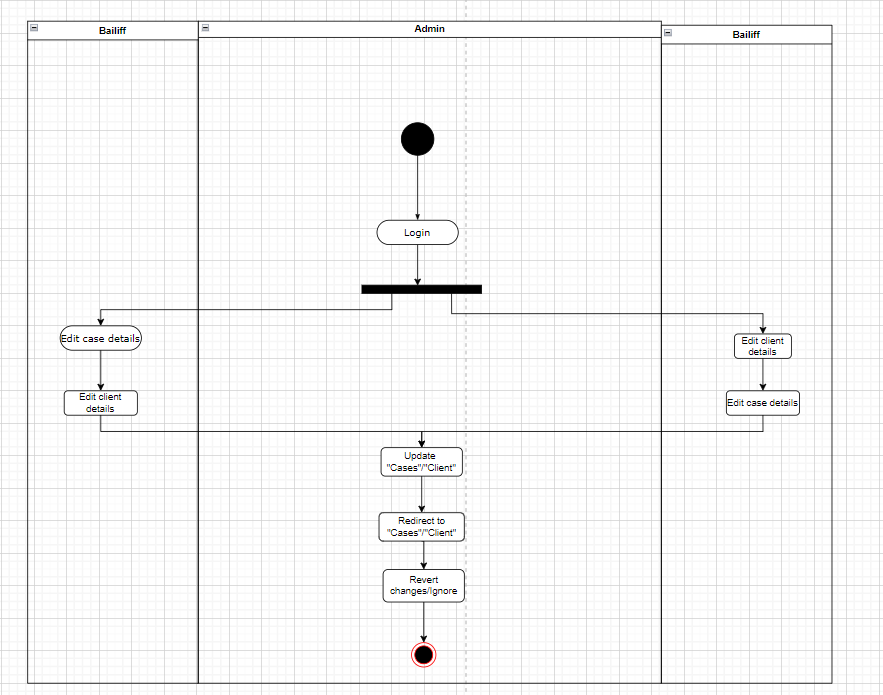
ENTITY RELATIONSHIP DIAGRAM

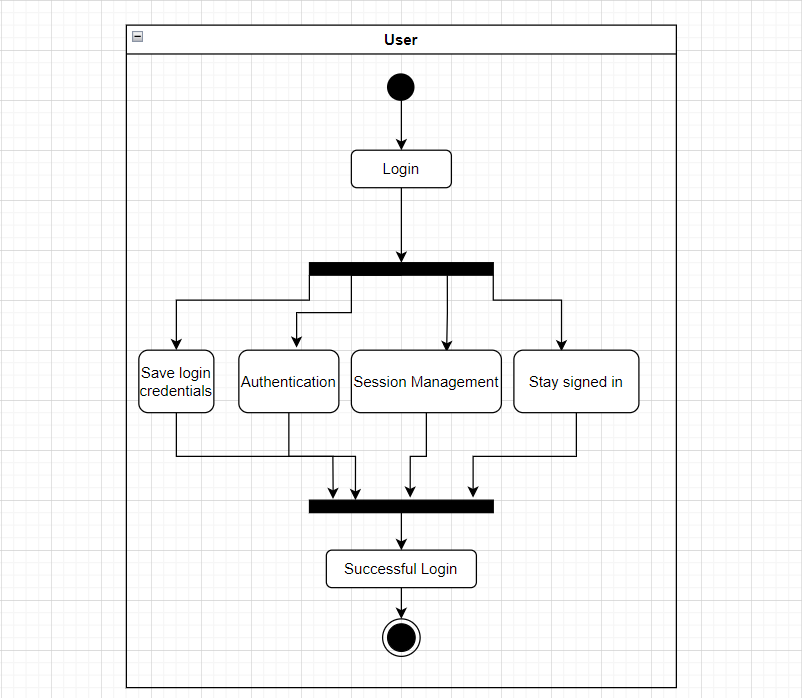


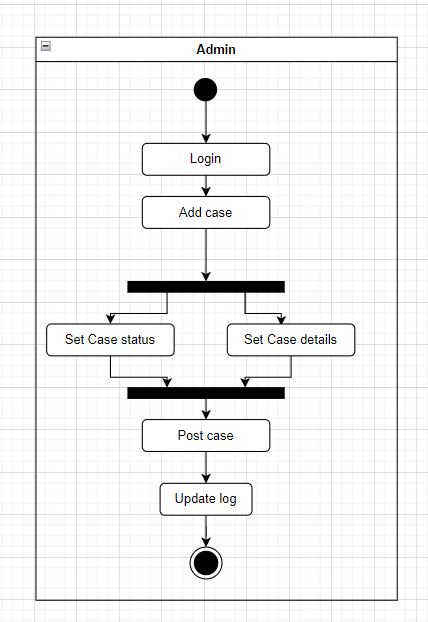
ACTIVITY DIAGRAMS

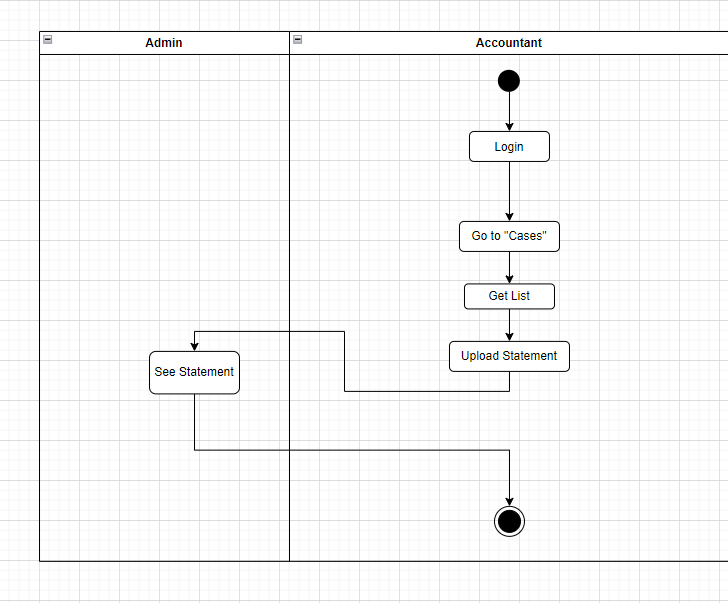






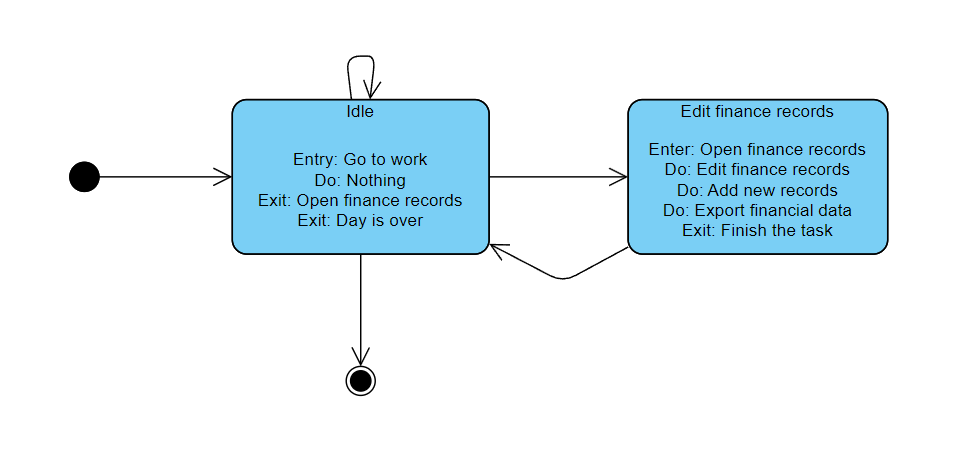




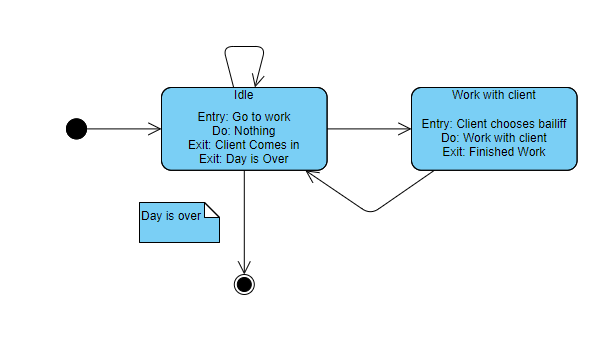


STATE DIAGRAMS

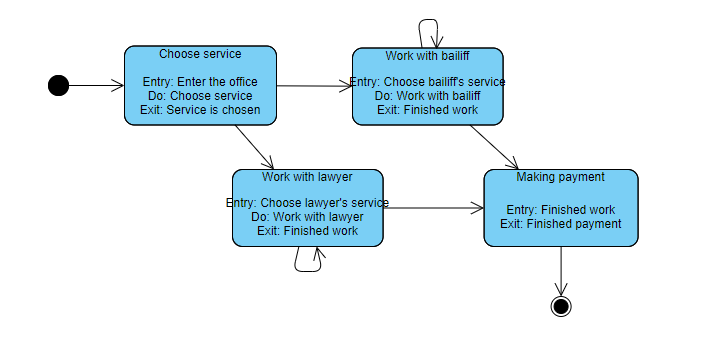
Accountant



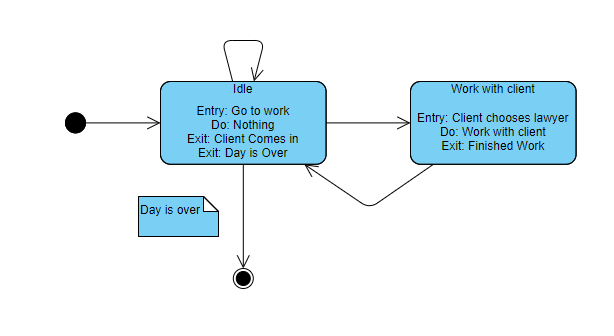
Bailiff



Client

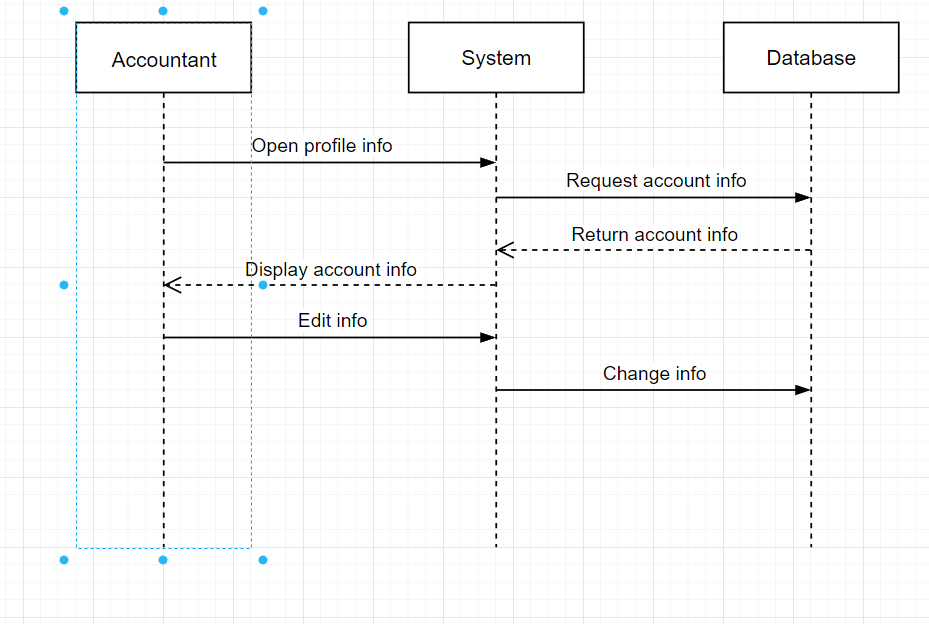


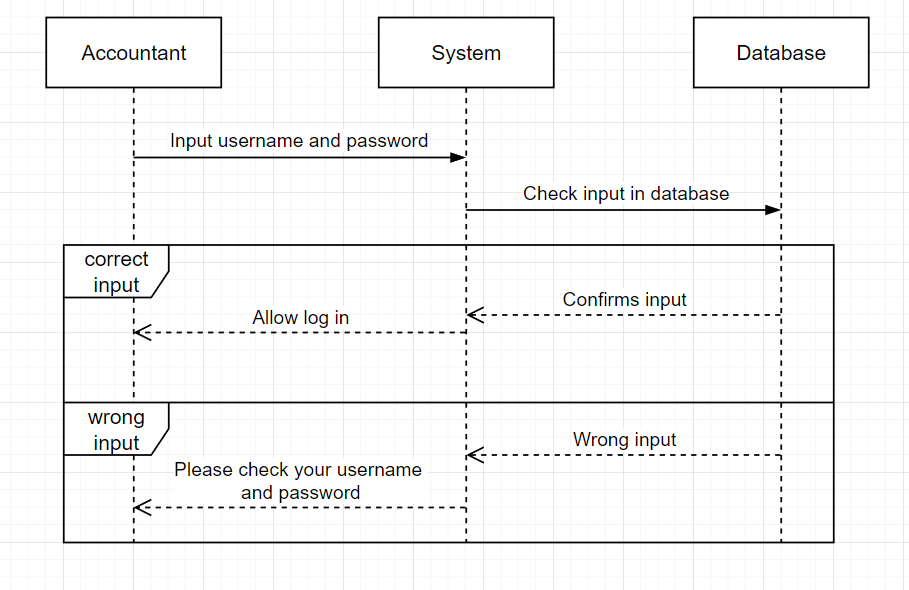
Lawyer

****

SEQUENCE DIAGRAMS

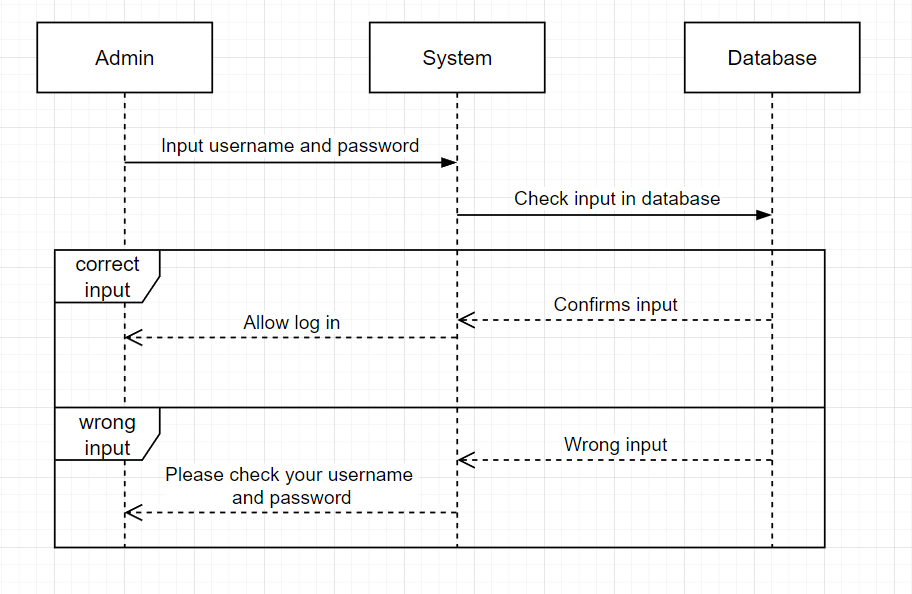
Account Edit Info

****

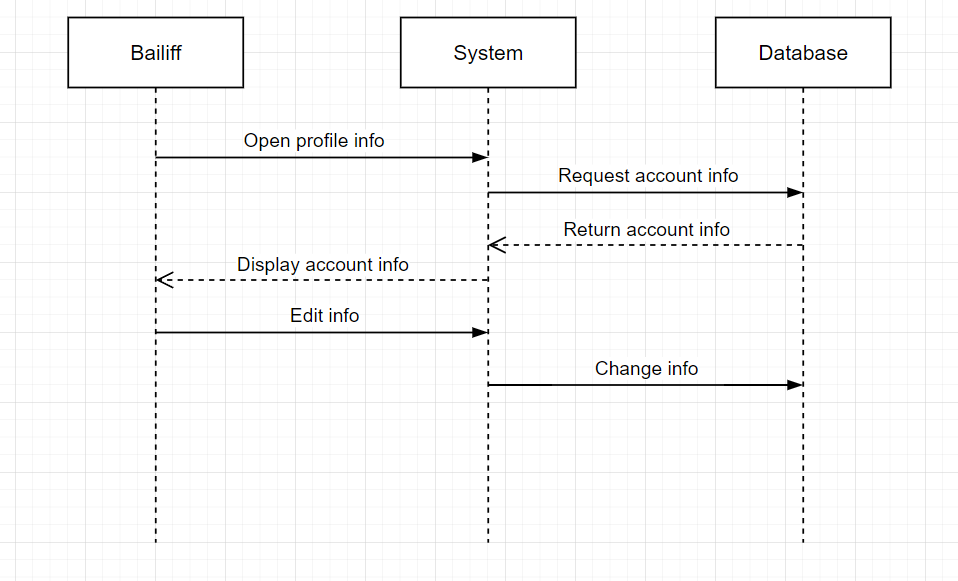
****

Account Login

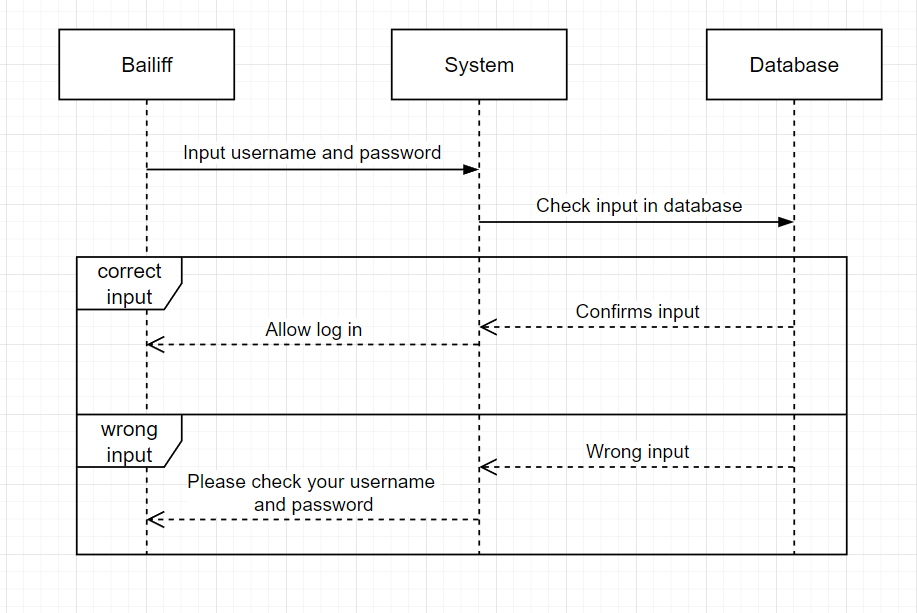
Admin Login

****

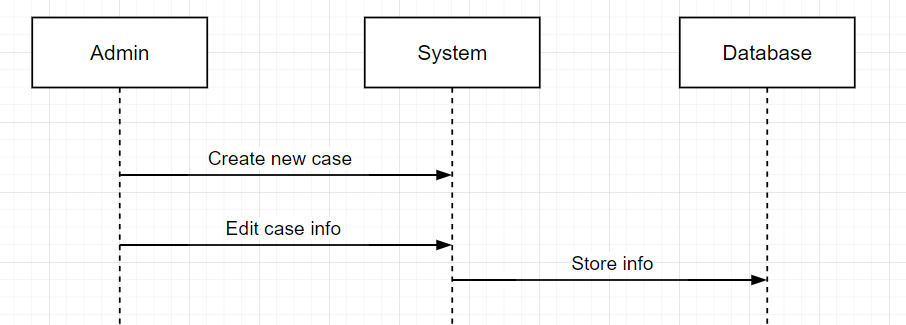
Bailiff Edit Info

****

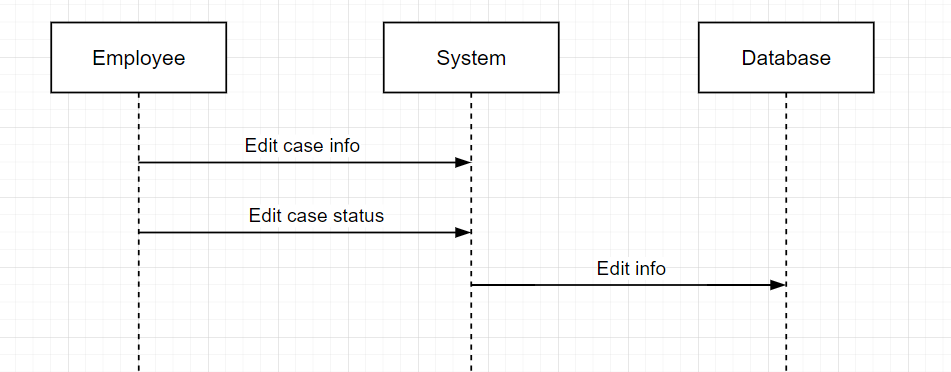
Bailiff Login

****

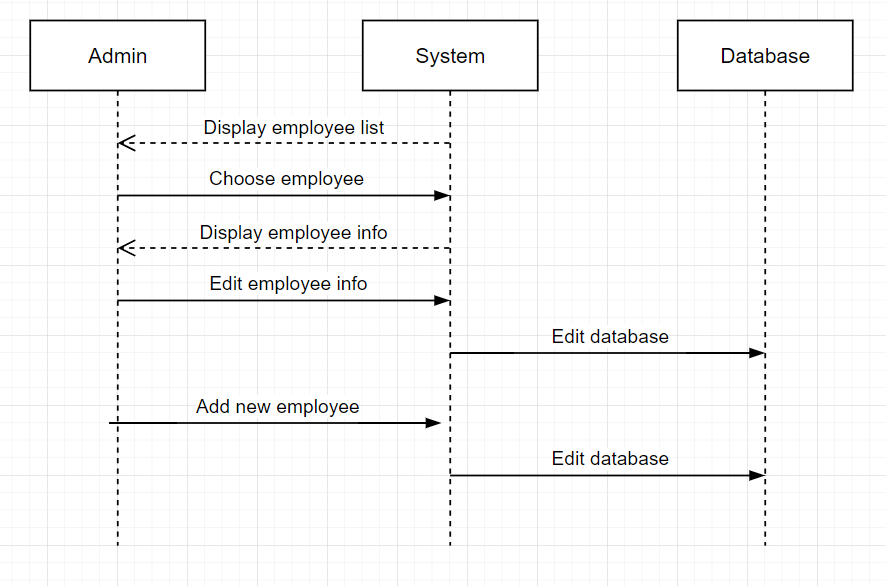
Create Case

****

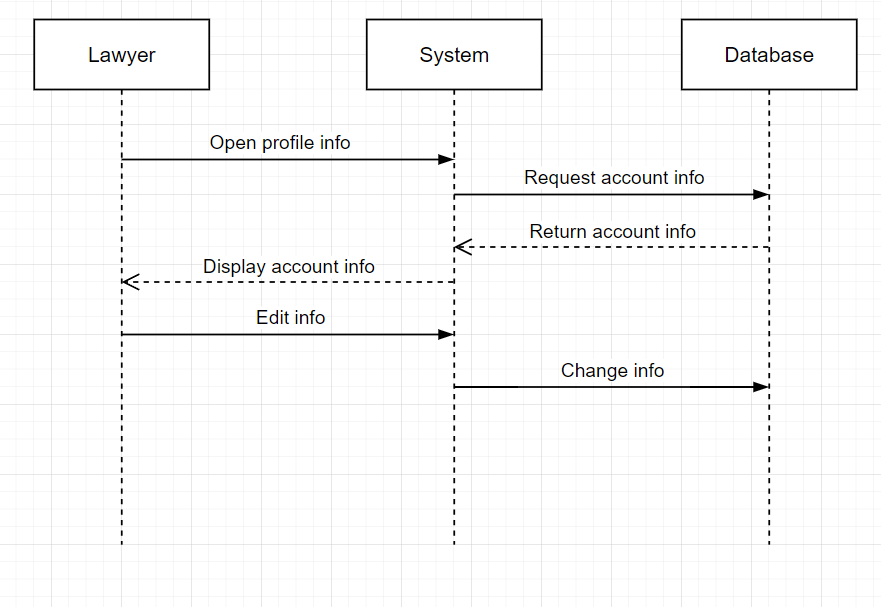
Edit Case

****

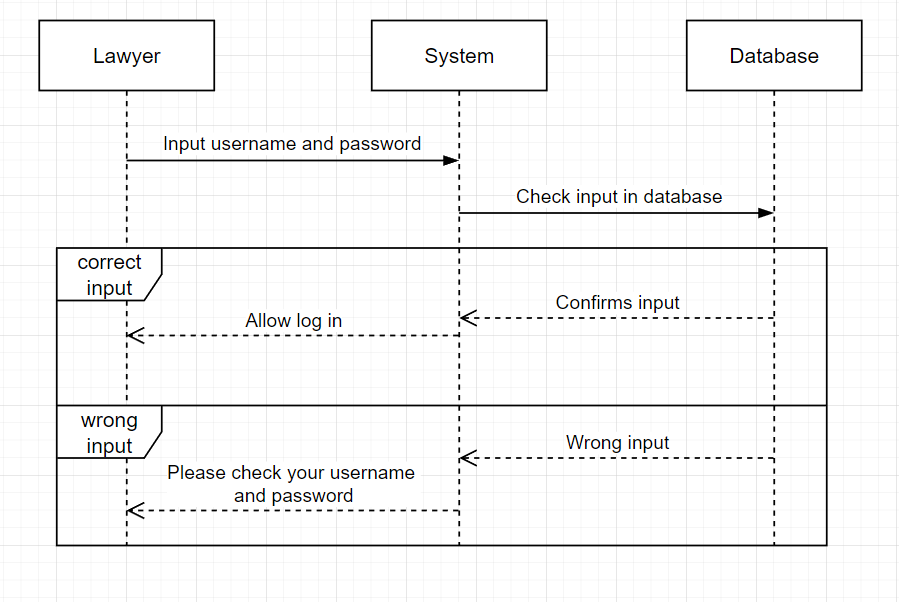
Edit Employee List

****

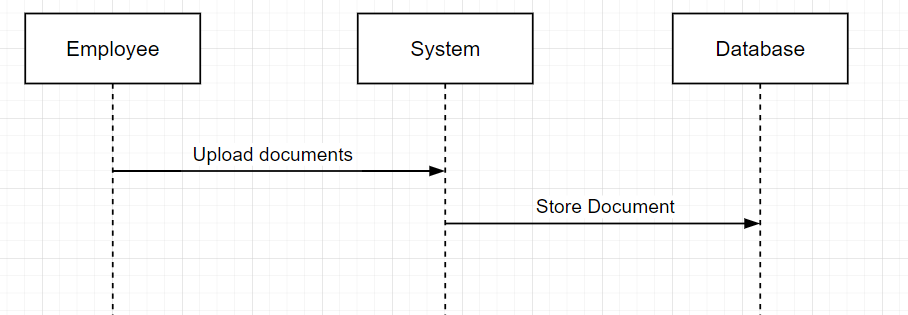
Lawyer Edit Info

****

Lawyer Login

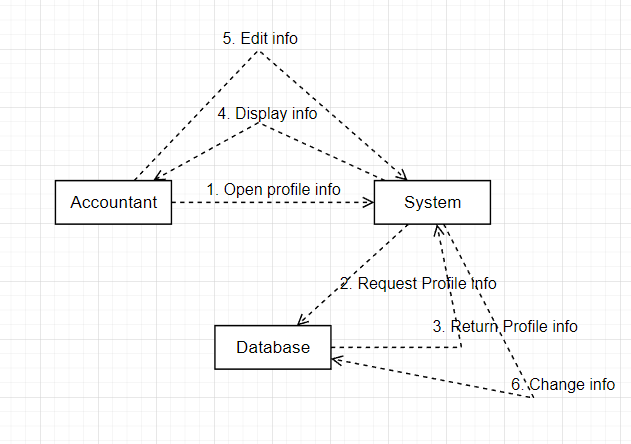
****

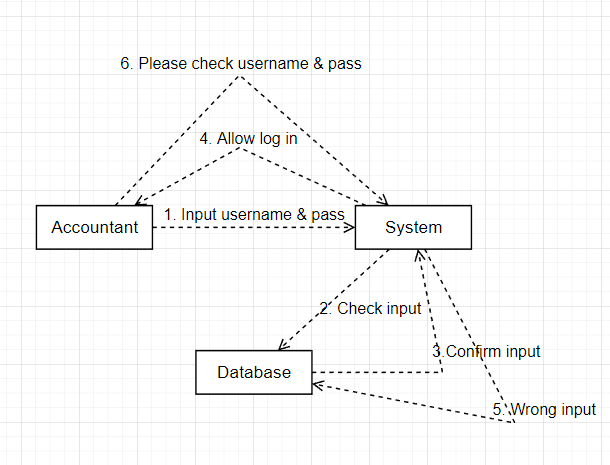
Upload Documents

****

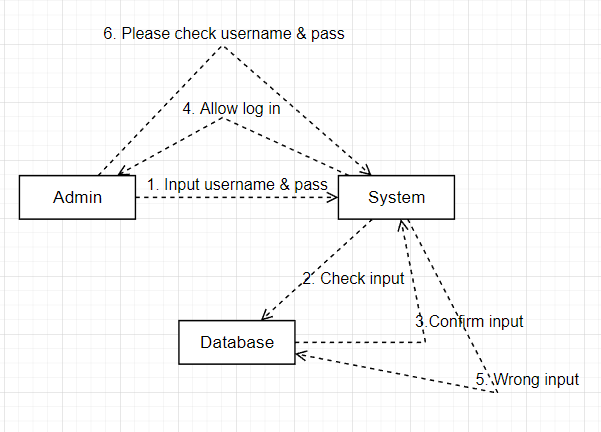
COLLABORATION DIAGRAMS

Account Edit Info

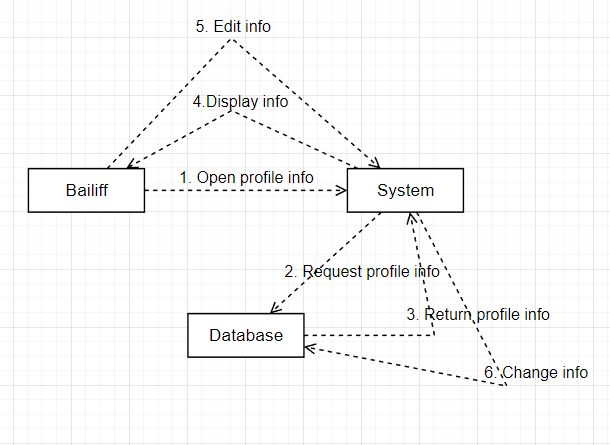
****

****Accountant Login

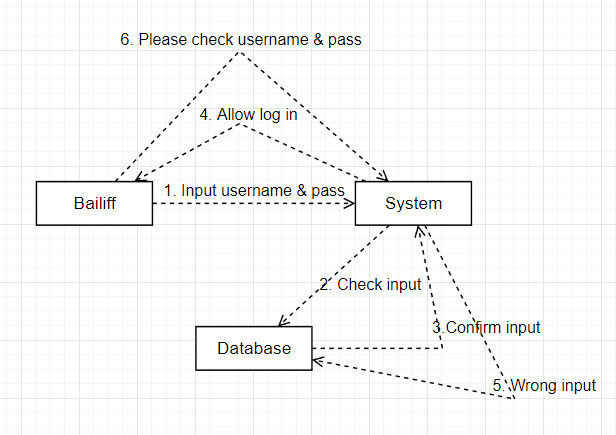
Admin Login

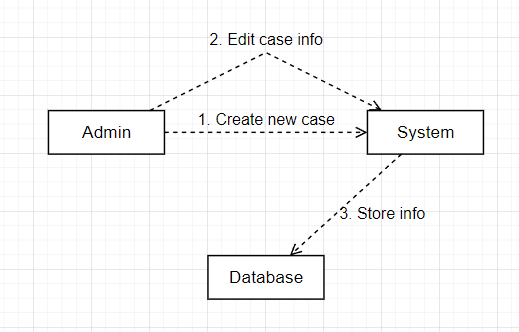
****

Bailiff Edit Info

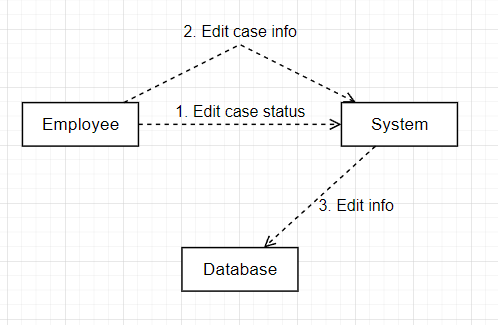
****

Bailiff Login

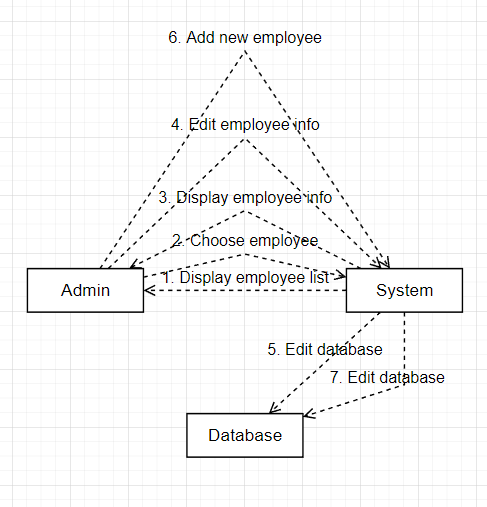
****

****Create Case

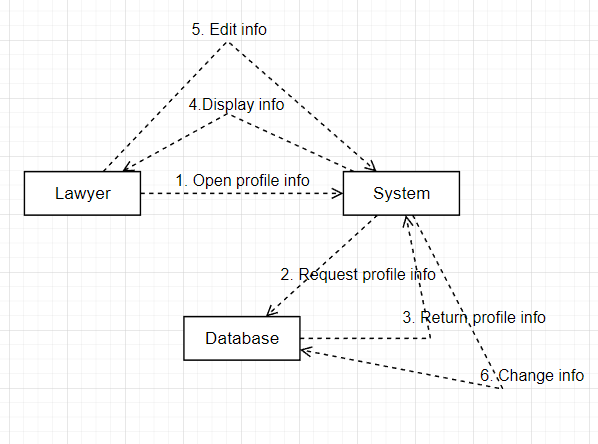
Edit Case

****

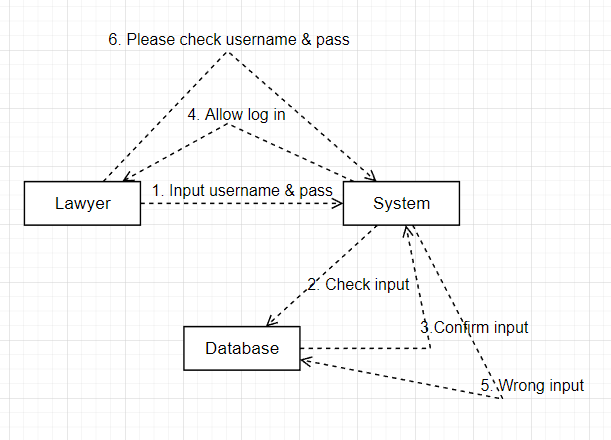
Edit Employee List

****

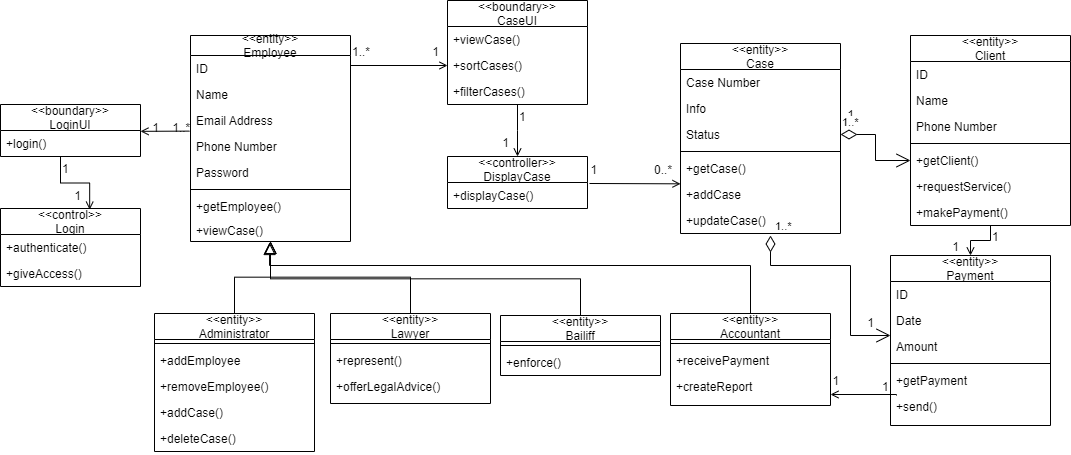
Lawyer Edit Info

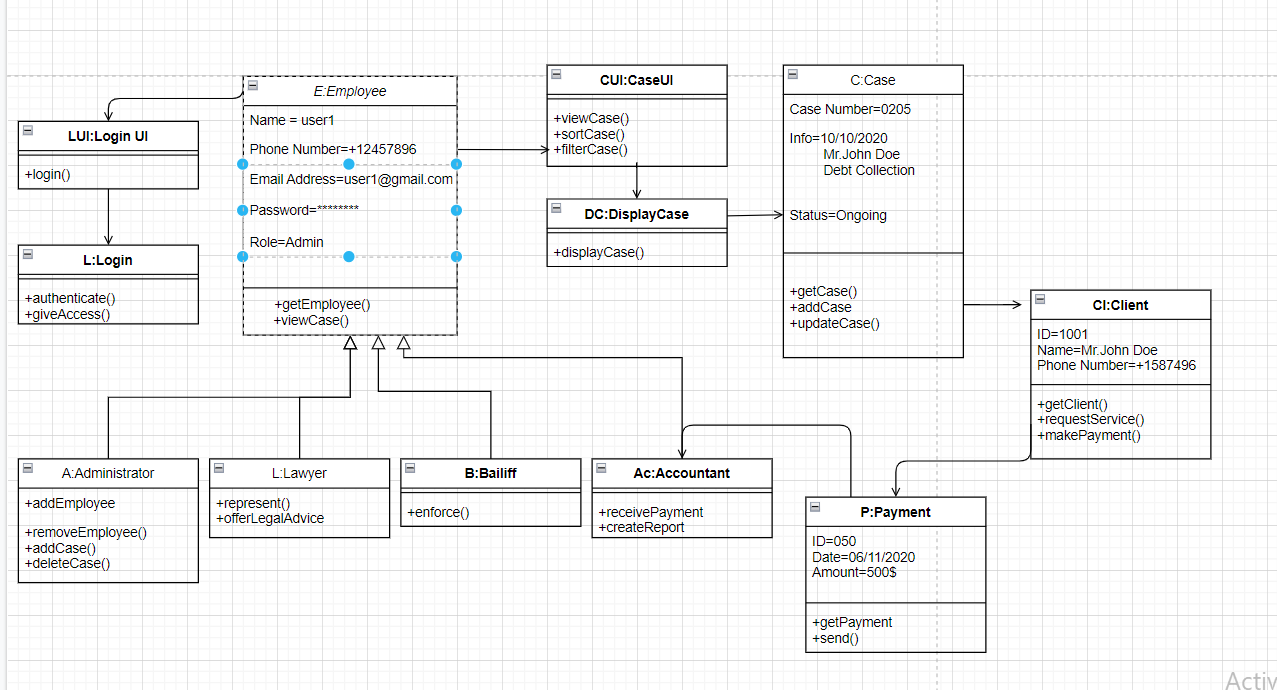
****

Lawyer Login

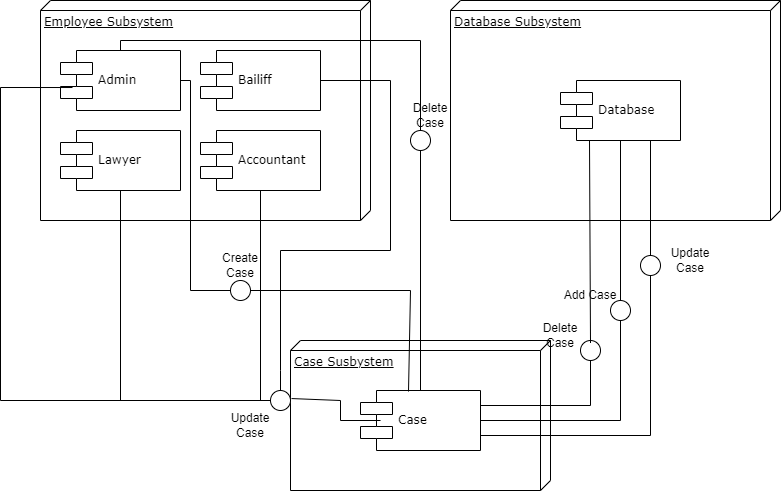
****

CLASS DIAGRAM

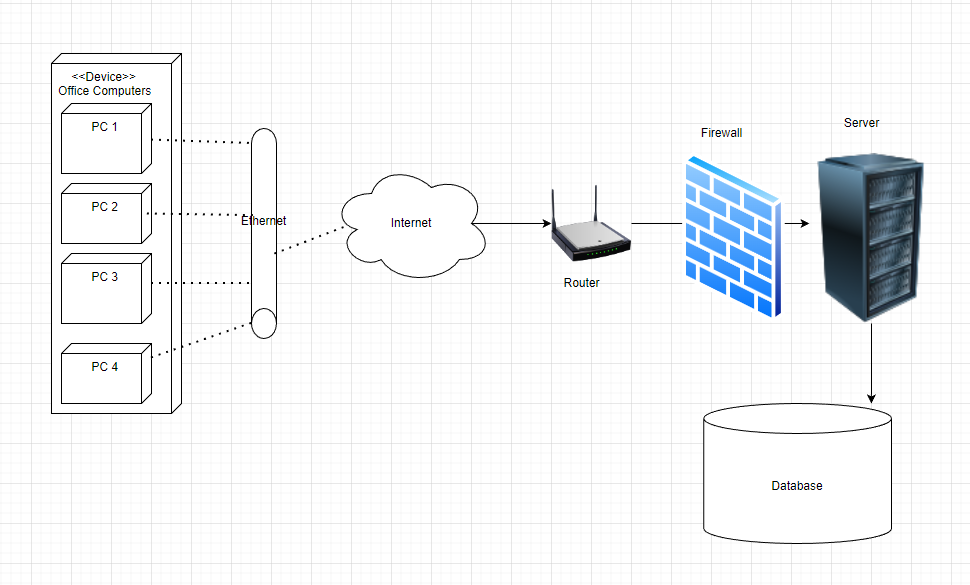


OBJECT DIAGRAM****

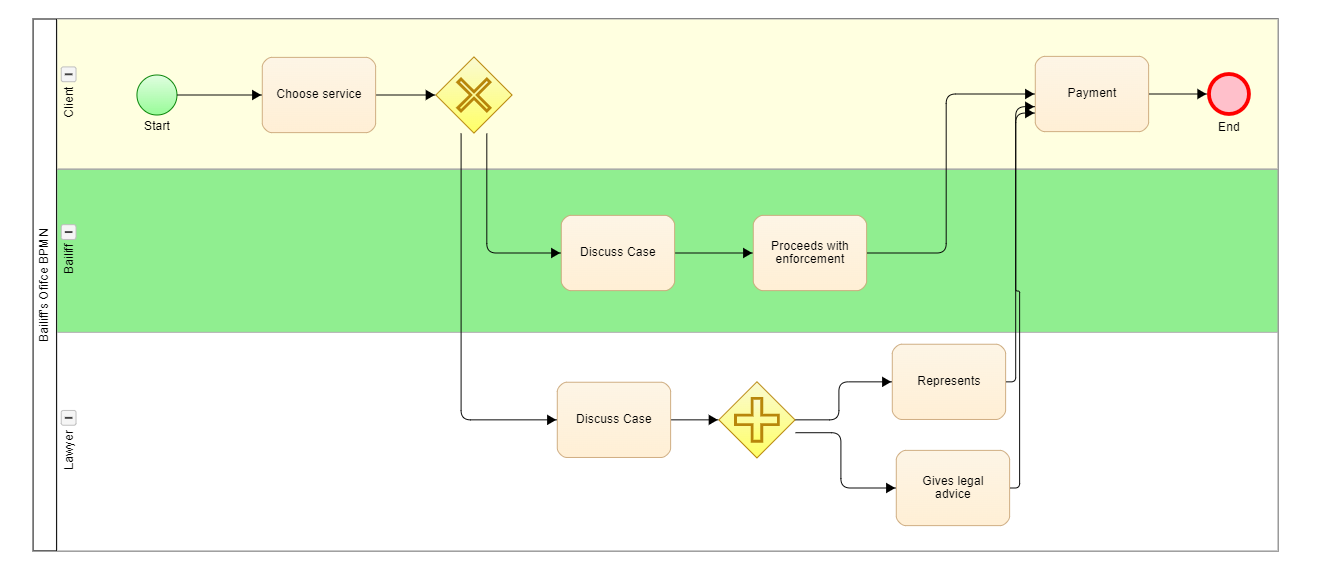
COMPONENT DIAGRAM

******

DEPLOYMENT DIAGRAM



BPMN

******

DESIGN PATTERNS

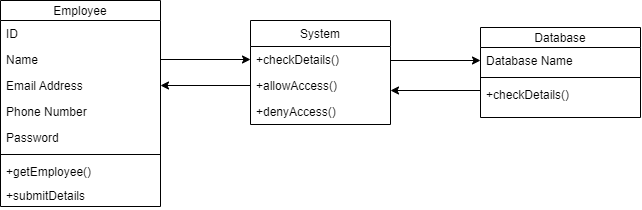
***Mediator***

The Mediator pattern is used to reduce communication complexity between multiple objects or classes. This pattern provides a mediator class which normally handles all the communications between different classes and supports easy maintenance of the code by loose coupling. Mediator pattern falls under behavioral pattern category.

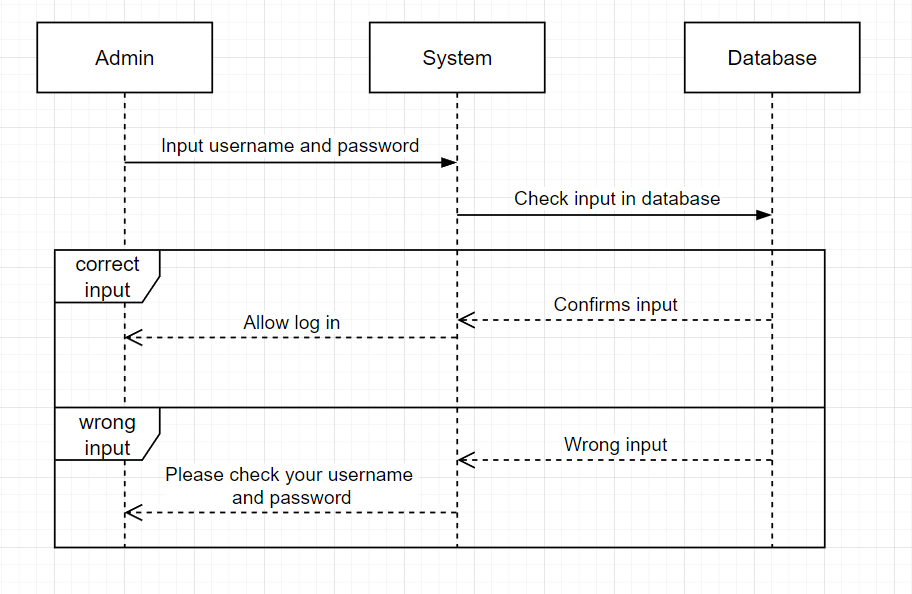
Application:

In this case, the system itself acts as the mediator between the user and the database. The system receives the input from the user and sends it to the database for checking. If the user’s credentials (username and password) are correct, then the user is given access. Otherwise, the user is denied access.

Implementation in Class Diagram:



Implementation in Sequence Diagram: (also applies to other Login sequences)



***Factory* *Method***

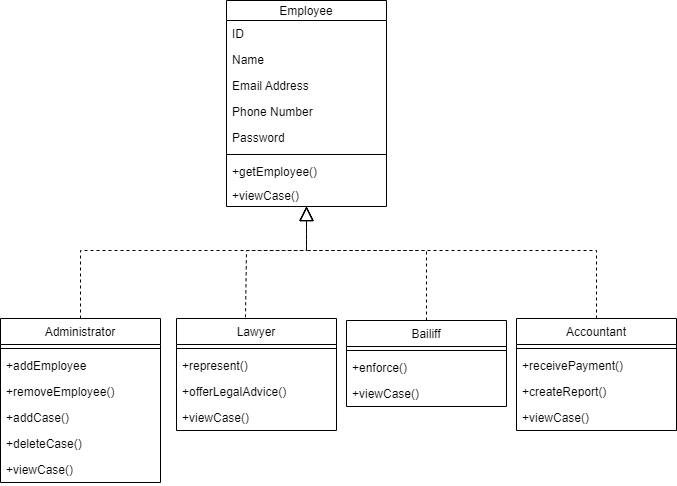
The Factory Method pattern is a creational pattern that uses factory methods to deal with the problem of creating objects without having to specify the exact class of the object that will be created. This is done by creating objects by calling a factory method—either specified in an interface and implemented by child classes, or implemented in a base class and optionally overridden by derived classes—rather than by calling a constructor.

Application:

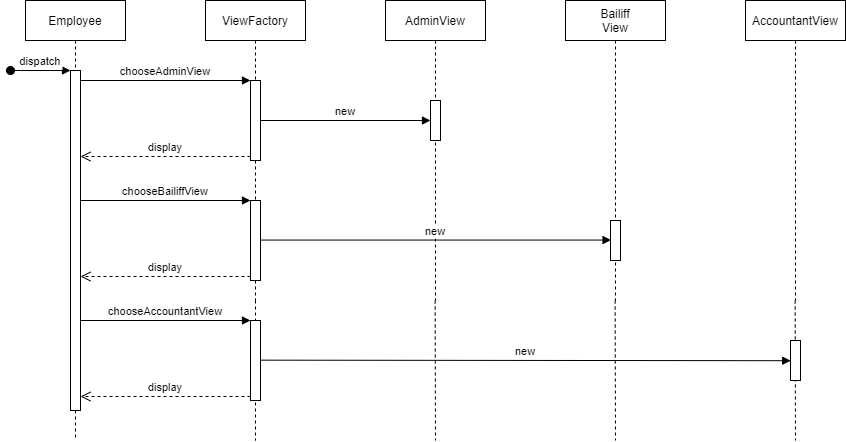
Different employees can all view the cases. But we may want to provide different views. For example, the accountant should not be able to view everything that the administrator can. This is a problem that can be solved with Design Patterns.

By using the Factory Method pattern, we create new subclasses that implement the same interface, but have a different implementation of the viewCase() method. This allows us to split the Employee class into subclasses, each of which can view different things by using the viewCase() method.

Factory Method implementation in Class Diagram:



Factory Method implementation in Sequence Diagram:



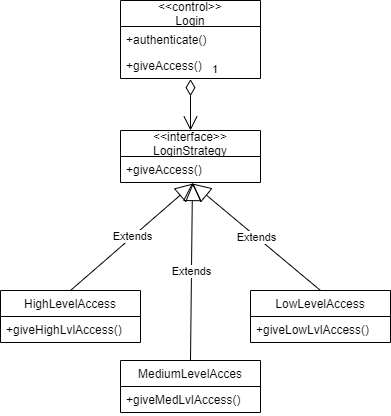
***Strategy***

The Strategy pattern is a behavioral software design pattern that enables selecting an algorithm at runtime. Instead of implementing a single algorithm directly, code receives run-time instructions as to which in a family of algorithms to use. Strategy lets the algorithm vary independently from clients that use it.

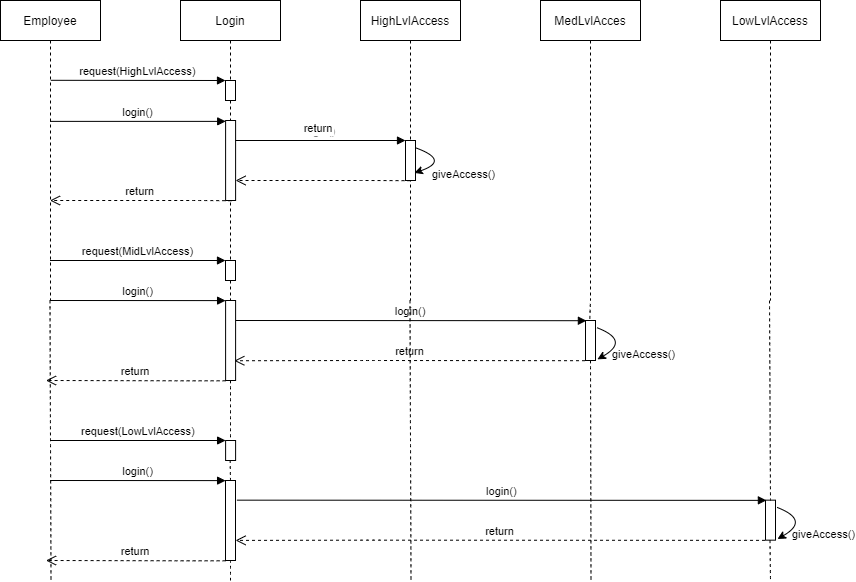
Application:

The Strategy pattern can be used in this project for dealing with user login. As different users will have different access levels, we can implement the Strategy pattern to split the login process into 3 separate algorithms, each of which gives the user a different access level. After the correct algorithm is chosen, only the needed one is executed at runtime.

Strategy implementation in Class Diagram:



Strategy implementation in Sequence Diagram:

******

***Architectural Choices***

As the Bailiff’s Office Software System will be a web application, we decided the most appropriate architectural pattern would be the client-server pattern.

In this pattern, there are two main components: the client, which is the service requester, and the server, which is the service provider. They communicate over the internet, on separate hardware. There is only one server, but many clients. The server is physically located in the Bailiff’s office, while the client refers to the web browser which the employees will use to access the web app.

The server perpetually listens for requests from the clients. When a request is received, the server processes it, and sends a response to the client. The client initiates interactions with the server to generate the services needed. Both components are linked by request/reply connectors.

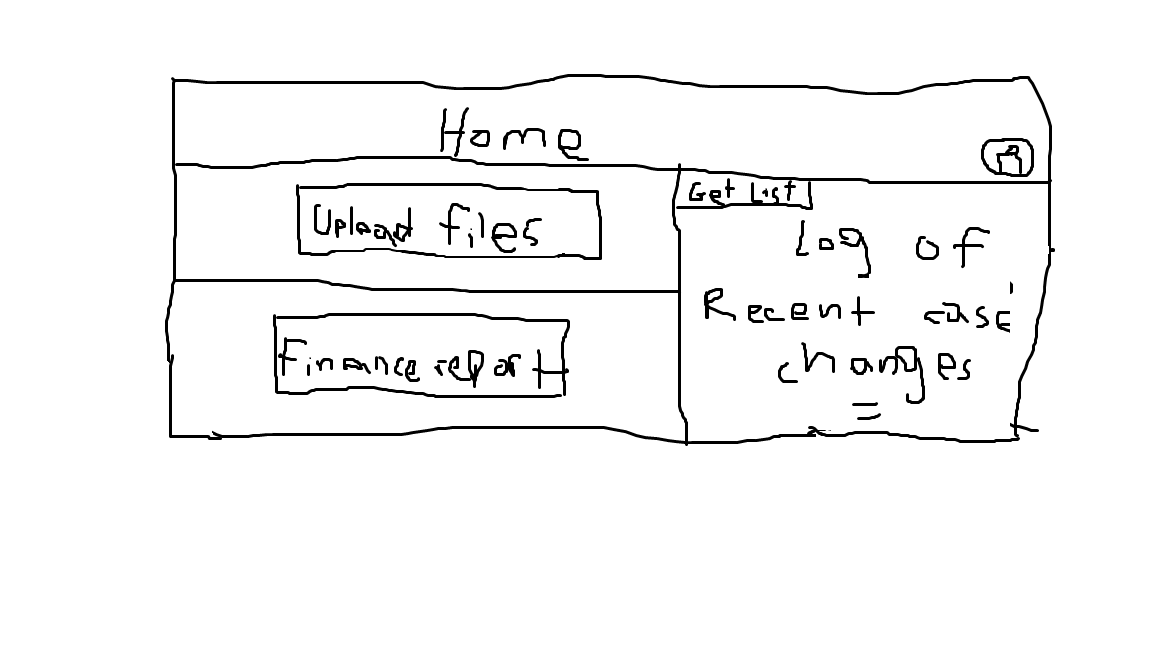
A major advantage of this pattern is the central computing of data. All the files are stored in a central location (the server), therefore the data is centrally controlled.

The client-side programming will be done using HyperText Markup Language (HTML). Additionally, Cascading Style Sheets (CSS) will be used to make the web app’s visual design more modern and aesthetically pleasing. JavaScript and the jQuery library will be used to make the web app more interactive for the users.

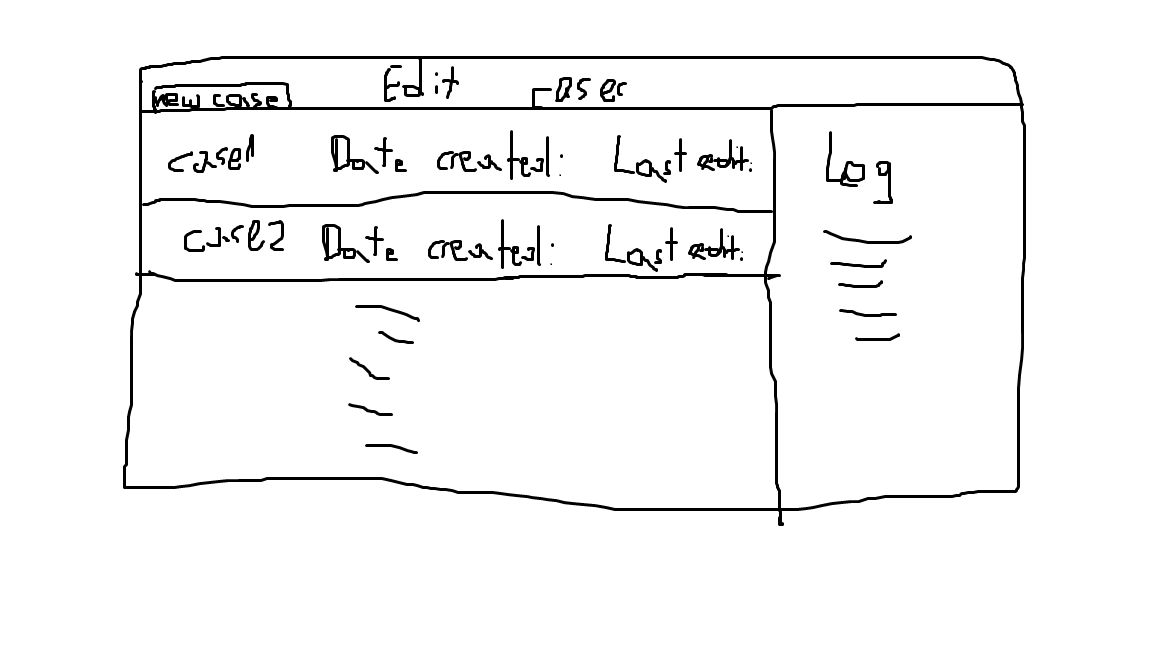
On the server side, MySQL database will be used to store data. PHP programming will be used for integrating the web app with the database

1. APPENDIX

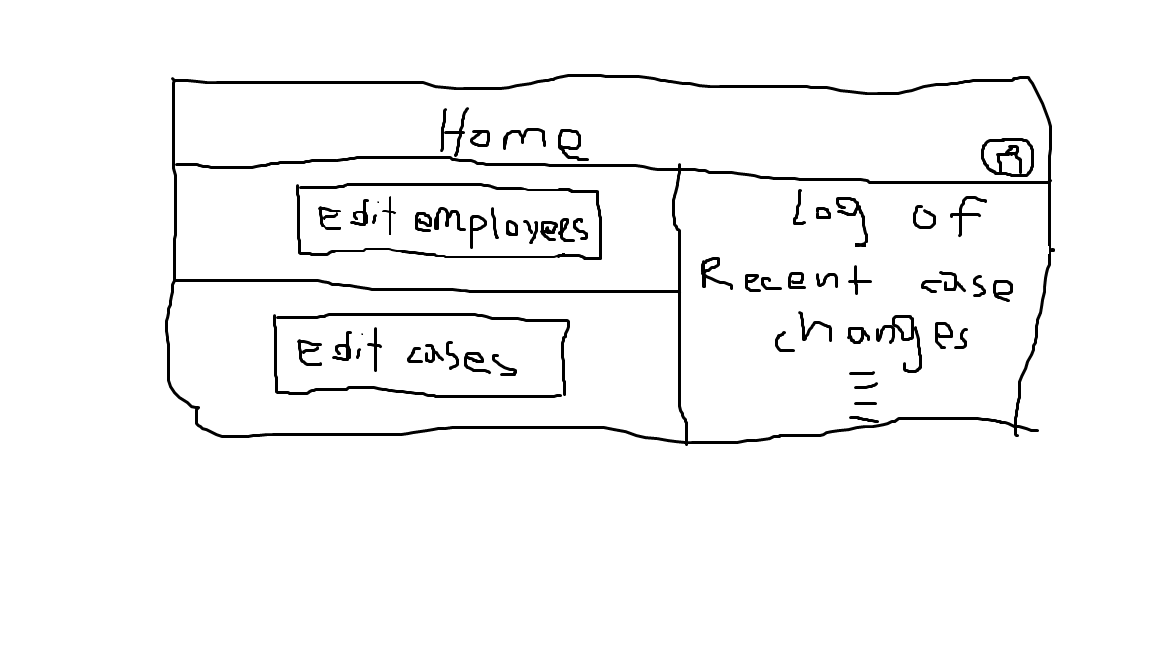
Accountant Home Page Interface



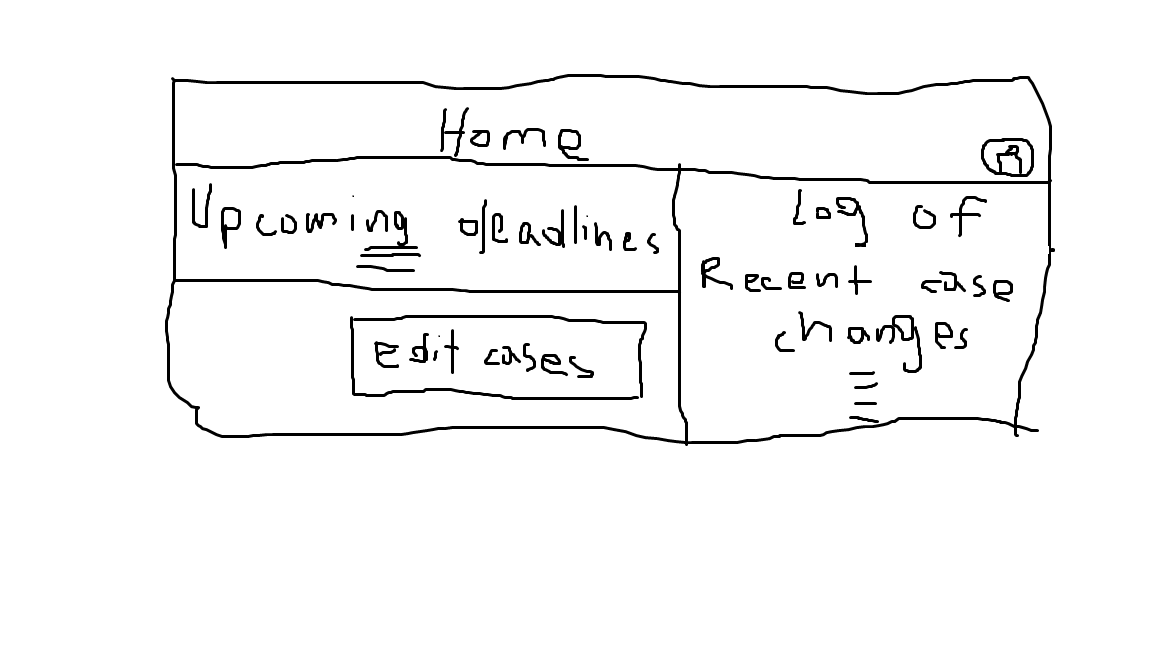
Admin Edit Case Interface



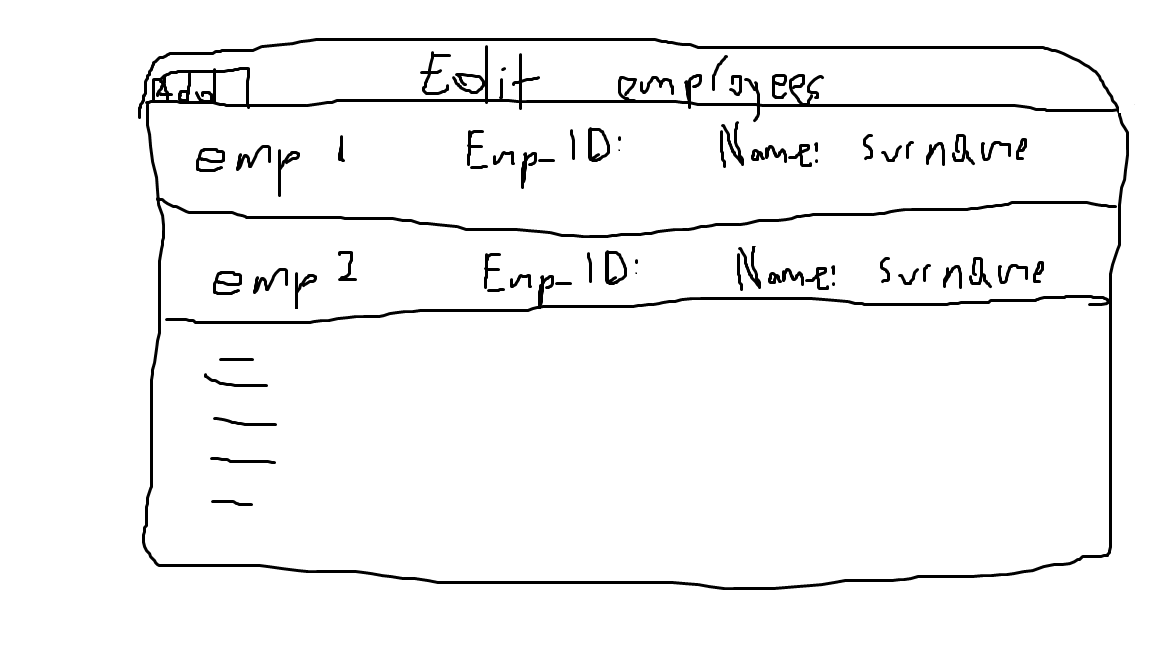
Admin Home Page Interface



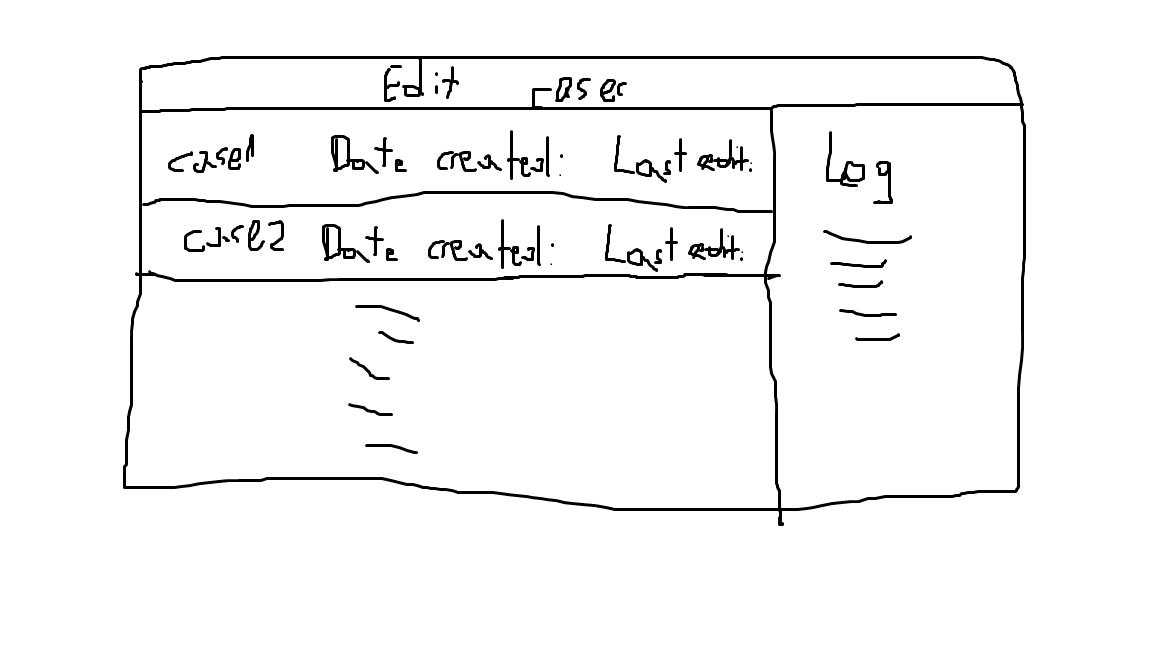
Bailiff & Lawyer Home Page Interface



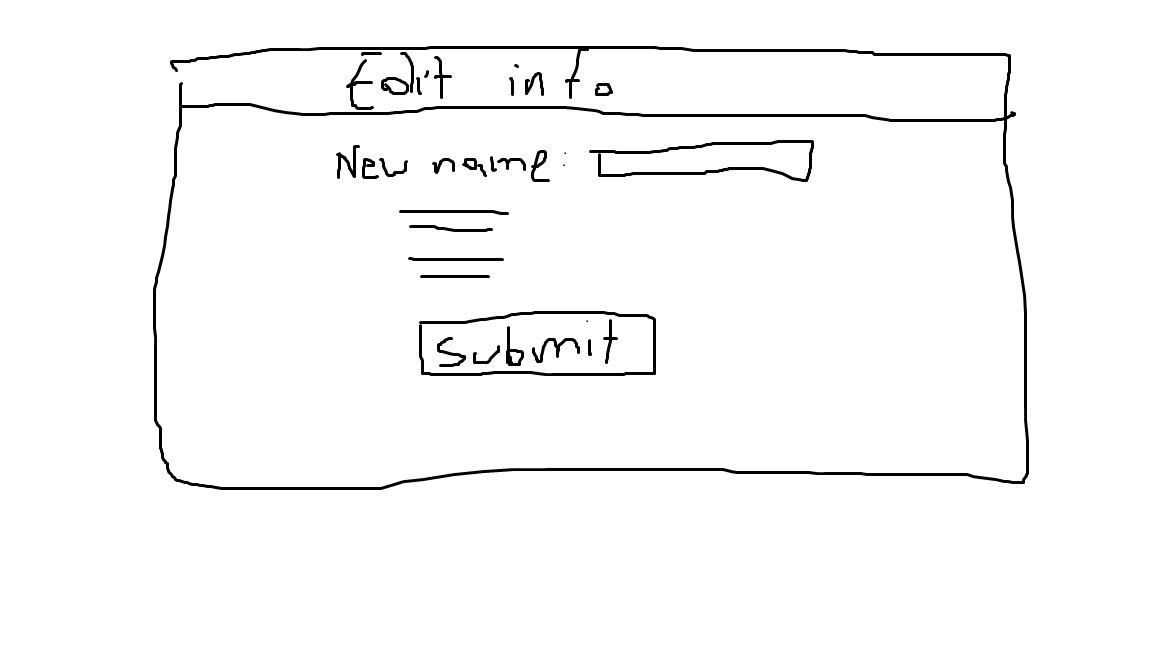
Edit Employees Interface



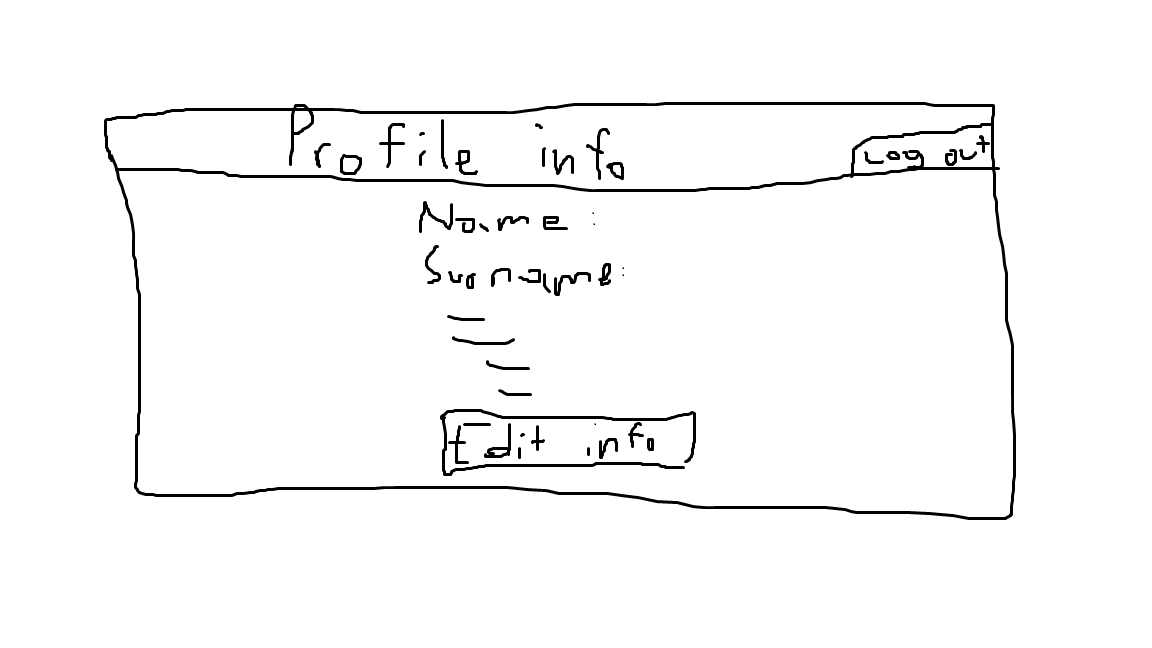
Employee Edit Case Interface



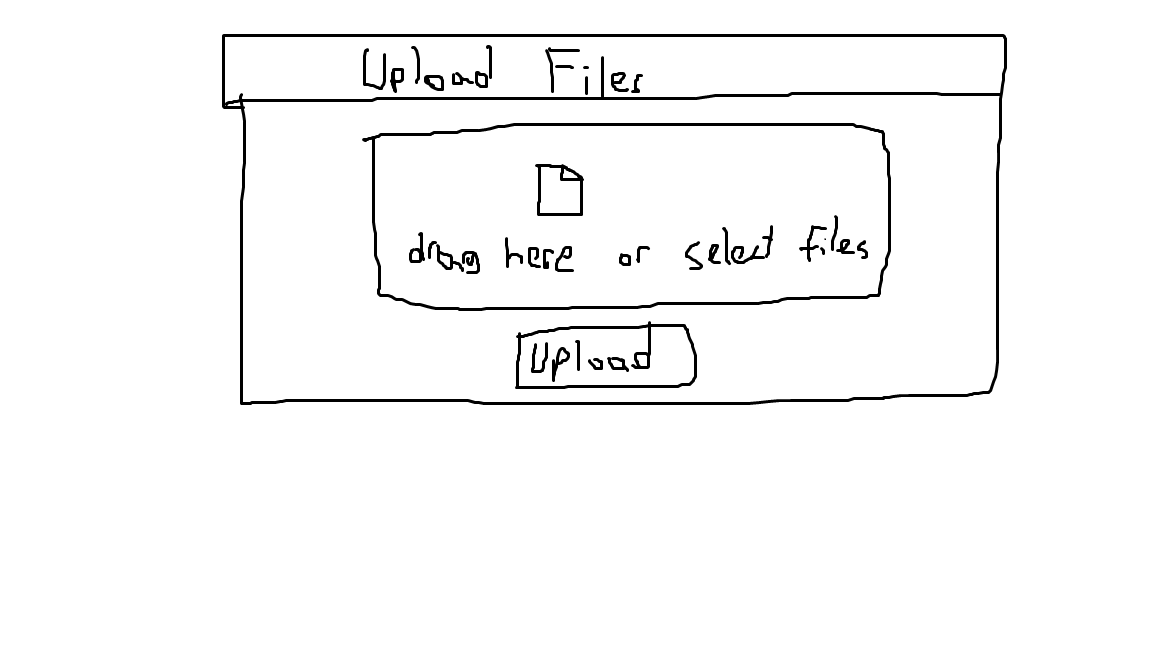
Employee Edit Info Interface



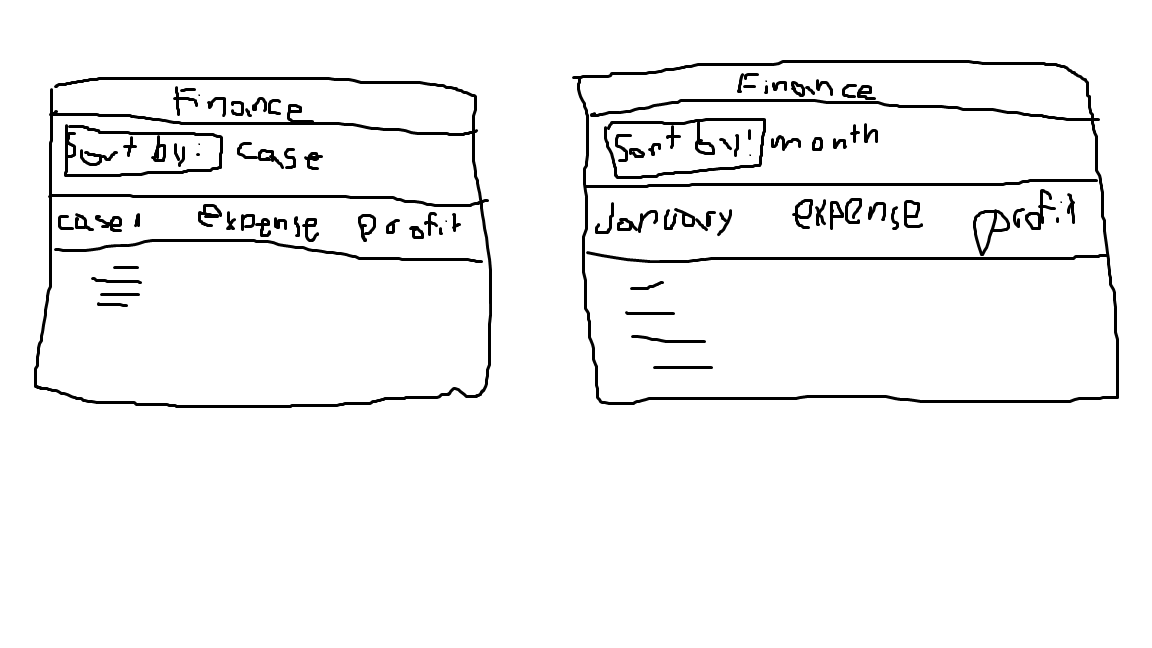
Employee Profile Page Interface

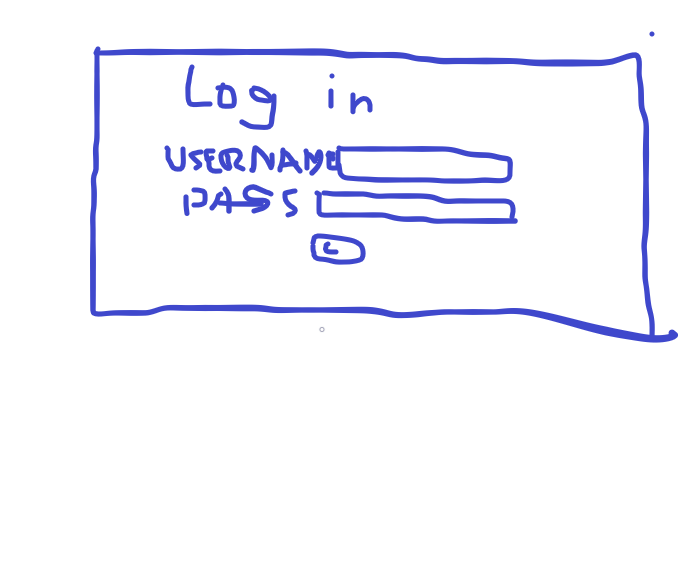


Employee Upload Files Interface



Finance Report Interface





Login Interface