

HUMAN RESOURCE MANAGEMENT SYSTEM



Supervisor:

Mr. Avay Kafle

Birendra Memorial College

A BSc.CSIT Final Year Project

Birendra Memorial College

Institute of Science and Technology

Dharan, Nepal

By:

Aastha Bajracharya (21420/075)

Kiran Nepal (21427/075)

Sandesh Adhikari (21437/075)

ABSTRACT:

Human Resource Management System (HRMS) is a software suite that provides a complete solution for managing HR processes throughout an employee's career. This project aims to design and develop an HRMS for a fictitious organization that can be used to manage employee data, attendance, payroll, and performance analysis. The system is designed to be web-based, allowing employees to access their information from anywhere with an internet connection. The project utilizes a range of technologies, including Flutter, Python, Django, MySQL, and APIs..

OUTLINES:



Introduction



Background Study



System Analysis



System Design



Implementation and Testing



Conclusion and Future
Recommendations

INTRODUCTION:

An HRMS is a suite of software programs to manage human resources and associated procedures into one package.

Some key points to describe our project:



Centralized Organizational platform



Automated Attendance using Face Recognition Technology



Focused on improving the productivity and efficiency of an organization



PROBLEM STATEMENT:



Absence of proper HRM software, create huge effect on managing resources in effective way.



Absence of work discipline and punctuality may lead to shrink in employee performance.



Flaws in the data organization can create huge loss for the organization.

OBJECTIVES:



To help the organization to mark the attendance of the employee automatically by keeping records of timestamps when the employee enters the premise with the use of Face Recognition Technology.



To help businesses manage and store employee data in a centralized location, making it easier to access and analyze data for reporting and decision-making purposes.

SCOPE AND LIMITATION:

Scope

To develop system that automates HR process

Provide Centralized Database

Track and Access Employee performance

Limitation

Not able to detect attendance beyond the premises

System cannot be scalable to other business with different requirements

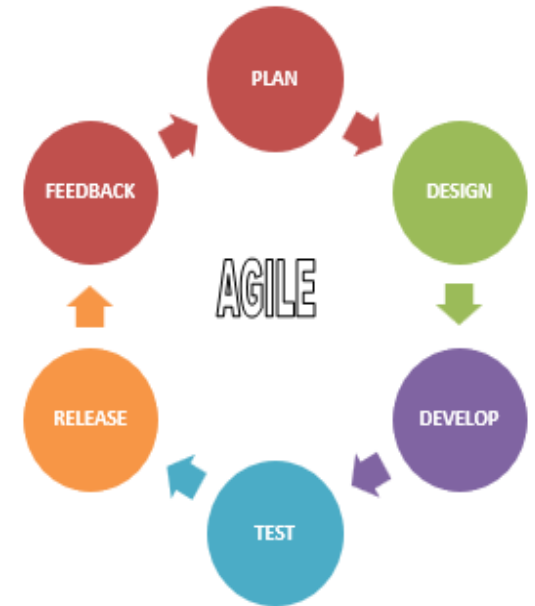
May not be able to covers all the HR functions.

METHODOLOGY:

The project was developed by following Agile Software Development Approach.

As, it is flexible and iterative approach that focuses on customer satisfaction and allows for continuous improvement throughout the project lifecycle.

Agile Software Development approach consists of six phases: plan, design, develop, test, release, and feedback.



BACKGROUND STUDY:

The HRMS project is developing a system that manages employee data and attendance using face recognition technology. HRMS systems play a critical role in managing an organization's workforce by ensuring the right people are in the right roles.

Face recognition technology matches an individual's facial features with a database of known faces, allowing for automated attendance.

Agile methodology is used to provide flexibility, collaboration, and continuous improvement. Implementing a system that uses biometric data requires careful consideration of ethical and legal implications, as organizations must comply with regulations to protect the privacy and security of employees' data.

SYSTEM ANALYSIS:

Systems analysis was done for this HRMS project to study the system, identifying requirements, solving problems, identifying opportunities.

It includes:



Requirement Analysis



Feasibility Analysis

REQUIREMENT ANALYSIS:

Functional Requirements

- Registration
- Login
- Attendance
- Designation
- Face Recognition

Non- Functional Requirements

- Accessibility
- Maintainability
- Scalability
- Portability

KEY CONSIDERATIONS IN FEASIBILITY STUDY:



Technical: System used Flutter, Python, Django, as programming languages and MYSQL as a database, which are readily available with extensive development support.



Operational: System is user-friendly and easy to adopt. Therefore, the organization will easily accept the system as the system is an effective and efficient solution for managing all HR tasks.



Economic: System is economically feasible as all the technologies used were already available



Schedule: System was developed with in the certain allocated time period.

ANALYSIS:

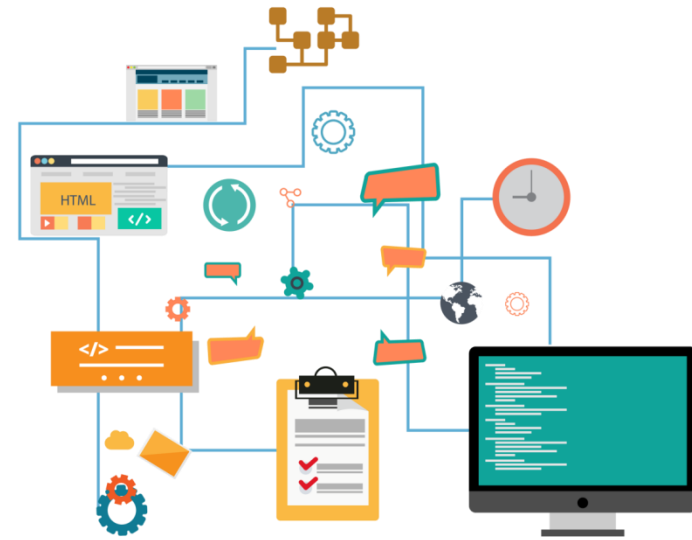
Object-oriented analysis is utilized in the HRMS project, as it offers a better understanding of the system's real-world objects and relationships, resulting in efficient software solutions.

The approach allows the identification of the system's objects, attributes, and relationships, leading to better comprehension of the system's behaviour.

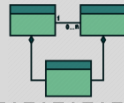
The object-oriented approach is ideal for long-term projects such as HRMS, where scalability, maintainability, and extensibility are critical requirements

SYSTEM DESIGN:

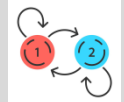
The system specifications from the analysis phase are transformed into a more detailed and structured representation of the system during the design phase.



MAIN DELIVERABLES OF SYSTEM DESIGN:



Class and Object Diagram



State Diagram



Sequence Diagrams



Activity Diagram

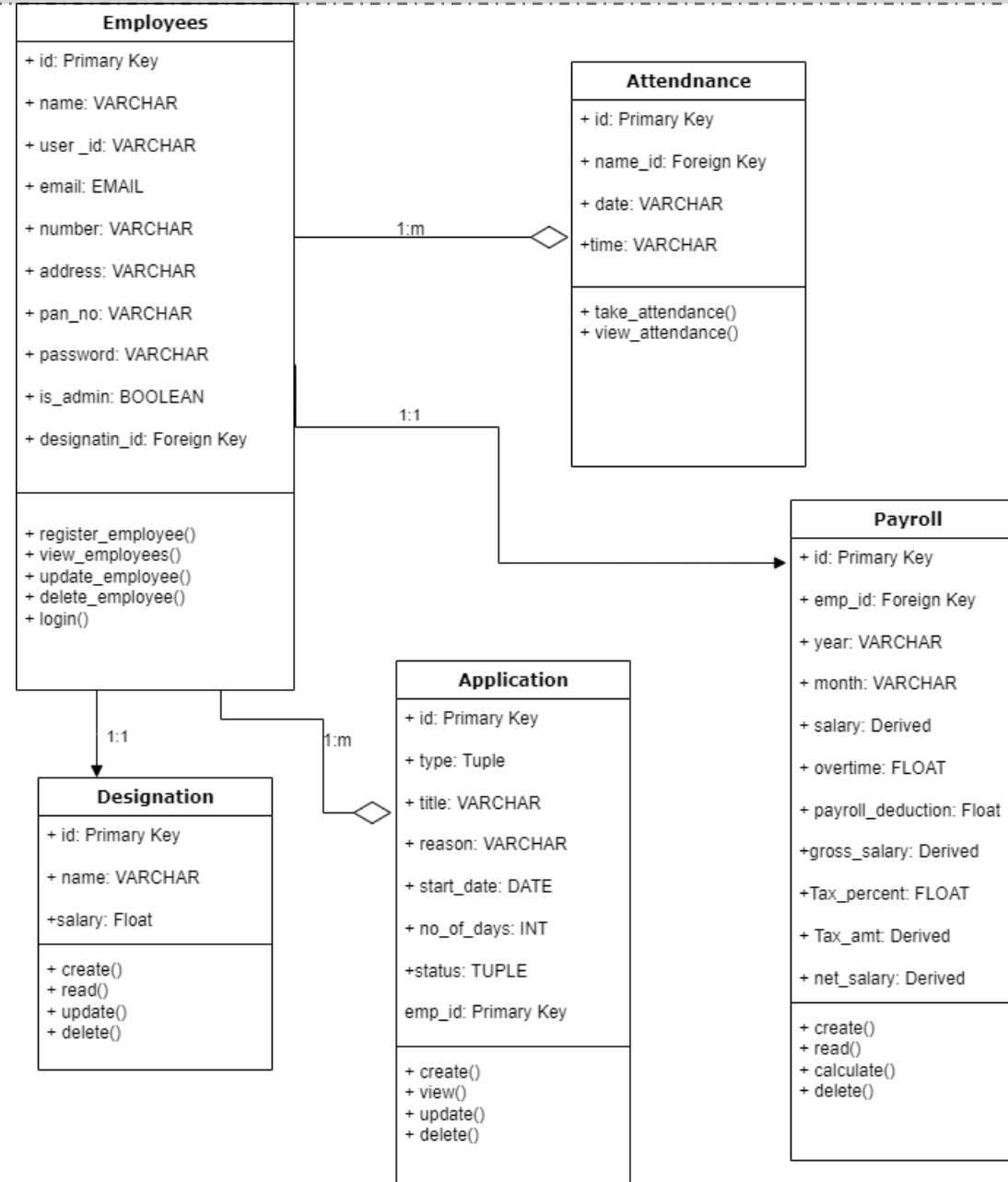


Component Diagram

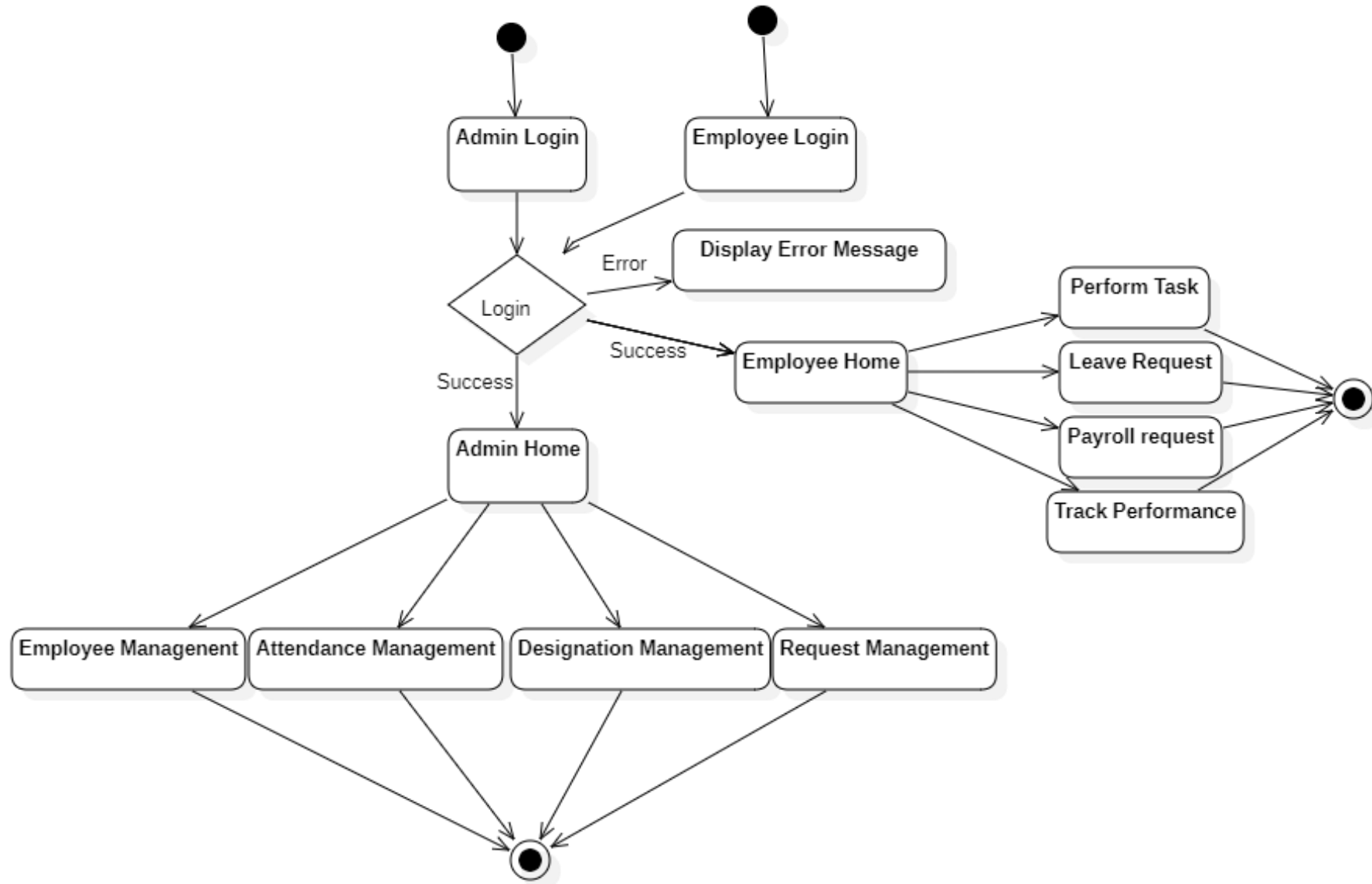


Deployment Diagram

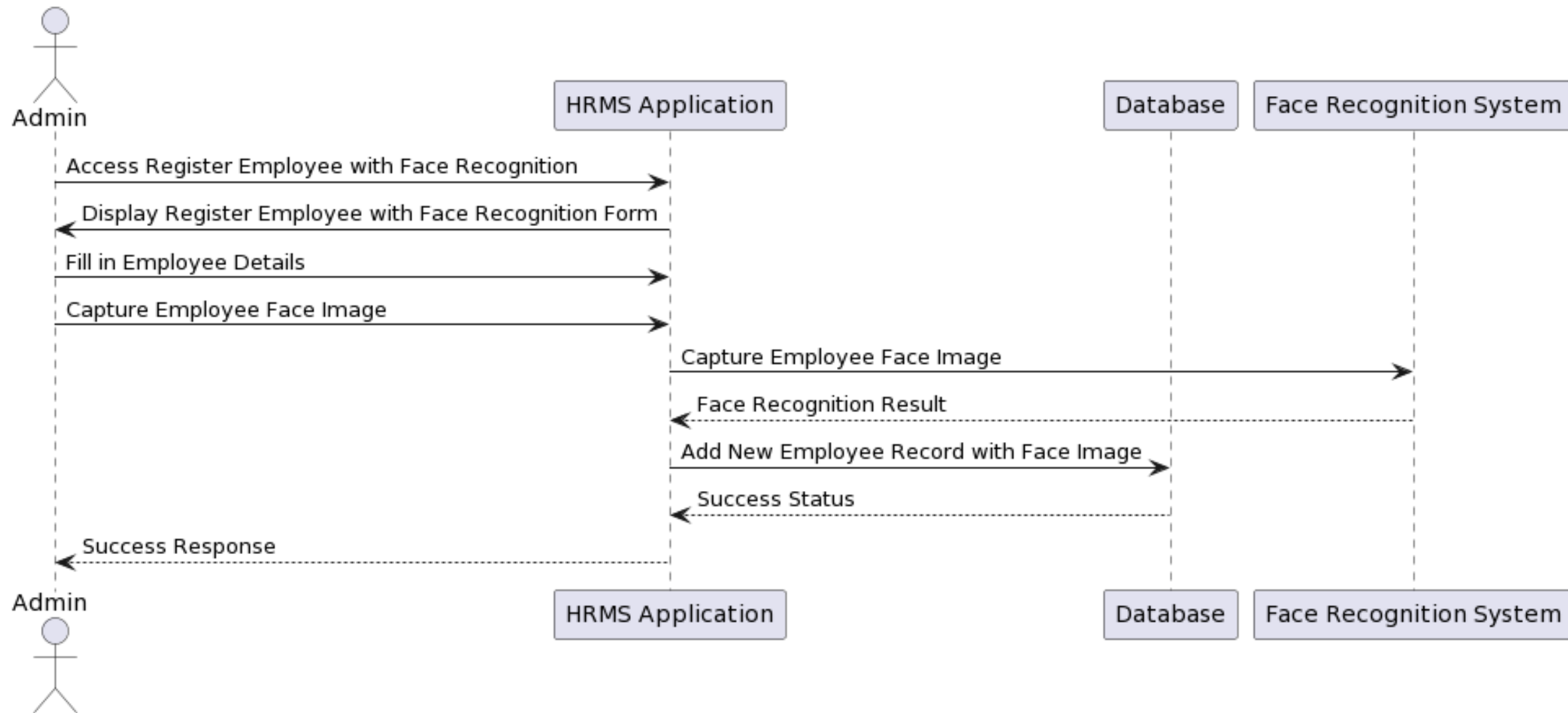
HRMS Class Diagram



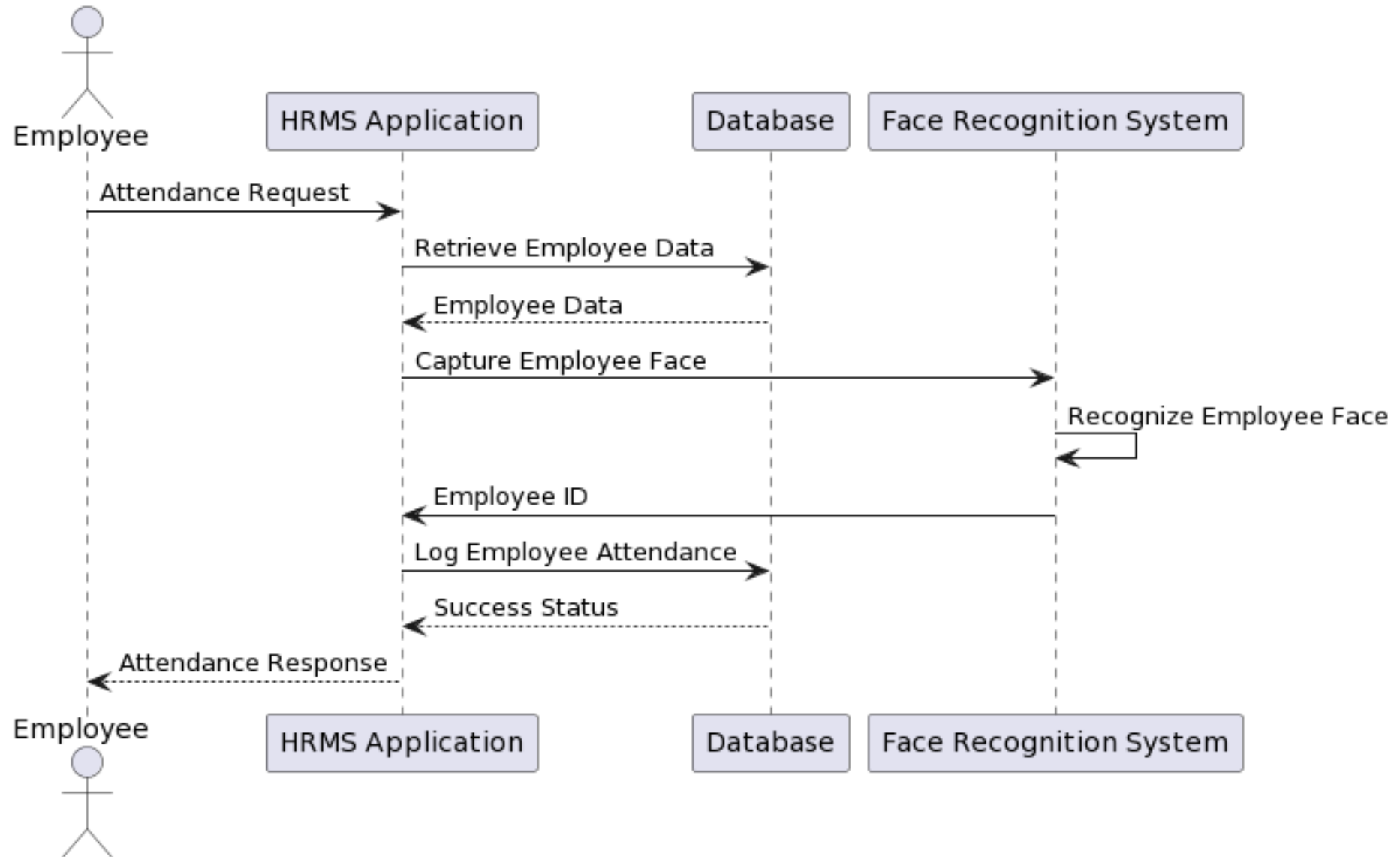
HRMS State Diagram



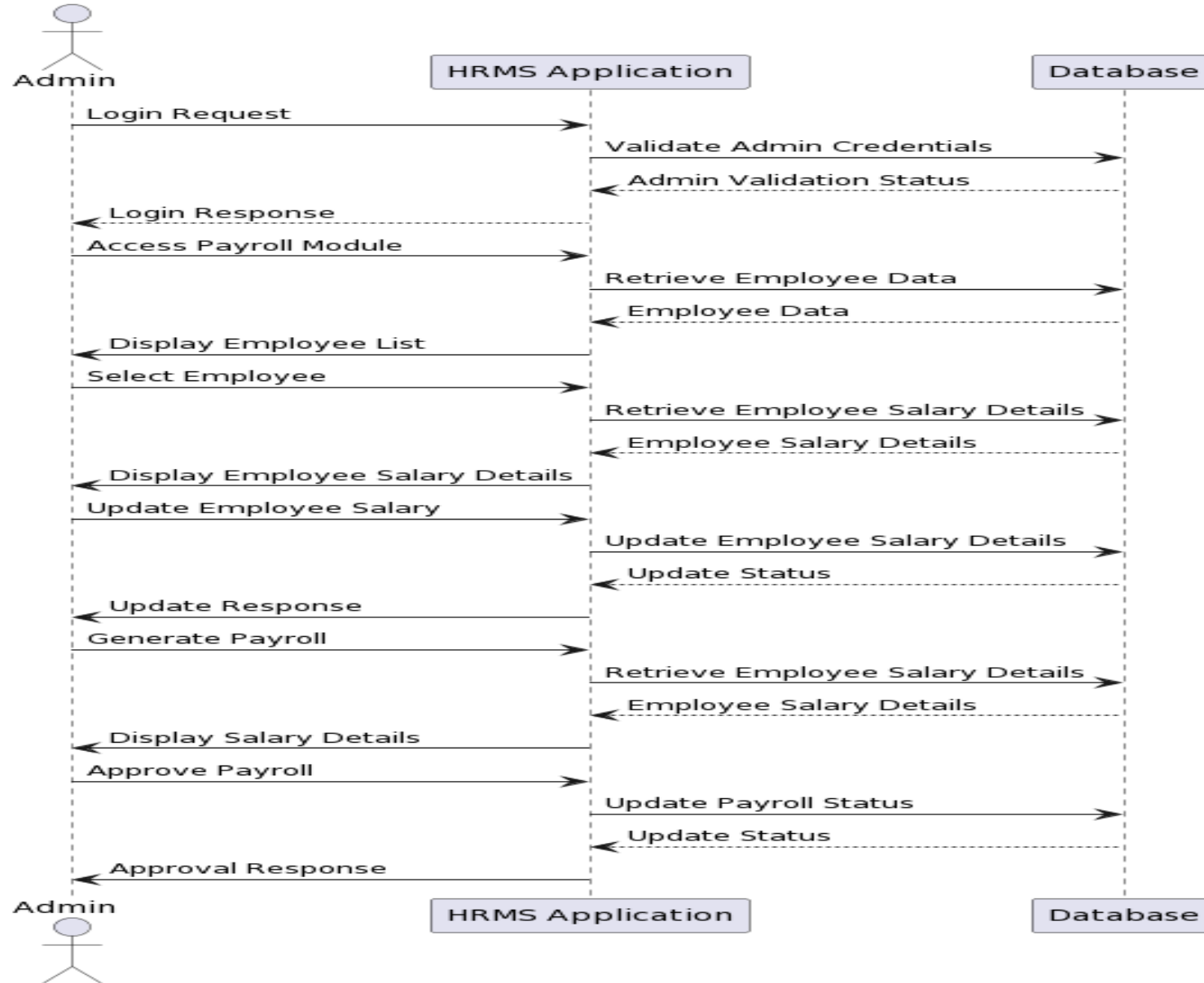
Employee Registration with Face Recognition Sequence Diagram



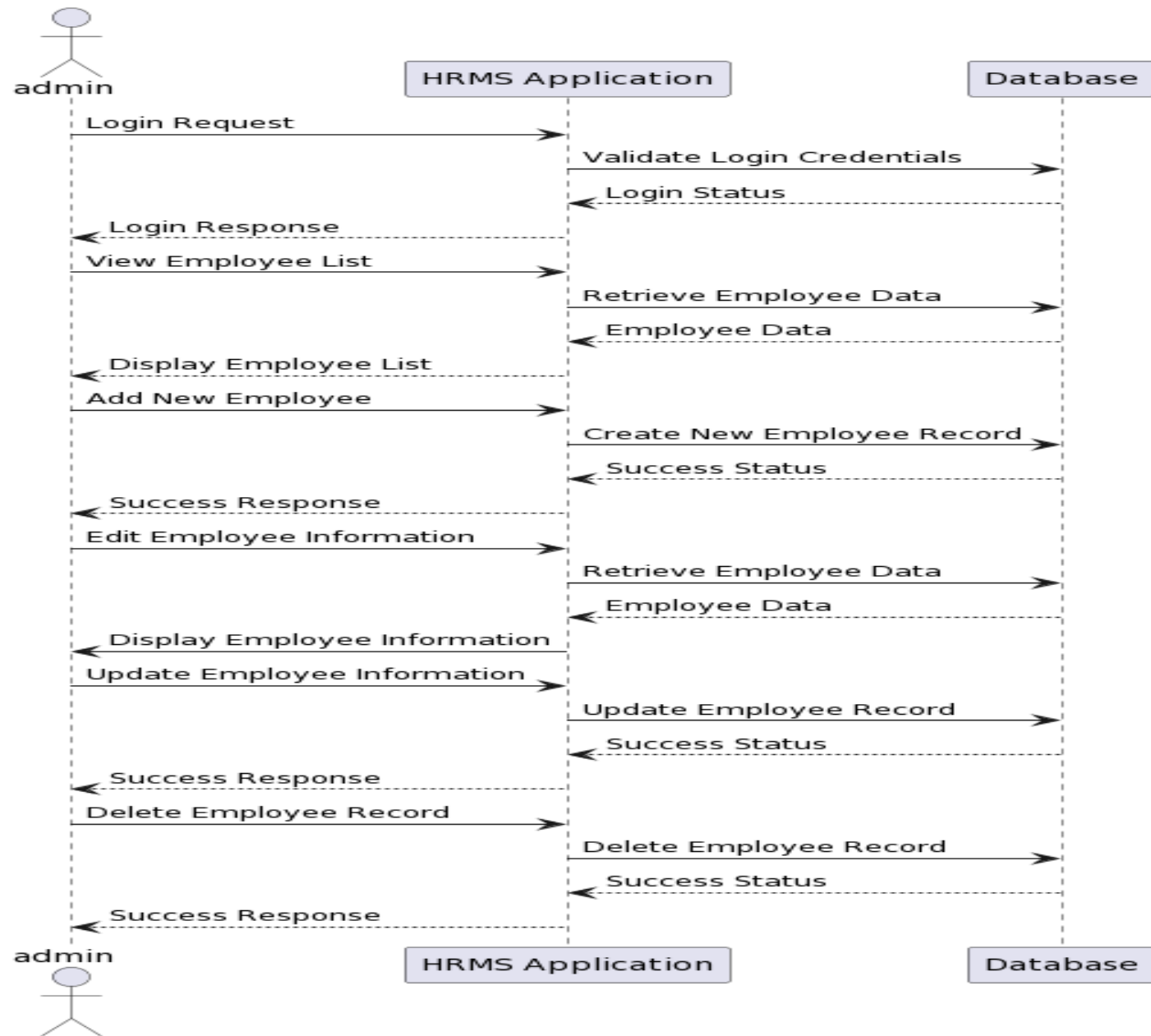
Attendance with Face Recognition Sequence Diagram



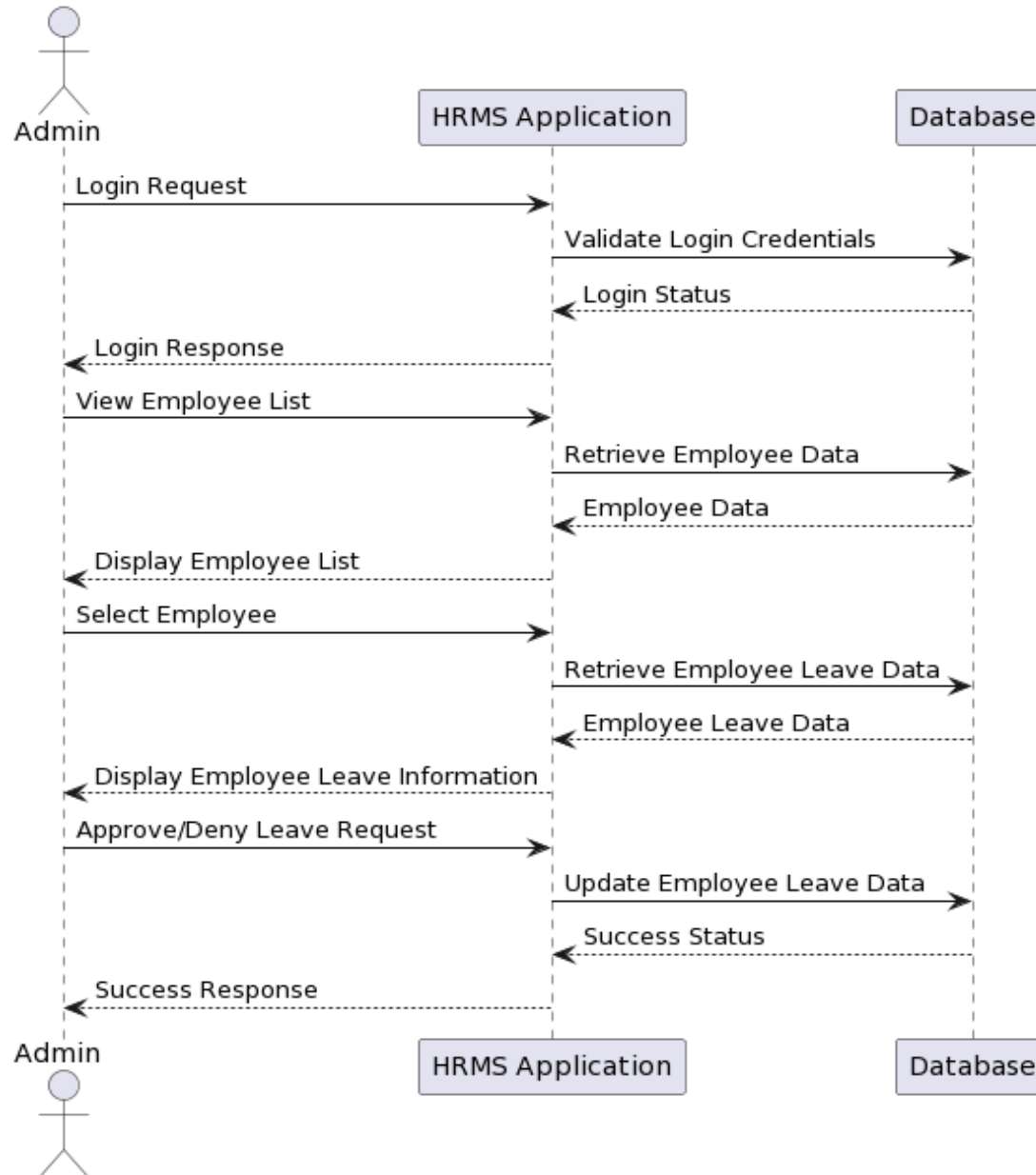
Admin Payroll Management Sequence Diagram



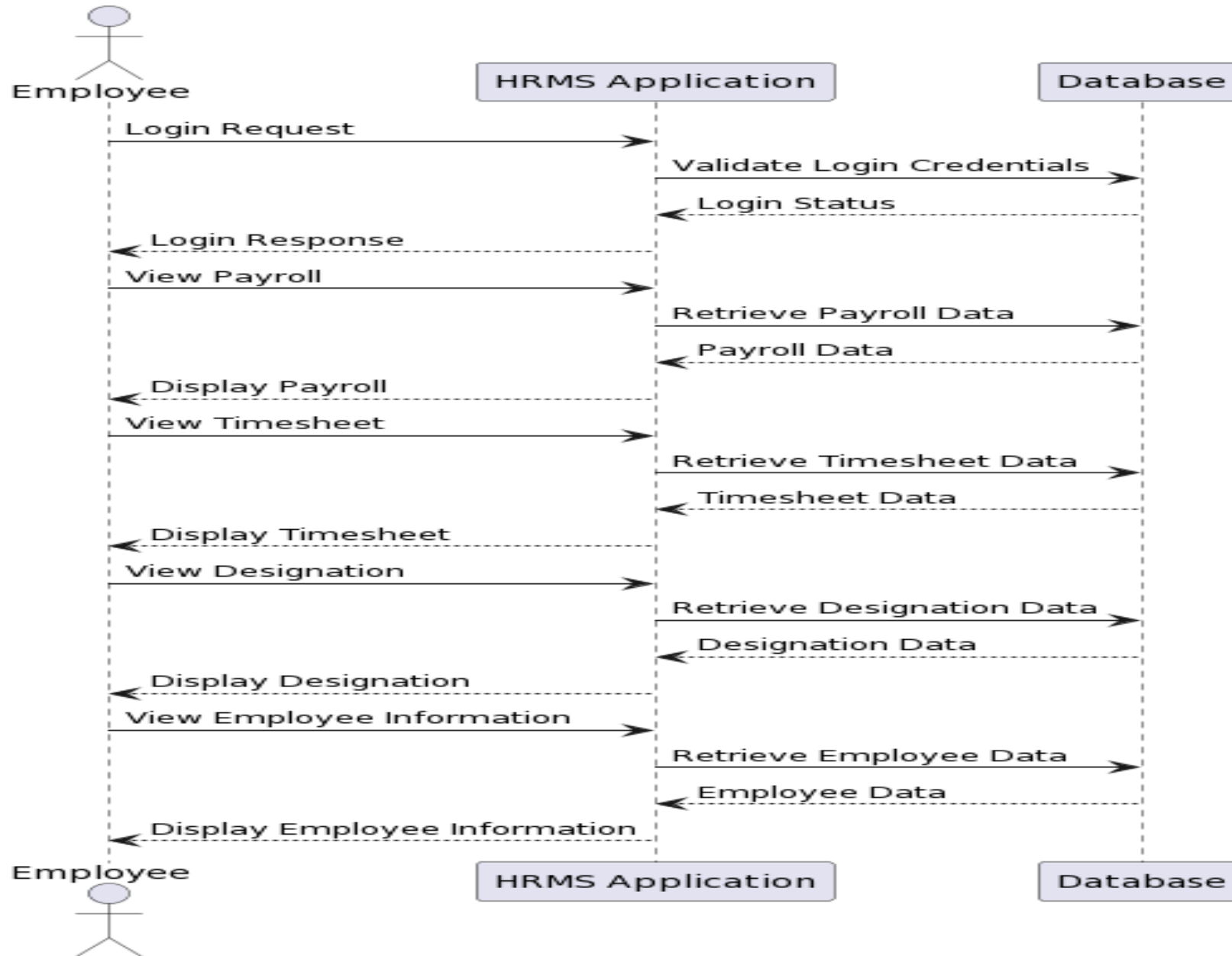
Admin Employee Management Sequence Diagram



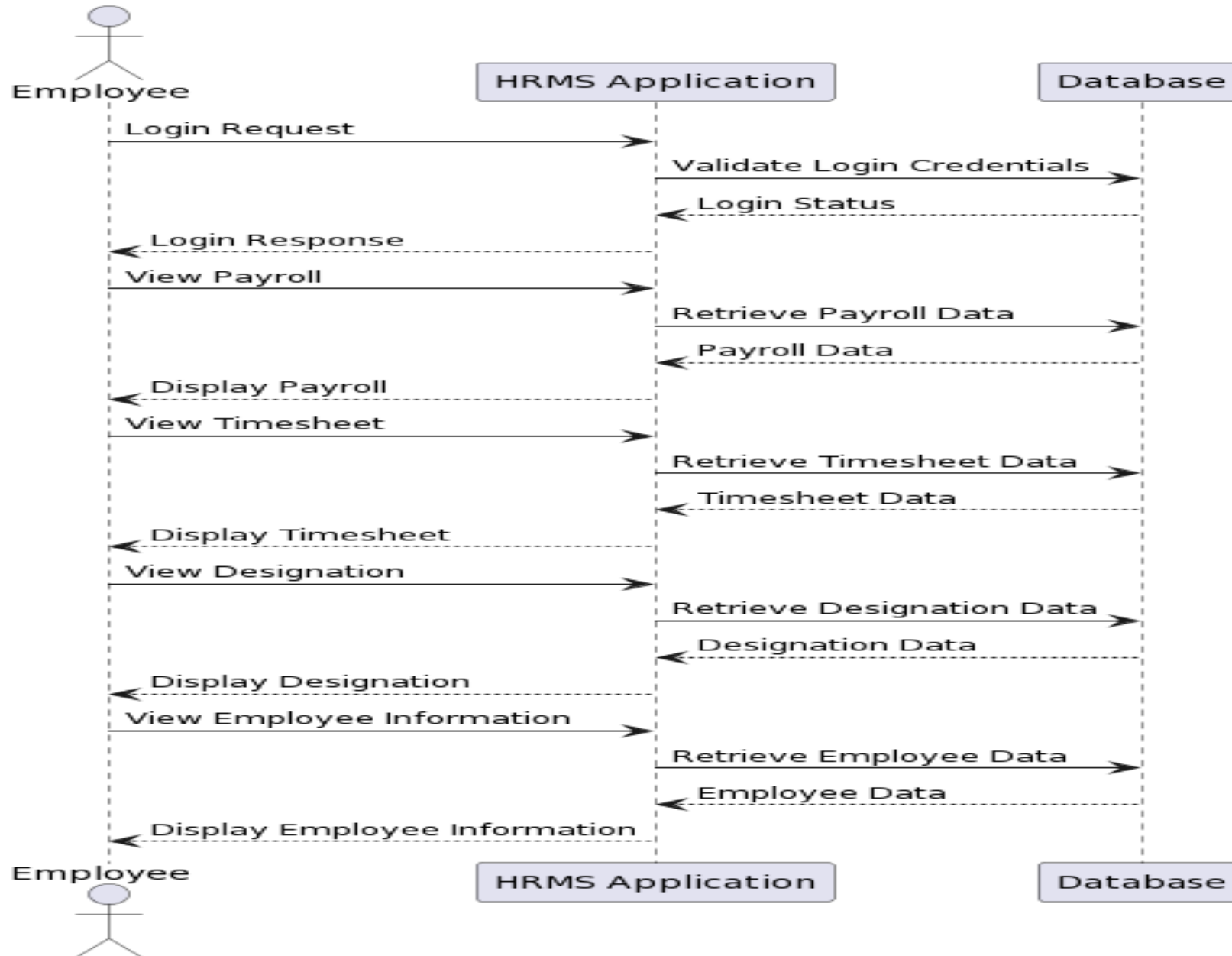
Admin Leave Management Sequence Diagram



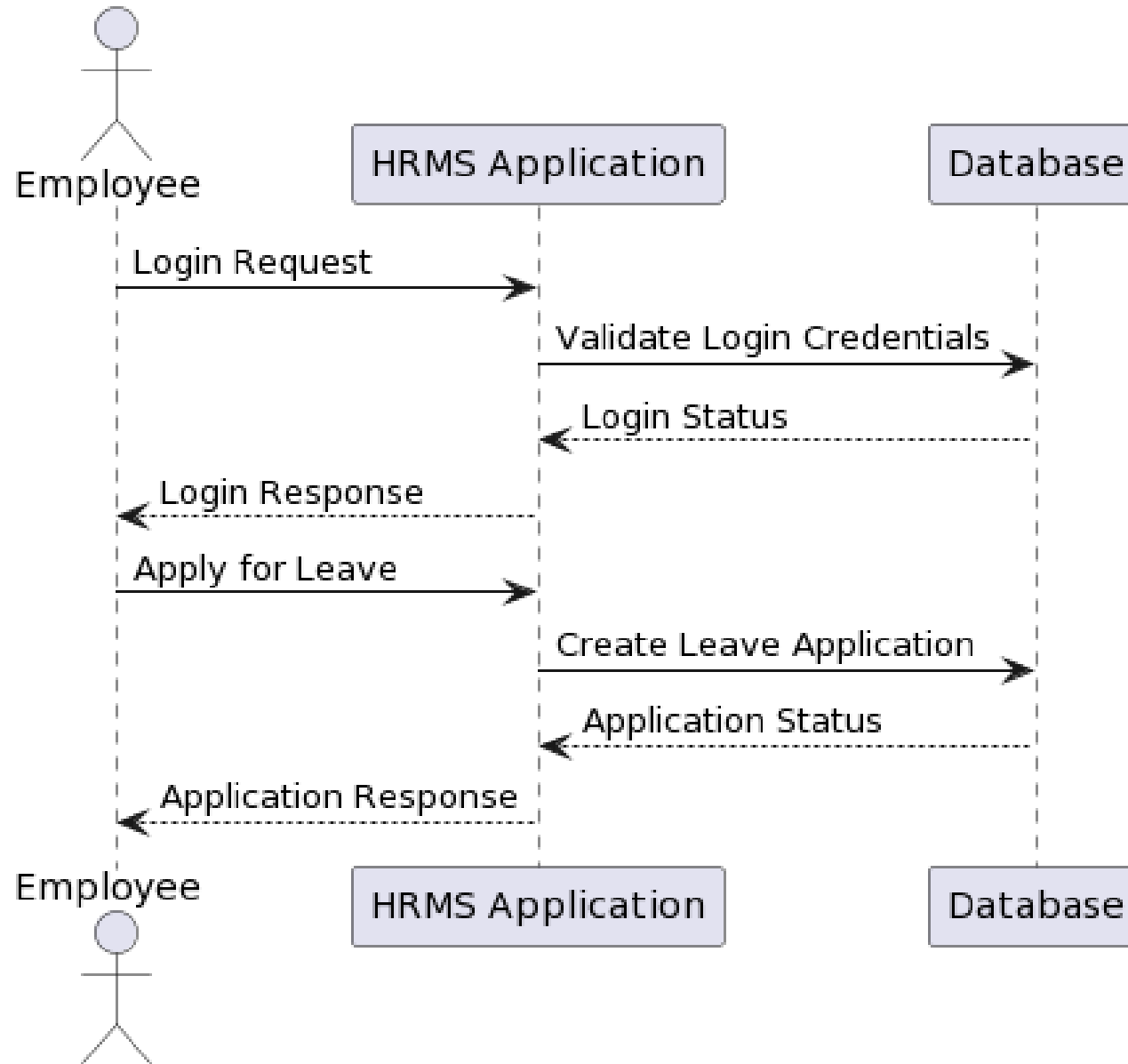
Employee View Payroll, Timesheet, Designation and Employee Information Sequence Diagram



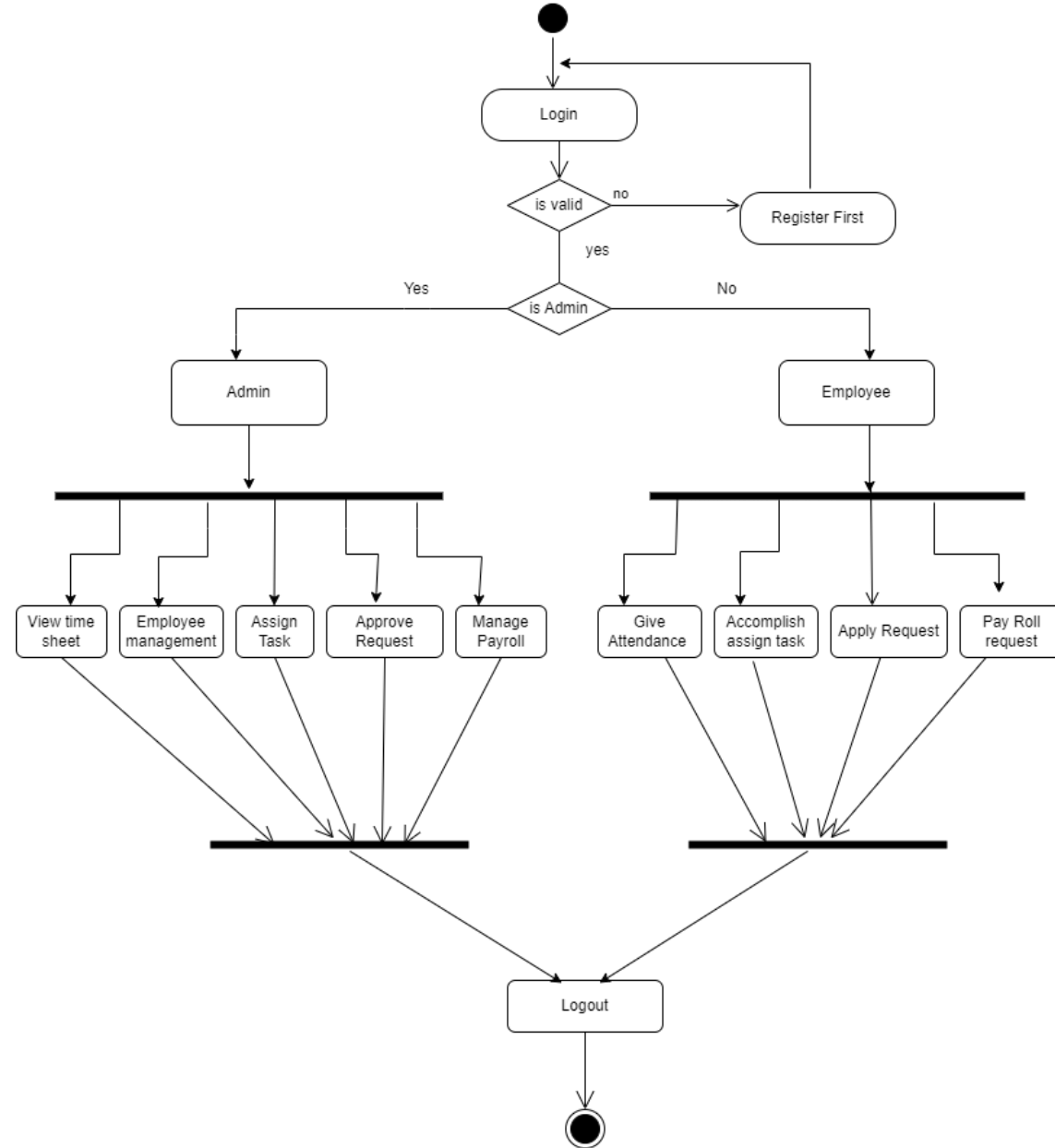
Employee Apply for Leave Sequence Diagram



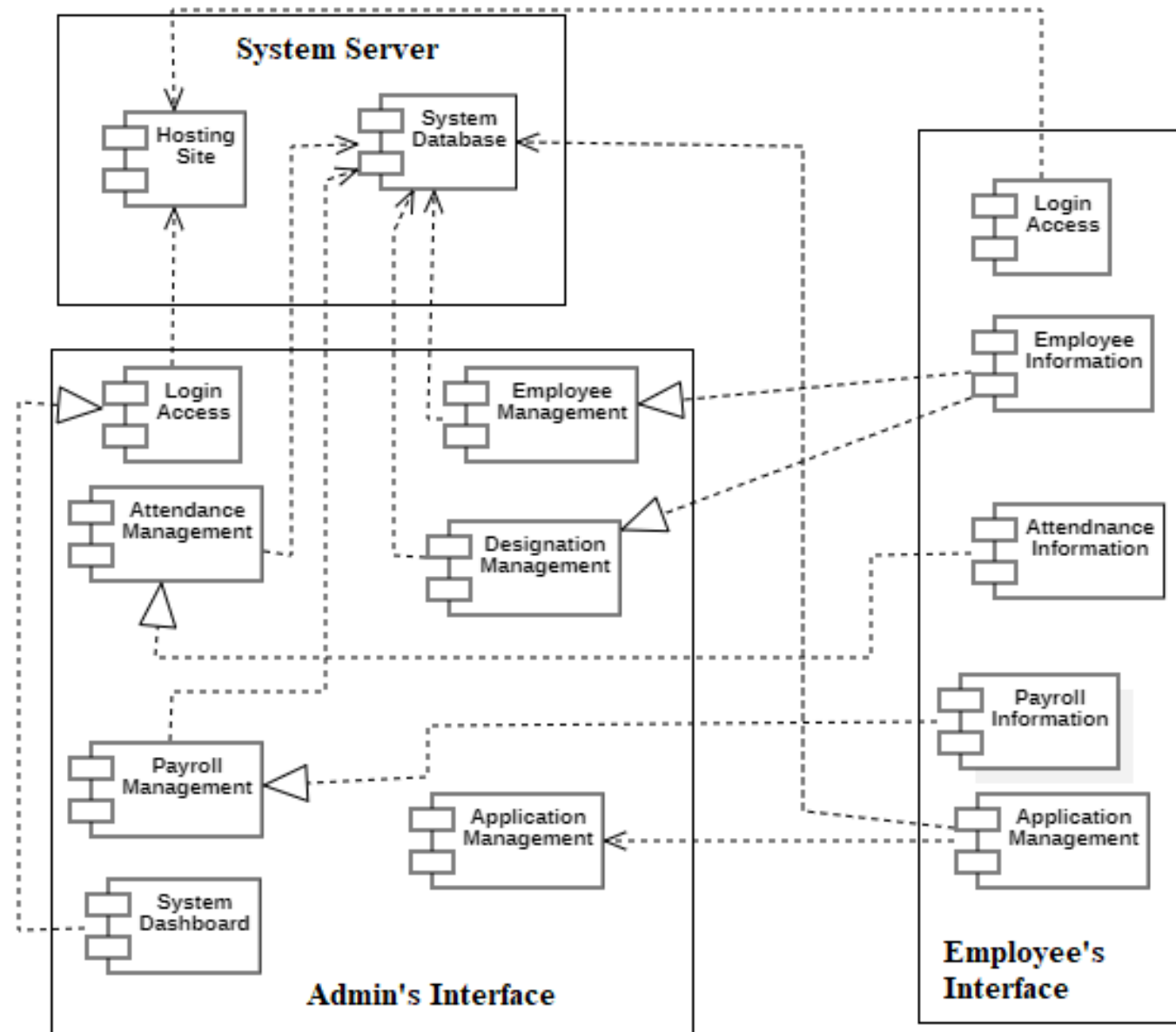
Employee Apply for Leave Sequence Diagram



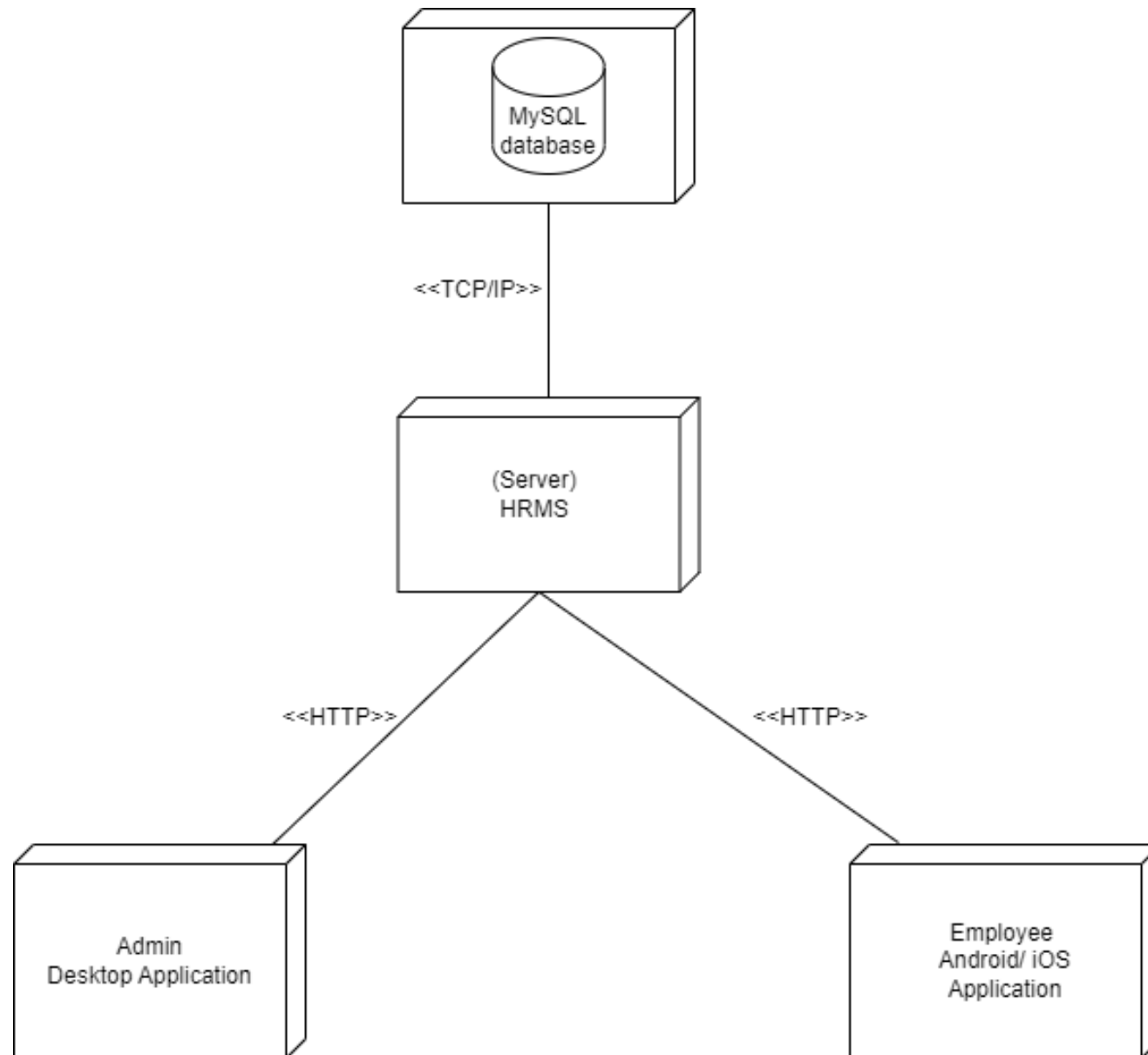
HRMS Activity Diagram



HRMS Component Diagram



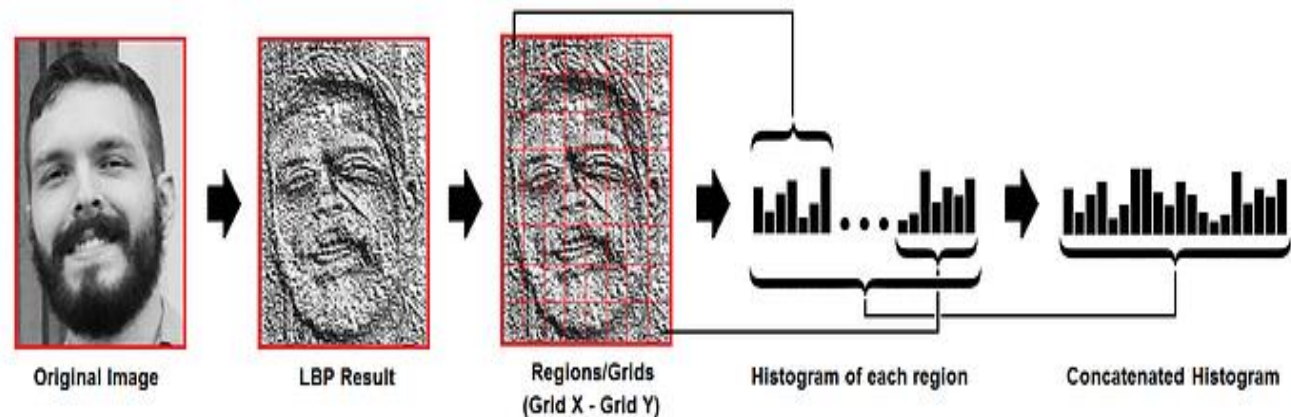
HRMS Deployment Diagram



ALGORITHM USED:

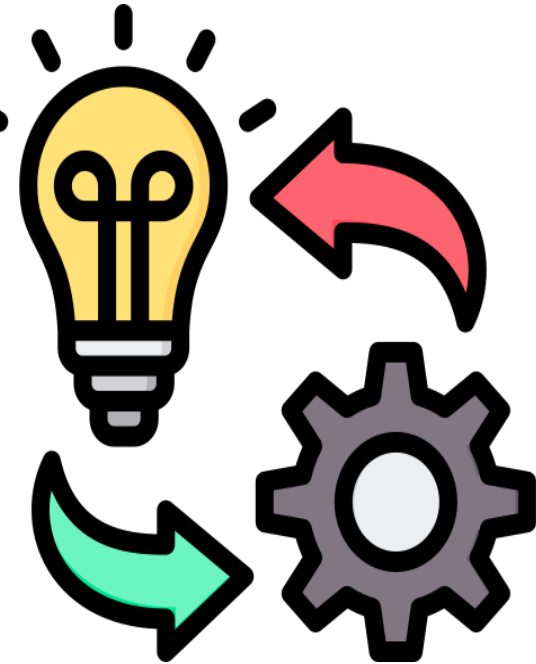
LBPH ALGORITHM:

Local Binary Pattern Histogram (LBPH) is a texture-based feature extraction technique used in computer vision applications like face recognition. It divides the image into small regions, calculates the Local Binary Pattern (LBP) features for each pixel within each region, and then constructs a histogram of the frequency of occurrence of the different LBP patterns. It is simple, efficient, and robust to illumination changes.



SOFTWARE IMPLEMENTATION:

Software implementation is the process of introducing new software into an organization's workflow, which involves complex tasks such as planning, designing, coding, and building the project. The implementation phase is the final step of the software development lifecycle and requires careful planning and execution to ensure a smooth transition from development status to production status.



TOOLS USED:



Flutter



Python



Django



MySQL



APIs



StarUML



Draw.io



PlantUML



Office 365

SOFTWARE TESTING :

Software testing of the HRMS project was done to identify defects , errors or vulnerabilities.

It comprises Verification and Validation.

Software testing includes:



Unit Testing



System Testing

UNIT TESTING :

Two different unit of the system were tested individually.

Table of Unit Testing Test Case 1

Test Case1	Input	Output	Test Result
Employee ID Input	Employee ID - 123	Employee details such as name, designation, department, etc.	The employee details were displayed correctly.

Table of Unit Testing Test Case 2

Test Case2	Input	Output	Test Result
Payroll Processing Error Handling	Invalid input such as negative salary or invalid employee ID	Error message indicating the issue with the input	The system correctly displayed an error message when invalid input was provided.

SYSTEM TESTING :

After the successful completion of unit testing, the system as a whole was tested to ensure whether the developed system was working well, and that the system meets its requirements or not. The test results were positive, that the system was working perfectly and has met the system requirements which were analyzed during the analysis and requirement phase.



CONCLUSION:

The HRMS project has the potential to significantly improve the efficiency and effectiveness of the company's HR processes. The feasibility analysis showed that the project is completely feasible, and the implementation phase was completed successfully. It successfully facilitates easy access to information and automates various Human Resource Management tasks resulting in a more efficient and structured organizational environment.

Thus, we believe our system will play a critical role in overcoming the challenges of human resource management tasks and decreasing the hassles that exist in the organizational sector.

FUTURE RECOMMENDATIONS



Develop system for remote workers & business trips to track attendance.



Ensure HRMS system is easily scalable for different business requirements.



Integrate with finance & accounting systems to streamline processes.

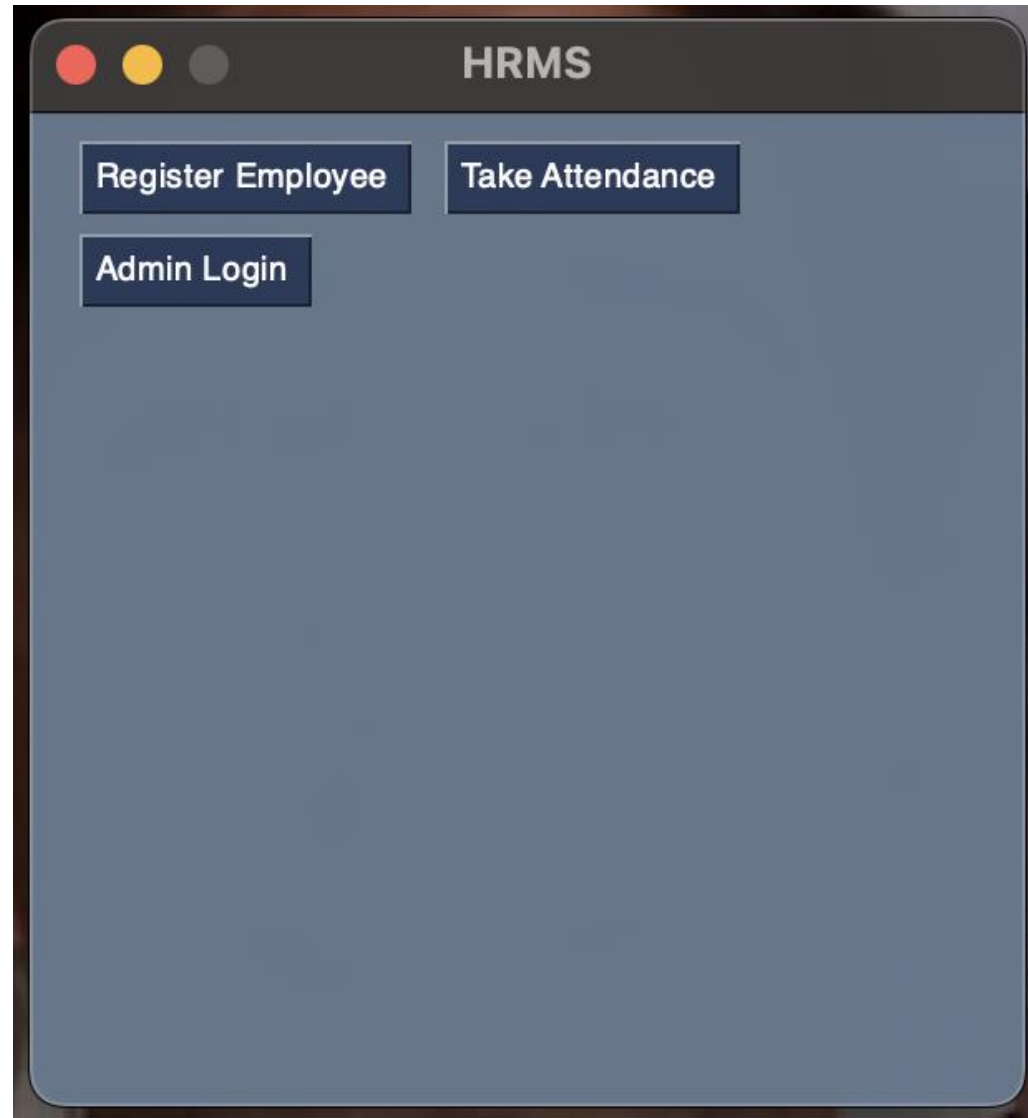


Address evaluator bias by using multiple evaluators and implementing measures.

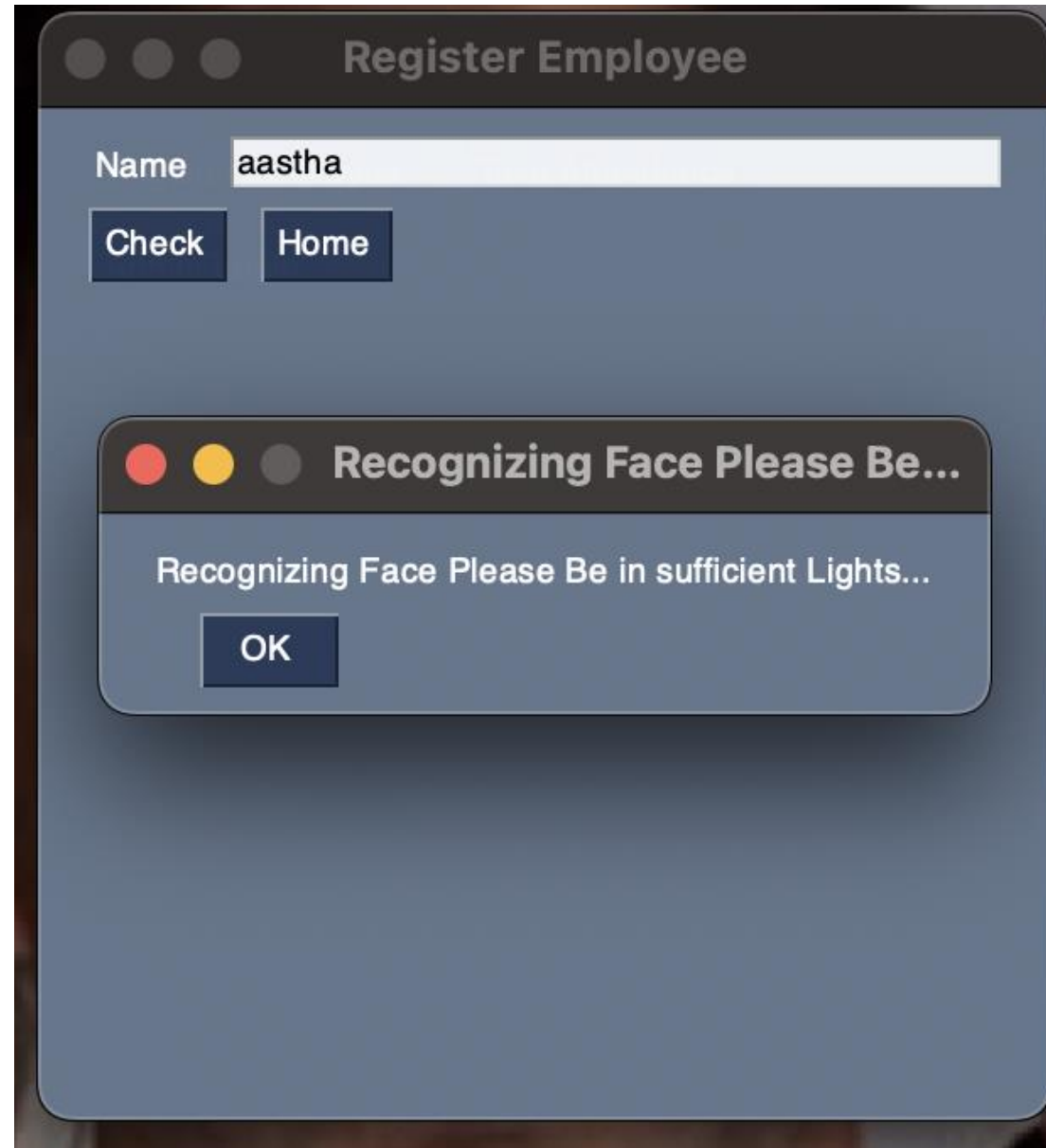


Explore expanding project to cover more HR functions & automate tasks.

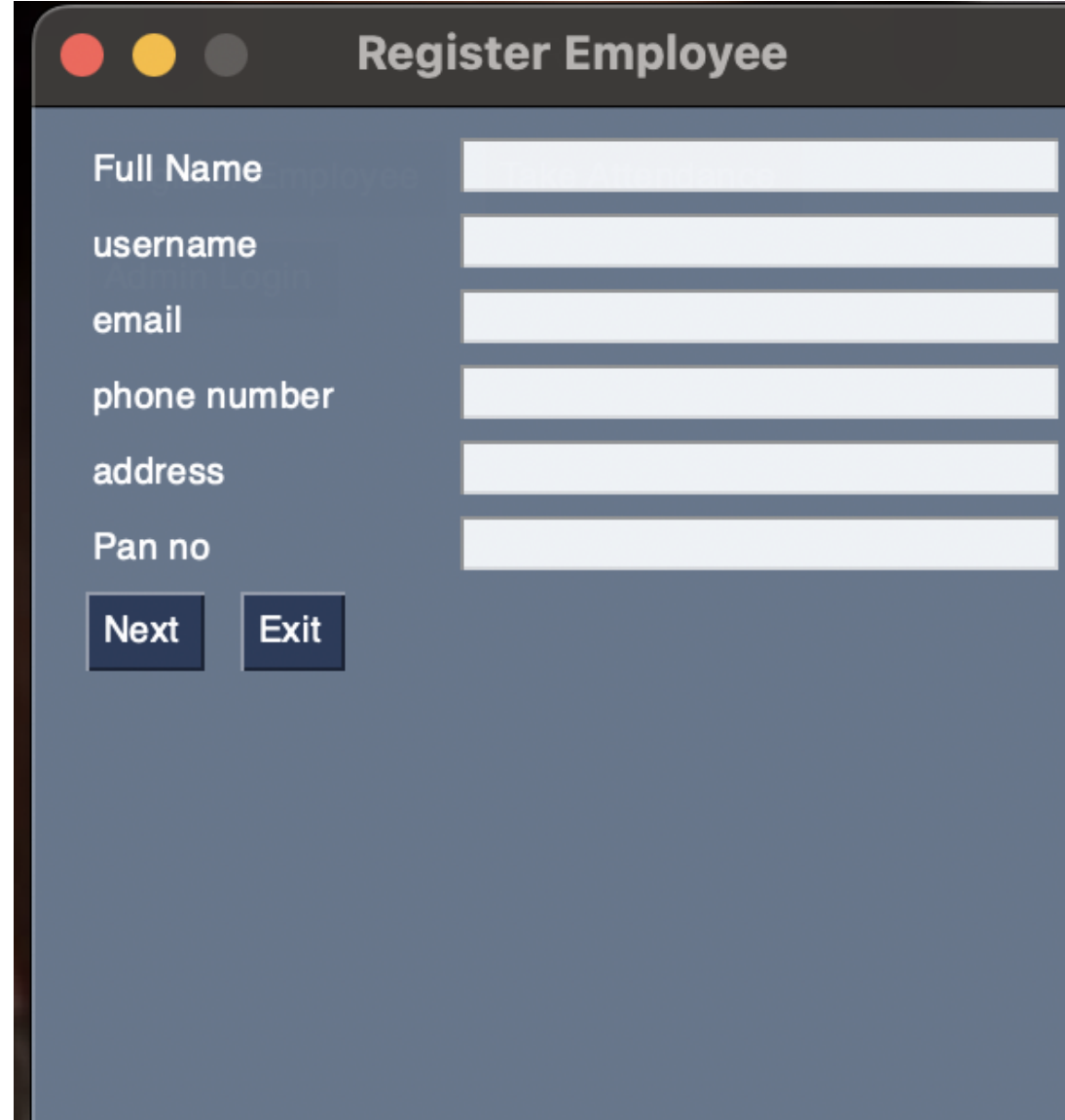
HRMS Admin Panel



Taking attendance using Face Recognizing



Employee Registration



A screenshot of a software application window titled "Register Employee". The window has a dark gray title bar with three colored window control buttons (red, yellow, and gray) on the left. The main content area has a blue-gray background. It contains six text input fields arranged vertically, each preceded by a label: "Full Name", "username", "email", "phone number", "address", and "Pan no". At the bottom left of the form, there are two dark blue buttons with white text: "Next" and "Exit".

Full Name	<input type="text"/>
username	<input type="text"/>
email	<input type="text"/>
phone number	<input type="text"/>
address	<input type="text"/>
Pan no	<input type="text"/>

Pay Roll Calculation

Payrolls										
id	name_id	year	month	salary	overtime	payroll_deduction	gross_salary	tax	tax_amt	net_salary
1	2	1904	Bhadra	30000.0	300.0	400.0	29900.0	0.15	4485.0	25415.0

Application Handling

Applications

All Applications

Choose Sta

Filter

id	type	title	reason	start_date	days
1	Suggestions	i need to suggest	People here are annoying	2023-04-13	2
2	Leave Request	i need leave	because i m very suck	2023-04-28	2
13	Suggestions	sdfdsf	dfdsf	2023-04-18	1
14	Leave Request	test	test	2023-04-17	1
15	Leave Request	Leave request for Friday	Please grant me a leave for 21st April, 2023 as I have to be present in my	2023-04-21	2

Select an Application to update its status

Choose ID

Choose status

Update Status

Code for Face Recognition - 1

```
if event == "Check":
    print("Checking")
    name = values['name'].capitalize()
    if not name:
        pg.popup("Please Enter all the details")
    else:
        employee = (name,)
        mycursor.execute(selemp,employee)
        myresult = mycursor.fetchone()
        if myresult == []:
            pg.popup("Can't find employee")
        else:
            name_id = myresult[0]
            pg.popup("Recognizing Face Please Be in sufficient Lights...",auto_close=True, auto_close_duration=5,)
            # Create a list of images and a list of corresponding names
            (images, labels, names, id) = ([], [], {}, 0)
            for (subdirs, dirs, files) in os.walk(datasets):
                for subdir in dirs:
                    names[id] = subdir
                    subjectpath = os.path.join(datasets, subdir)
                    for filename in os.listdir(subjectpath):
                        path = subjectpath + "/" + filename
                        label = id
                        images.append(cv2.imread(path, 0))
                        labels.append(int(label))
                    id += 1
            (width, height) = (130, 100)

            (images, labels) = [numpy.array(series) for series in [images, labels]]

            model = cv2.face.LBPHFaceRecognizer_create()
            model.train(images, labels)
            (_, cap_data) = video_cap.read()
            col = cv2.cvtColor(cap_data, cv2.COLOR_BGR2GRAY)
            faces = face_cap.detectMultiScale(col, 1.3, 5)
            for (x, y, w, h) in faces:
                cv2.rectangle(cap_data, (x, y), (x + w, y + h), (255, 0, 0), 2)
                face = col[y : y + h, x : x + w]
                face_resize = cv2.resize(face, (width, height))
                prediction = model.predict(face_resize)
                cv2.rectangle(cap_data, (x, y), (x + w, y + h), (0, 255, 0), 3)
                if prediction[1] < 500:
                    cv2.putText(
                        cap_data,
                        "%s - %.0f" % (names[prediction[0]], prediction[1]),
                        (x - 10, y - 10),
                        cv2.FONT_HERSHEY_PLAIN,
                        1,
```

Code for Face Recognition - 2

```
cv2.rectangle(cap_data, (x, y), (x + w, y + h), (0, 255, 0), 3)
if prediction[1] < 500:
    cv2.putText(
        cap_data,
        "% s - %.0f" % (names[prediction[0]], prediction[1]),
        (x - 10, y - 10),
        cv2.FONT_HERSHEY_PLAIN,
        2,
        (255, 255, 255),
    )
else:
    cv2.putText(
        cap_data,
        "not recognized",
        (x - 10, y - 10),
        cv2.FONT_HERSHEY_PLAIN,
        1,
        (255, 0, 0),
    )

if names[prediction[0]] == name and prediction[1] > 50.0:
    print("Match")
    now = datetime.datetime.now()
    date = now.strftime("%Y-%m-%d")
    time = now.strftime("%H:%M:%S")
    atnd = (name_id, date, time)
    mycursor.execute(attendancesave, atnd)
    mydb.commit()
    window = pg.Window("Your Attendance have been recorded", success_layout, size=(300, 300))
else:
    window = pg.Window("Error Occurred", error_layout, size=(300, 300))

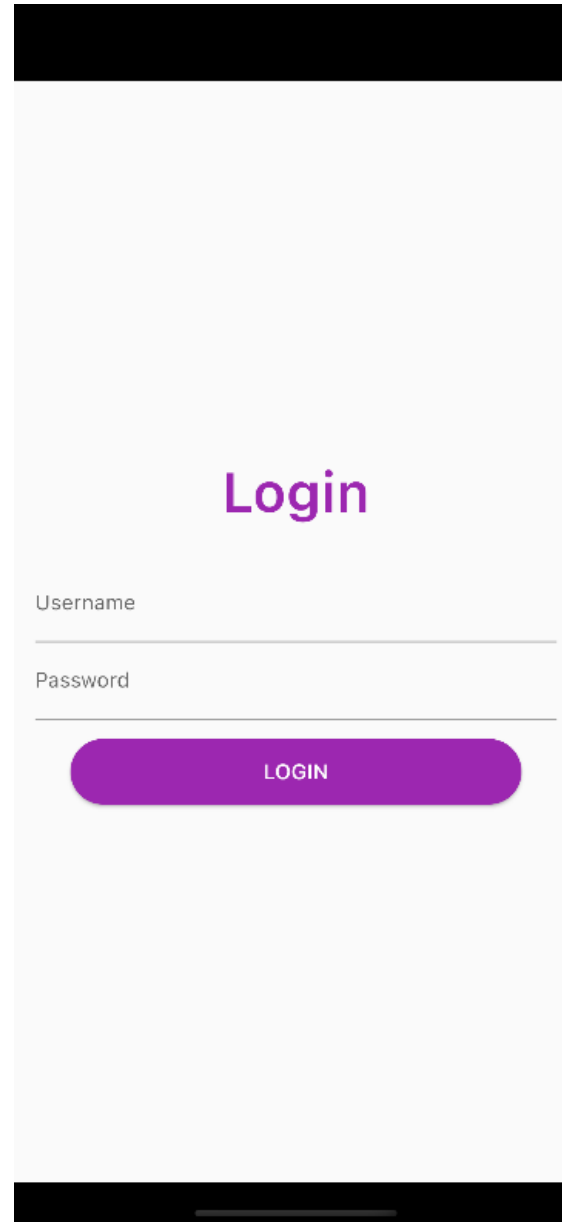
if event == "GeneratePass":
    characters = string.ascii_letters + string.digits + string.punctuation
    password = ''.join(random.choice(characters) for i in range(10))
    window["gen"].update(password)

if event == "Home":
    window = pg.Window("HRMS", another_layout, size=(300, 300))

if event == "Set Password":
    pass1 = values['pass1']
    pass2 = values['pass2']
    desg = int(values['desg'].split("-")[0])

    if not pass1 or not pass2 or not desg:
        pg.popup("Error")
    elif len(pass1) < 10:
```

Employee Mobile app Login Interface



A mobile app login interface with a black header and footer. The main content area is light gray. The word "Login" is centered in purple. Below it are two input fields: "Username" and "Password", each with a horizontal line. A purple rounded button labeled "LOGIN" is centered below the password field.

Login

Username

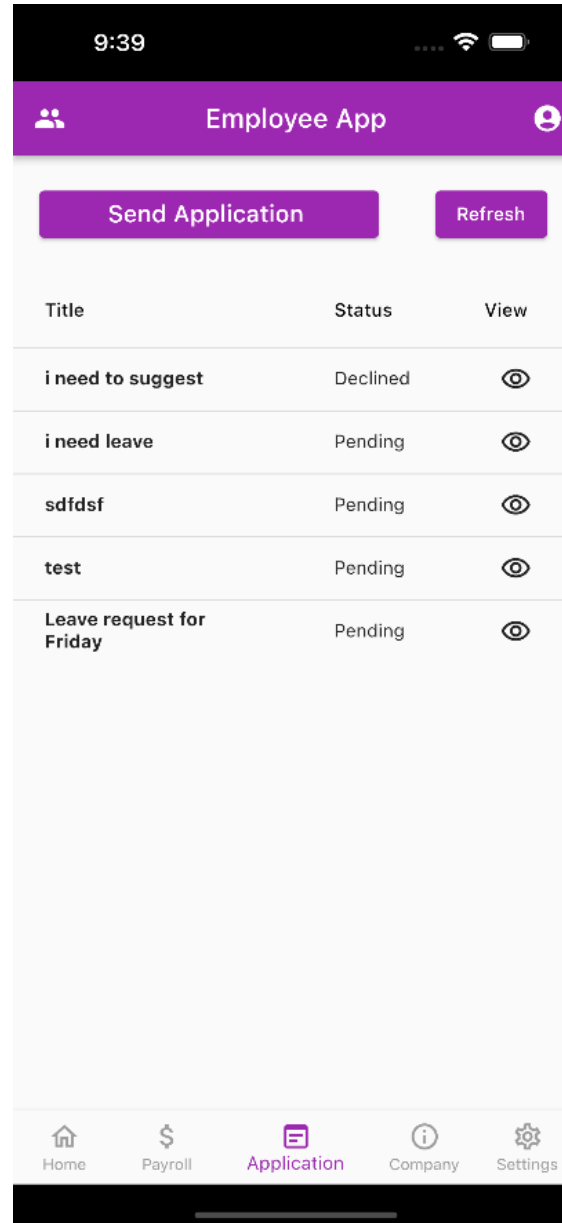
Password

LOGIN

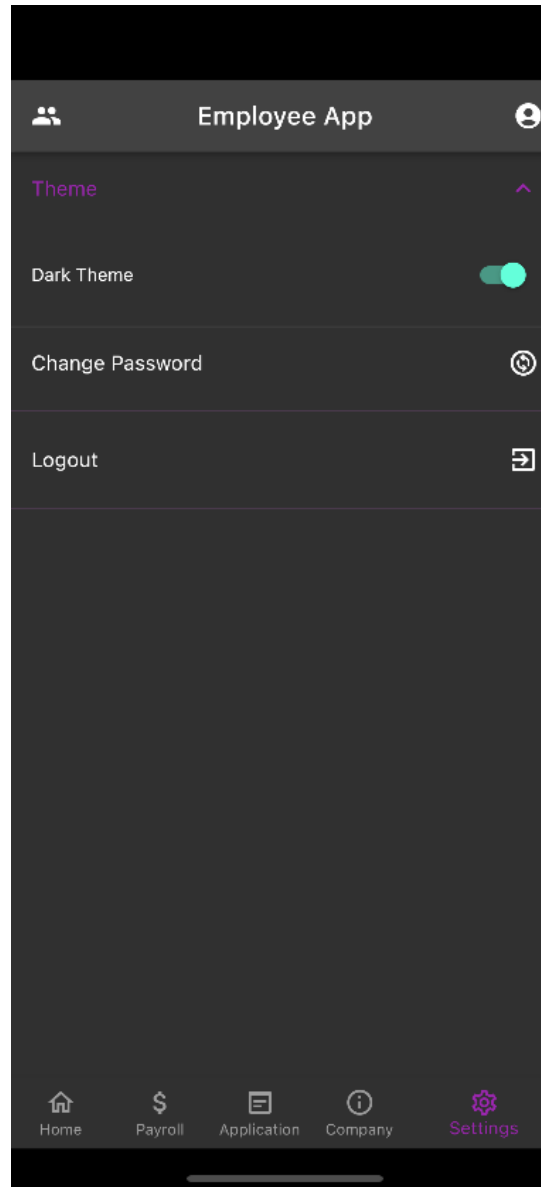
Employee Mobile App Home Interface



Employee Mobile App Application Interface



Employee Mobile App Settings Interface



List of APIs

Using the URLconf defined in `fyp.urls`, Django tried these URL patterns, in this order:

1. `api/ company/` [name='company']
2. `api/ employees/` [name='employee']
3. `api/ employee/<int:pk>/` [name='employee-view']
4. `api/ designations/` [name='designation']
5. `api/ attendances/` [name='attendance']
6. `api/ employeeattendance/<int:name_id>/` [name='attendance-view']
7. `api/ payrolls/` [name='payroll']
8. `api/ employeepayroll/<int:name_id>/` [name='payroll-view']
9. `api/ applications/` [name='applications']
10. `api/ employeeapplication/<int:pk>` [name='application-view']
11. `api/ login/` [name='login']
12. `api/ logout/` [name='logout']
13. `api/ logoutall/` [name='logoutall']