****

**ARISA EMS Requirements Specification**

**Group Members**

**1. Arisa Ndreu**

**2. Lorenc Noka**

**3. Vanesa Muka**

Table of Contents

1. EXECUTIVE SUMMARY....................................................................................................... 2

1.1 Project Overview……...................................................................................................................................2

1.2 Purpose and Scope of this specification…………………………………………………………………….2

2. Stakeholders………………………………………………………2

3. Requirements ………………………....………………………………………………………5

3.1 Functional Requirements…………………………………………………………………………………...9

3.2 Non-functional Requirements…………………………………………………………………………….9

3.2.1 Product Requirements………………………………………………………………………………...9

3.2.1.1 User Interface Requirements……………………………………………………………………...10

3.2.1.2 Usability…………………………………………………………………………………………..10

3.2.1.2.1 Accessibility…………………………………………………………………………………...10

3.2.1.2.2 Responsiveness………………………………………………………………………………10

3.2.1.2.3 Flexibility……………………………………………………………………………………...10

3.2.1.2.4 Effectiveness………………………………………………………………………………….10

3.2.1.2.5 Efficiency……………………………………………………………………………………..11

3.2.1.3 Efficiency…………………………………………………………………………………………11

3.2.1.3.1 Performance Requirements……………………………………………………………………11

3.2.1.3.2 Space Requirements…………………………………………………………………………...11

3.2.1.4 Dependability……………………………………………………………………………………11

3.2.1.4.1 Availability……………………………………………………………………………………11

3.2.1.4.2 Maintenance ………………………………………………………………………………...11

3.2.1.4.3 Integrity………………………………………………………………………………………..11

3.2.1.5 Security……………………………………………………………………………………………12

3.2.2 Domain Requirements………………………………………………………………………………...12

4. SOFTWARE DESIGN…........................................................................................................ 12

4.1 Use Cases……………………………………………………………………………………………18

4.2 Use Case Extended………………………………………………………………………………………20

4.3 BPMN……………………………………………………………………………………………………21

4.4 Activity Diagrams………………………………………………………………………………………..22

4.5 State Diagrams…………………………………………………………………………………………...26

4.6 Sequence Diagrams……………………………………………………………………………………..30

4.7 Class Diagrams…………………………………………………………………………………………35

4.8 Collaboration Diagrams……………………………………………………………………………….42

4.9 DFD……………………………………………………………………………………………………49

4.10 Communication Diagrams……………………………………………………………………………52

4.11 Component Diagrams…………………………………………………………………………………58

4.12 Object Diagrams………………………………………………………………………………………60

4.12 Deployment Diagram…………………………………………………………………………………63

4.13 Package Diagram……………………………………………………………………………………..64

5. DESIGN PATTERNS………………………………………………………………………………65

***1. Executive Summary***

*1.1 Project Overview*

ARISA CONFECTION was founded in 1993, located in Tirana, Albania. Its main activity is the making of different types of clothes such as T-shirts, jackets, blazers, pants, capes, hoodies, etc. They get orders from different companies in Italy and Germany mainly. Their products are then exported in Italy, Germany and their partners. Due to the high amount of work that the company and its workers have to deal in the daily basis the efficiency is not 100% all the time. And this is exactly why we as developers thought to create such a software to optimize the amount of time that different people in the company spent to fulfill different tasks given.

By building AEMS, we tend to give the company a long-time solution to deal with its problems, by optimizing the time spent on daily tasks responsible for future growth.

*1.2 Purpose and scope of this Specification*

Our purpose as developers is to simplify most of the tasks that the company is dealing with in the daily life, make them easy manageable and less time consuming as possible. The software will be able to provide the following:

1. Recordkeeping

2. Worker Connection

3. Quality controlling

4. Fast access to most used kind of records

5. Location updating

6. Employee tracking

* 1. *Stakeholders*

**2.1.1 Core Team:**

*2.1.1.1* Project Manager: Responsible for overall project planning, execution, and success.

*2.1.1.2* Software Developers: Design and develop the employee management system.

*2.1.1.3* Quality Assurance/Testers: Ensure the system is bug-free and meets quality standards.

* + - 1. Business Analysts: Gather and document system requirements.

**2.1.2 Immediate Stakeholders**:

*2.1.2.1* Human Resources (HR) Team: They are the primary users of the system and play a critical role in defining requirements.

*2.1.2.2* IT Department: Provides technical support, infrastructure, and integration expertise.

*2.1.2.3* End Users (Employees): Those who will interact with the system regularly.

**2.1.3 Management and Leadership:**

*2.1.3.1* Executive Management: Responsible for strategic alignment and project funding.

*2.1.3.2* Department Managers: Utilize the system for employee performance assessments, resource allocation, and reporting.

*2.1.3.3*Legal and Compliance Teams: Ensure the system aligns with labor laws and regulations.

**2.1.4 Supporting Stakeholders:**

*2.1.4.1* Payroll Department: Integration with payroll systems for salary processing.

*2.1.4.2* Finance Department: Budgeting and financial tracking related to human resources.

*2.1.4.3* IT Security: Ensures data security and system protection.

*2.1.4.4* Training and Development: Provides training and onboarding for employees using the system.

*2.1.4.5* Employee Unions or Representatives: May need to be consulted regarding changes in employee management processes.

**2.1.5 External Stakeholders**:

2.1.5.1Vendors and Suppliers: If the system relies on external services or products.

2.1.5.2Government Agencies: For compliance with labor and tax regulations.

2.1.5.3Industry Associations: For best practices and benchmarks in employee management.

*2.2 The project infrastructure.*

**2.2.1 Core Project Goals and Objectives**:

2.2.1.1Defining the primary objectives of the project, e.g., improving HR processes, increasing efficiency, andensuring compliance with labor laws.

**2.2.2Project Planning and Execution**:

2.2.2.1 Project Management: Creating a project plan, defining milestones, and allocating resources.

2.2.2.2 Requirement Gathering: Collecting detailed requirements from stakeholders.

2.2.2.3 System Design: Creating a blueprint of the employee management system.

2.2.2.4 Development: Building the system based on the design and requirements.

2.2.2.5 Testing: Ensuring the system functions as intended and is free of defects.

**2.2.3 Infrastructure and Integration**:

2.2.3.1 IT Infrastructure: Setting up the necessary hardware and software infrastructure.

2.2.3.2 Data Integration: Connecting the employee management system with other systems like payroll, time tracking, and databases.

2.2.3.3 Security Measures: Implementing data security and access controls.

**2.2.4 Stakeholder Engagement**:

2.2.4.1 User Training: Preparing employees and HR staff to use the system effectively.

2.2.4.2 Change Management: Managing the transition from existing HR processes to the new system.

2.2.4.3 Communication: Keeping stakeholders informed about project progress and changes.

2.2.4.4 Feedback Collection: Gathering input from users and making necessary adjustments.

**2.2.5 External Factors**:

2.2.5.1 Regulatory Compliance: Ensuring the system adheres to labor laws and industry regulations.

2.2.5.2 Industry Best Practices: Incorporating HR and IT best practices.

2.2.5.3 Market Trends: Considering emerging technologies and practices in HR and IT.

2.2.5.4 Economic and Environmental Impact: Assessing the financial and ecological consequences of the project.

*3.1 Functional Requirements*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Req# | Requirement | Comments | Priority | Date Rvwd | reviewed/approved |
| BR\_01 | The software should support inventory management system. | This helps on better organization of work. | 1 |  |  |
| BR\_02 | The software should support employee payroll system. | Make it easier to identify/change wages. | 1 |  |  |
| BR\_03 | The software should be able to assist in the preparing of company financial records. | When financial reports needed, they can be easily found. | 1 |  |  |
| BR\_04 | The software should include a special part for the location of the company. | For new employees/customers to find easier. | 2 |  |  |
| BR\_07 | The software may provide an employee chat system (only for important announcements). | Not everyone might check the whatsapp or other social media. | 1 |  |  |
| BR\_08 | The super-admin must be able to create the employees profiles. | Since the data of employees must remain private. | 1 |  |  |
| BR\_09 | The super-admin should be able to check the all employee wages. This data will be best to be represented in tabular form. | More organization | 1 |  |  |
| BR\_11 | Only the super-admin must have access in the employee chat system. | Not to create confusion and to keep clear the function of the chat. | 1 |  |  |
| BR\_12 | Super-admin must have access in the financial records of the company. He can add financial records related with the business. All the data will be represented in tabular form, also provided pie charts and column diagrams. | In order not to create confusion on mass. | 1 |  |  |
| BR\_14 | The super-admin can give permission to workers to leave, by clicking the button “approved” when receives the notification by the department head. In the notification will be specified the hour, name/surname of the employee, what is his duty, the reason why asks to leave and the department header name/surname. The notification will have 2 options “Accepted” “Denied”. | This will be less time consuming than face-to-face method. | 2 |  |  |
| BR\_16 | The MANAGER can check the monthly quantities of products generated. This data will be presented in tabular form. | This allows him to calculate the work left to be done and also not to waste time. | 1 |  |  |
| BR\_18 | The MANAGER can register his workers and their working-hours. This data will be registered in tabular form. | This is done in order to keep track of the working hours (appx 40h/w) and also to have a fair competition for the “employee of the month”. | 2 |  |  |
| BR\_19 | The MANAGER can ask for permission to take sick leave for himself & other workers under his command. In order to do it he must fill a form with basic information like: the hour, name/surname of the employee, what is his duty, the reason why asks to leave and the department header name/surname. After he must press the button “Ask for permission”. | This will be less time consuming than face-to-face method. | 2 |  |  |

*3.2 Non-Functional Requirements*

**3.2.1 Product Requirements**

*3.2.1.1 User Interface Requirements*

* The system is thought to be simple and communicative, so it won’t be necessary for the user to read the guideline. In order to achieve this, the system will not have any complicate interfaces for any of its users.
* The main page will be informative page about the business and also will display different pictures on the background. At the top is thought to have a main menu in which the user can select the profile he wants to be identified during the usage of the app (administrator, manager, etc) and after selecting a simple log in page will be shown requiring the email/username and also the password. At the end of the page will be shown the contact information (a phone number, company’s instagram, facebook and email).
* The administrator page (super-admin page) will be organized in different sectors for each of company’s processes including: production, finance, transport, security, chat etc. It is thought that when the administrator will be logged in, all the sites where he has access will be displayed in the middle of the screen separated in squares, so not to create any confusion. When clicking the desired page they want to go, when the page will be opened the other pages names will be displayed in as a column in the left of the screen.
* The managers page (admin page) will be organized in different sectors for each of company’s processes including: production, finance, transport, security, chat etc. It is thought that when the admin will be logged in, all the sites where he has access will be displayed in the middle of the screen separated in squares, so not to create any confusion. When clicking the desired page they want to go, when the page will be opened the other pages names will be displayed in as a column in the left of the screen.
* The DH page (admin page) will be organized in different sectors for each of company’s processes including: production, finance, transport, security, chat etc. It is thought that when the DH will be logged in, all the sites where he has access will be displayed in the middle of the screen separated in squares, so not to create any confusion. When clicking the desired page they want to go, when the page will be opened the other pages names will be displayed in as a column in the left of the screen.

*3.2.1.2 Usability*

**3.2.1.2.1****Accessibility:**

As mentioned before, AEMS will be an application and it can be accessed only if the user has internet connection and also is inside the business environments.

**3.2.1.2.2 Responsiveness:**

The application is thought to be extremely responsive and fast too.

**3.2.1.2.3****Flexibility:**

The application is thought to adapt the updates very easy but also deal with errors/problems as quick as possible.

**3.2.1.2.4****Effectiveness:**

* This application will be easy to use and understand and as user-friendly as possible.
* To minimize the chances of failing in understanding AEMS, in the app will be included a section called “About Me”, in which each user can find all the procedures needed to be done in order for him to use the application.
* In case the user deals with an error, the messages will be easy to understand and also associated with a step-by-step procedure in order to get rid of it.

**3.2.1.2.5****Efficiency:**

* The application is going to be very efficient and 0 time-consuming, meaning that each user will fulfill his tasks super fast and in the best case with no errors/warnings at all.
* The interface will also be very easy and user-friendly with no complicated buttons or actions.

*3.2.1.3 Efficiency*

**3.2.1.3.1 Performance Requirements**

* Since AEMS is going to be installed in a computer/laptop/mobile, the data will be delivered in real-time, and it will only be accessible inside company’s environment it will require good internet connection, and new tech devices in order to reach its best performance, but also with mid-range devices it will work just fine.
* The bigger the space, the more will be the users therefore the better the performance.

**3.2.1.3.2 Space Requirements**

* The app will be able to handle at least 100 users at a time, and the database system should handle at least 300-400 users at any given time.
* Maximum user load: 100.

*3.2.1.4 Dependability*

**3.2.1.4 .1Availability:**

* The application will be available only during the working hours, but for specific users it will be available 24/7.
* The app will be accessed only inside the business environments.
* Internet access is necessary.

**3.2.1.4 .2Maintanence:**

* The app will be updated when necessary in order to process all the operations in real time.
* In case the system will crash, the app should not display anything to the user except an error message “The system isn’t available right now! See you soon!”
* If the crash happens the system should start as soon as possible.

**3.2.1.4 .3Integrity:**

* The personal information must be available only for the super-admin.
* The super-admin account is the only one responsible for adding/deleting new employee accounts.
* All users should provide personal credentials to log in into the system and should be authenticated before accessing their own profiles.

*3.2.1.5 Security*

* When the super-admin creates a new profile, an informative email must be sent to the employee to let him know about the action.
* When a new user is logged in, a randomly generated password must be sent to it by the super-admin email.

**3.2.2 Domain Requirements**

AEMS is going to be an application that can be easily downloadable from both AppStore and PlayStore, and is going to be an organizational app exclusively for the “Arisa” company. The main purpose of this app is the digitalization of the company’s data and careful handling of the production information. The application is thought to be divided in 2 parts for use, one use will be from the company workers (CEO, managers, employees) and the other one will be for clients but with a limited actions (like viewing products and making orders-specifically big orders). The app will be mostly employee-use oriented. In order to access the functionalities of the app the user must firstly be logged in with the email that they have been registered to the company and the whole data will be manipulated into company’s servers. A must for using the application will be location of the user (inside company’s environments) and stable internet connection.

***4. Software Designs***

**4.1 User Scenarios**

*1. Successful log in:*

1. The user will manually fill the email input box.

2. The user will manually fill the password input box.

3. The app will check inputs with database records, if they match the user is successfully logged in.

4. In the users’ screen will be displayed the home page of his/her own page.

*2. Failing to log in*

1. The user will manually fill the email input box.

2. The user will manually fill the password input box.

3. The app checks the input with the database records, inputs doesn’t match.

4. A message will be displayed to user, asking to enter the wrong input again.

5. User remains at the log in page.

**Administrator Scenarios:**

*1. Creating the employee account*

1. The administrator is firstly logged in.

2. He clicks on “Employee Accounts” section, on a left column in the screen where are listed all sections.

3. After page is opened it selects the option “Create” from 3 options, “Create”, “Delete”, “Update”.

4. The administrator starts filling the field with information about the new employee.

5. After adding all the information, he clicks the button “Confirm”.

6. A message “Are you sure you have entered the correct information” will be shown in the screen.

7. Administrator clicks “Yes” and the account is successfully created.

*2. Deleting the employee account*

1. The administrator is firstly logged in.

2. He clicks on “Employee Accounts” section, on a left column in the screen where are listed all sections.

3. He searches for the name of the employee at the search bar.

4. When found, the administrator clicks on the button “Delete”.

5. A confirmation table will be shown in the middle of the screen with message “Are you sure you want to delete it?”

6. After confirmation, the account will be deleted but not permanently (it will basically be sent into a bin).

*3. Controlling employees payments*

1. The administrator is firstly logged in.

2. He clicks on “Employee Accounts” section, on a left column in the screen where are listed all sections.

3. He searches for the name of the employee at the search bar.

4. Inside the account is the “Monthly payment” field.

*4. Update the employee chat system*

1. The administrator is firstly logged in.

2. He goes to section “Chat” on a left column in the screen containing all sections.

3. He types the message that want to sent to the employees and press “Sent”.

*5. Control financial records*

1. The administrator is firstly logged in.

2. He goes to section “Finance” on a left column in the screen containing all sections.

3. Another page organized in square will be shown where each square represents one financial record like: Sales revenue, Service Revenue, Unpaid Revenue, Miscellaneous expense, Accounts receivable, Income Summary, Retained Earnings etc.

4. The administrator click to one of the squares and the table containing information for that specific record will be shown on the screen.

*6. Permissions for sick leaves (this operation can be done through the phone)*

*6.1 Request accepted*

1. A notification will be sent to the administrator’s phone.

2. The administrator logs in.

3. He goes to section “Requests” on a left column in the screen containing all sections.

4. He reads the request sent from the department header.

5. He accepts the request by clicking “Confirm”.

*6.2 Request denied*

1. A notification will be sent to the administrator’s phone.

2. The administrator logs in.

3. He goes to section “Requests” on a left column in the screen containing all sections.

4. He reads the request sent from the department header.

5. He denies the request by clicking “Deny”.

6. An input box will be shown in the screen with description “Reason”.

7. Administrator fills the box with a brief explanation why the request was denied and press “Sent”.

8. The message is sent to the department head.

**Head Managers (admin-s) scenarios:**

*Finance Manager:*

1. He is logged in.

2. A page organized in square will be shown where each square represents one financial record like: Sales revenue, Service Revenue, Unpaid Revenue, Miscellaneous expense, Accounts receivable, Income Summary, Retained Earnings etc.

3. The manager click to one of the squares and the table containing information for that specific record will be shown on the screen.

*Security Manager:*

1. He is logged in.

2. A page organized in square will be shown where each square represents one environment of the company like: CEO office, Production line 1, Production line 2, Yard, Warehouse, Manager’s Office etc.

3. The manager click to one of the squares and the current cameras recording will be shown on the screen.

4. He can set on/off alarms by clicking a button in the top right of the screen.

**Department Header Scenarios:**

*1. Fulfilling daily reports of production:*

1. The department header is firstly logged in.

2. In front of him will be shown a button “Fill a daily report” by default.

3. When clicked it will open a form with necessary fields to fill (Date, name, surname, employees included, working hours etc).

4. After filling the user will press “Send”.

5. A confirmation table will be shown up.

6. After pressing “Yes” the report will be totally submitted.

*2. Check monthly quantities of products generated:*

1. The department header is firstly logged in.

2. In the left column of sections, he will click on “Product” section.

3. After clicking a table with the name of product, its code, quantity, deadline and details columns will be shown on the screen.

4. At the search bar at the top of the page checks for product by entering its name or code and then check the quantity.

*3. Check deadline of products generated:*

1. The department header is firstly logged in.

2. In the left column of sections, he will click on “Product” section.

3. After clicking a table with the name of product, its code, quantity, deadline and details columns will be shown on the screen.

4. At the search bar at the top of the page checks for product by entering its name or code and then check the deadline.

*4. Registering his workers and working hours:*

1. The department header fills the daily report.

2. Before submitting the form he marks as “Yes” the “Register employees and working hours box.

*5. Ask for permission:*

1. The department header is logged in.

2. In the left column of sections, he will click on “Permissions” section.

3. A form with different fields including: name of employee, date, day, reason for request and his name, will appear on the screen.

4. The department header press “Ask” button.

*4.1 Onion Models*

**External Stakeholders**

**Supporting Stakeholders**

**Management and Leadership**

**Immediate Stakeholders**

**Core Team**

**The project**

* + 1. The onion model for the stakeholders of our project.

**External Factors**

**Stakeholder Engagement**

**Infrastructure and Integration**

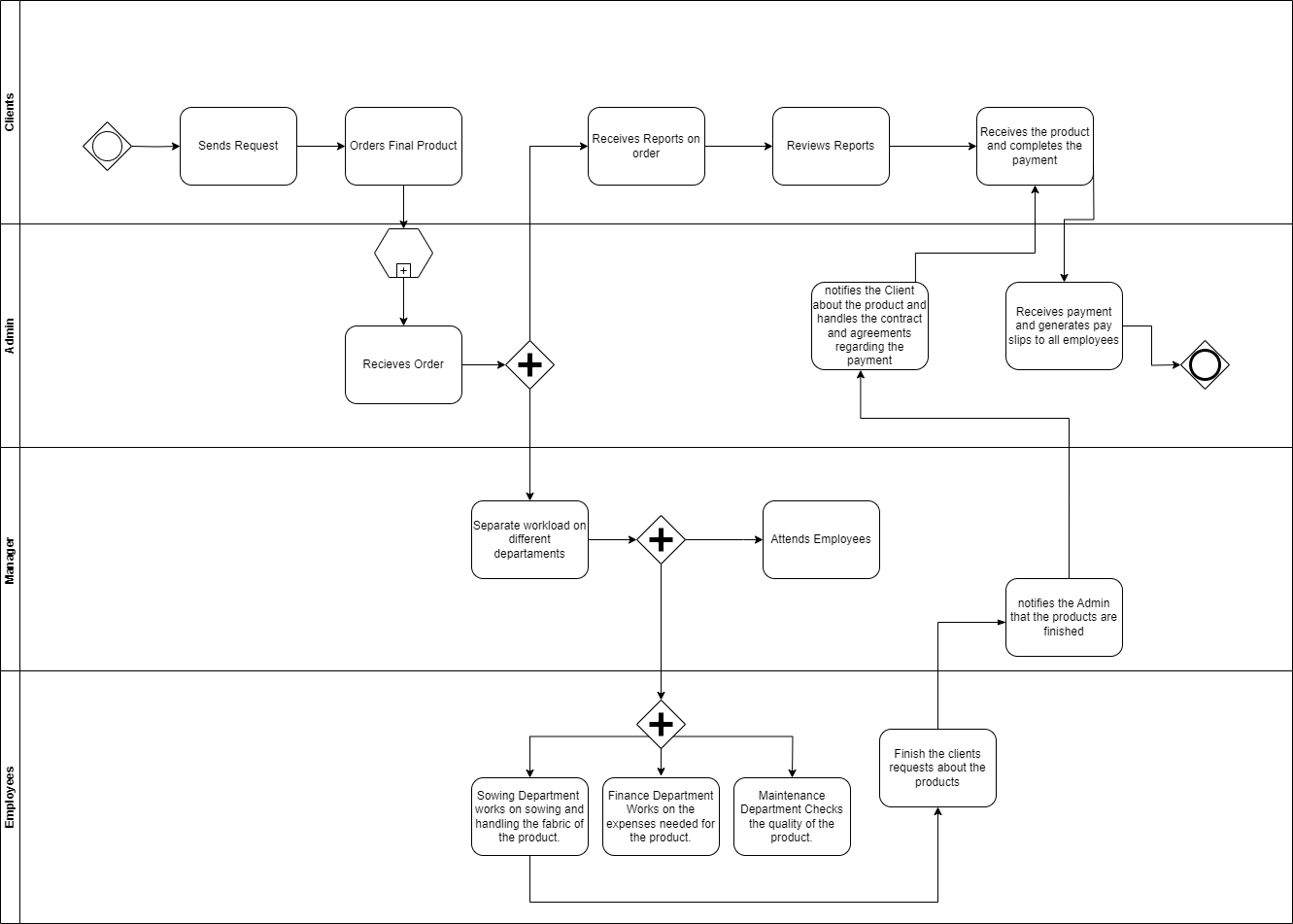
**Project Planning and**

**Execution**

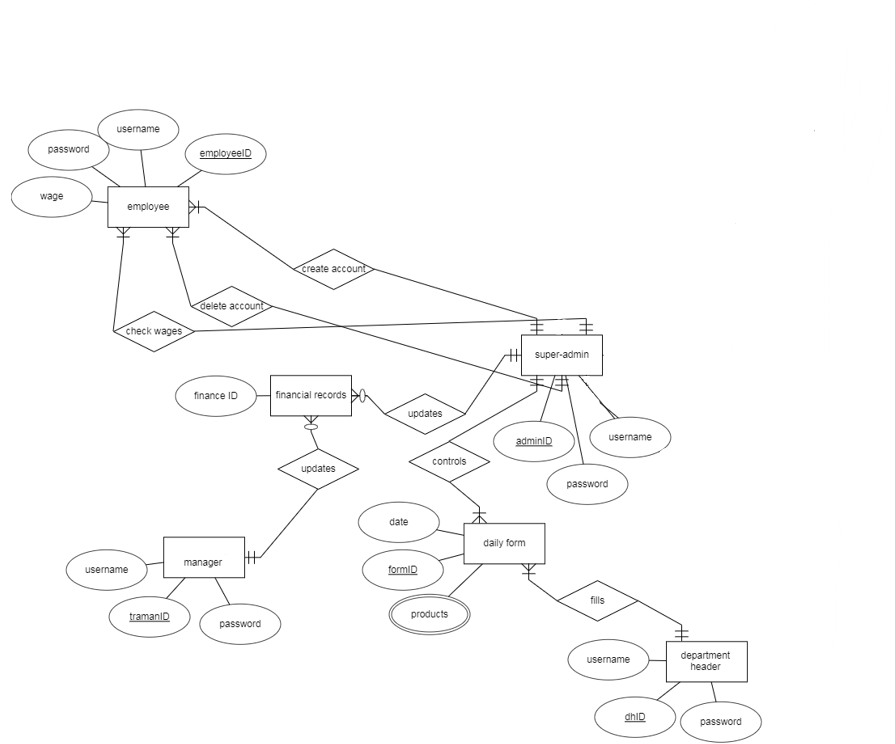
**Core Project Goals**

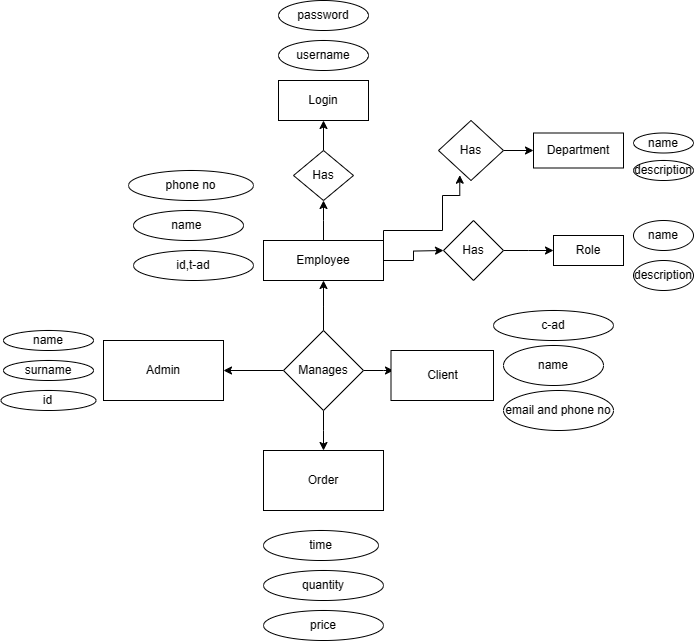
**and Objectives**:

* + 1. The onion model for our project infrastructure.
  1. BPMN, ERD, RS
     1. BPMN

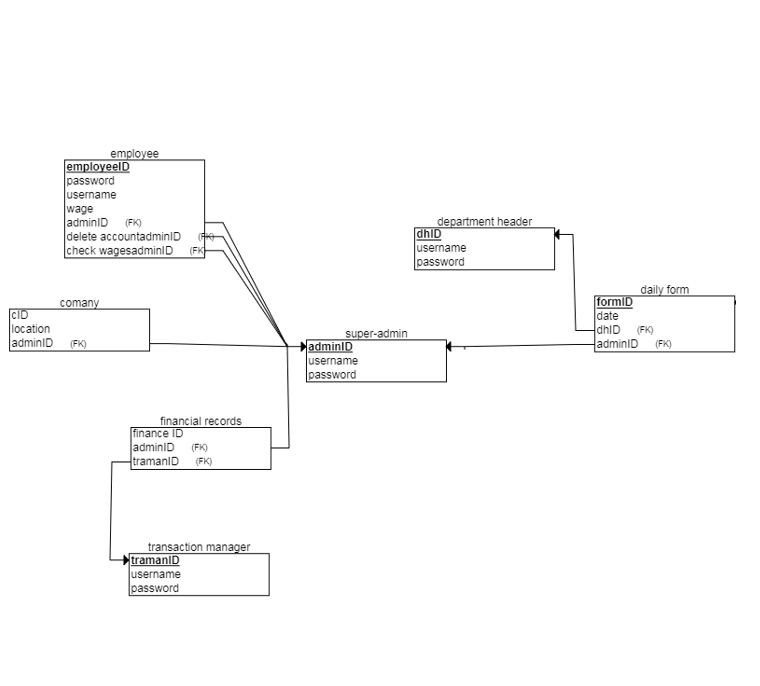


* + 1. ERD



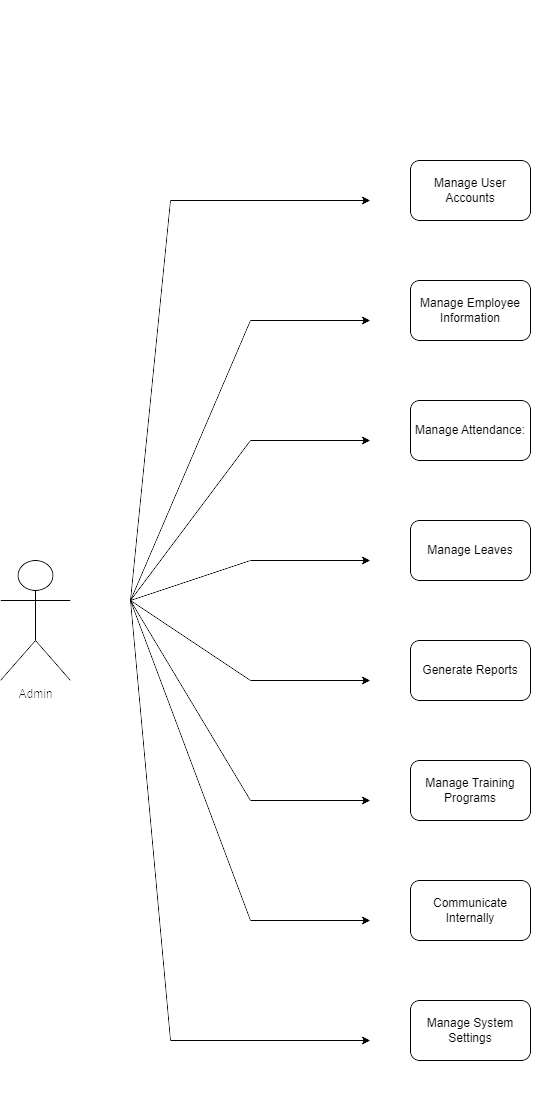


* + 1. RS

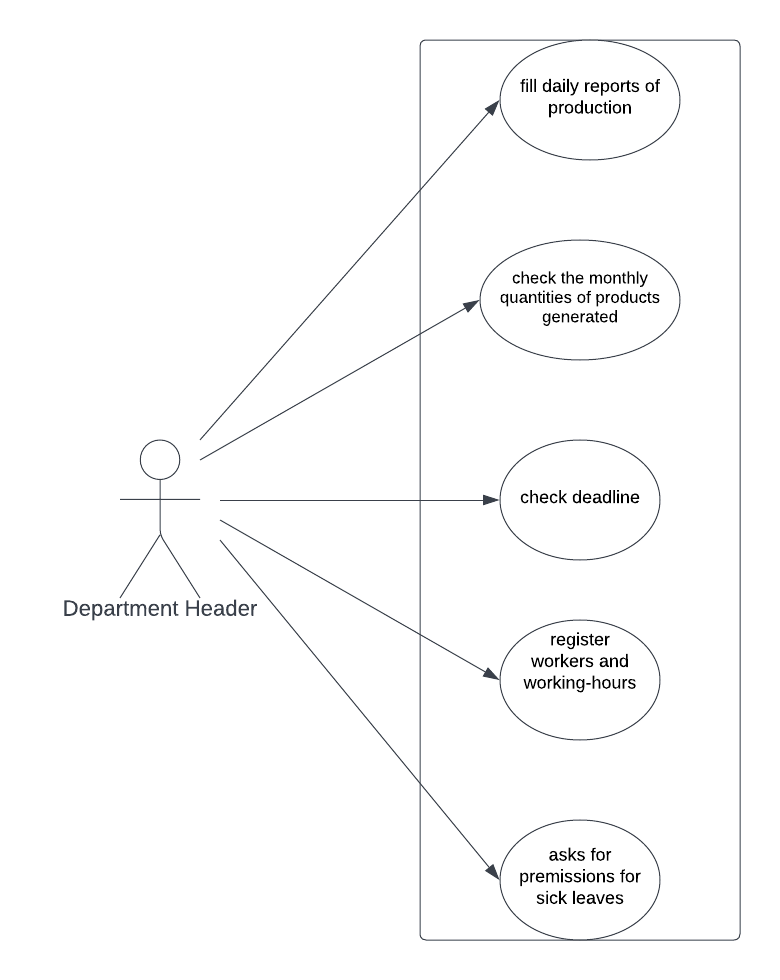


* 1. Use Case

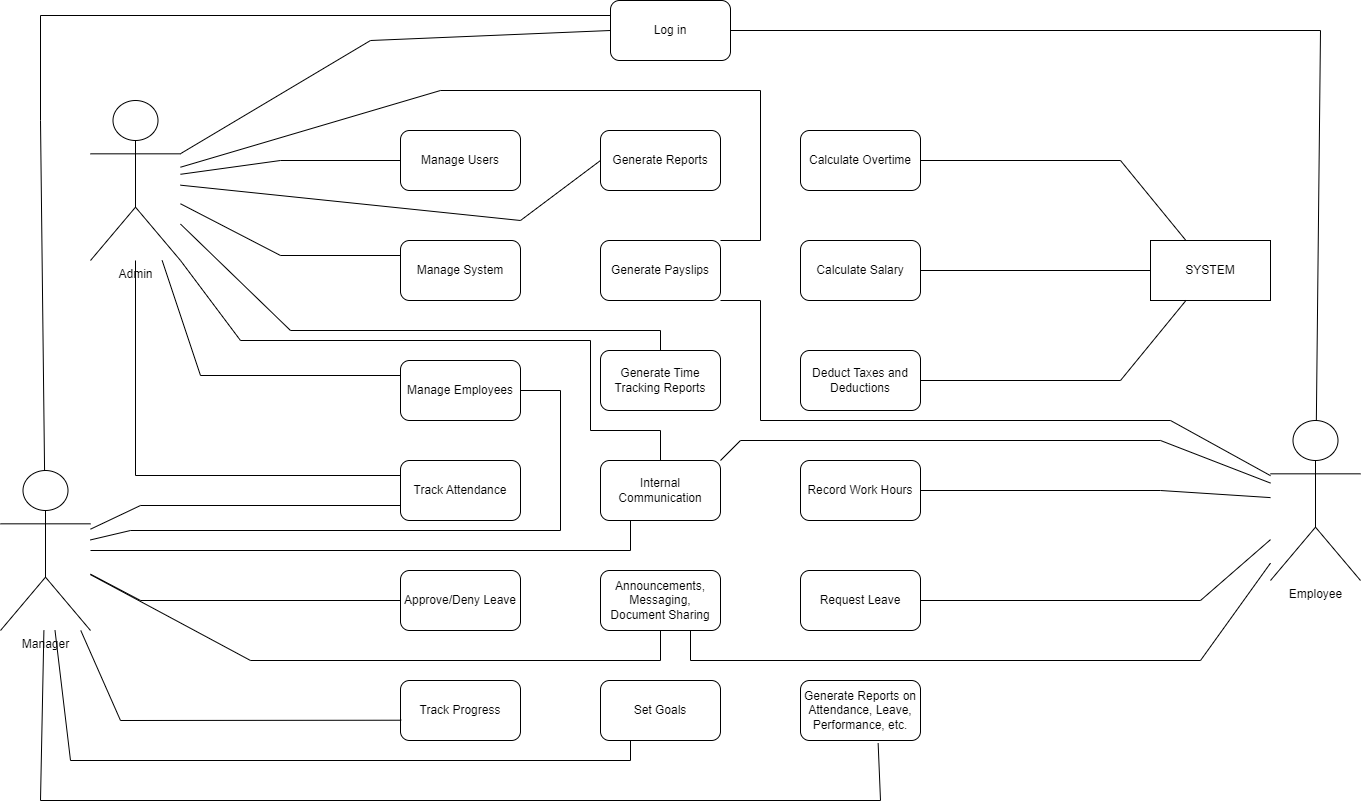
4.3.1 Administrator



4.3.2 Department Head



4.3.3



**4.3 Use Case Extended**

|  |  |
| --- | --- |
| Use Case (UC\_1.1) | Create Employee Profile |
| Scope: | AEMS application |
| Level: | User Level |
| Intention Context: | Each time a new employee comes to the company he must have an account. |
| Minimum Guarantees: | Account is created but not used. |
| Success Guarantees: | Account is successfully created, given to the employee and used by the employee. |
| Primary Actor: | Administrator |
| Stakeholder’s interest: | To simplify the work for specific actions and save time. |
| Precondition: | At least 1 employee. |

|  |  |
| --- | --- |
| Use Case (UC\_1.2) | Delete Employee Profile |
| Scope: | AEMS application |
| Level: | User Level |
| Intention Context: | Each time a new employee leaves the company the account must be deleted. |
| Minimum Guarantees: | Account is not deleted. |
| Success Guarantees: | Account is successfully deleted. |
| Primary Actor: | Administrator |
| Stakeholder’s interest: | To simplify the work for specific actions and save time. |
| Precondition: | At least 1 employee. |

|  |  |
| --- | --- |
| Use Case (UC\_1.3) | Check Employee Payments |
| Scope: | AEMS application |
| Level: | User Level |
| Intention Context: | As times as necessary check the employee payment. |
| Minimum Guarantees: | Administrator doesn’t check. |
| Success Guarantees: | When necessary, he checks the payment of specific employees. |
| Primary Actor: | Administrator |
| Stakeholder’s interest: | To simplify the work for specific actions and save time. |
| Precondition: | At least 1employee. |

|  |  |
| --- | --- |
| Use Case (UC\_1.4) | Control Employee Chat System |
| Scope: | AEMS application |
| Level: | User Level |
| Intention Context: | When there are new and important news about the company that all the employees should be informed |
| Minimum Guarantees: | Administrator sends a short message to all employees. |
| Success Guarantees: | Administrator successfully informs all the employees about a specific event. |
| Primary Actor: | Administrator |
| Stakeholder’s interest: | To simplify the work for specific actions and save time. |
| Precondition: | At least 1 event/news. |

|  |  |
| --- | --- |
| Use Case (UC\_1.5) | Control Financial Records |
| Scope: | AEMS application |
| Level: | User Level |
| Intention Context: | For administrator to be informed about the financial state of the its company. |
| Minimum Guarantees: | Administrator checks the financial records once in a while. |
| Success Guarantees: | Administrator checks the financial records frequently. |
| Primary Actor: | Administrator |
| Stakeholder’s interest: | To simplify the work for specific actions and save time. |
| Precondition: | At least 1 financial record. |

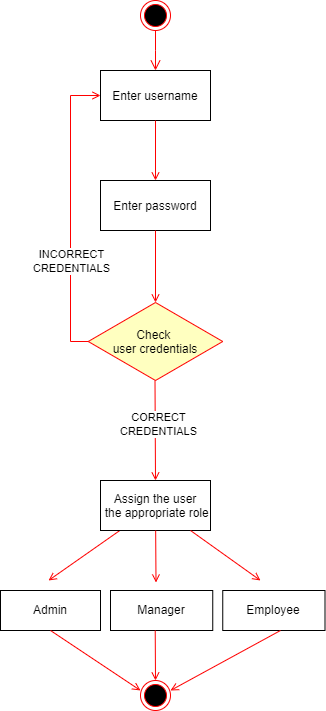
|  |  |
| --- | --- |
| Use Case (UC\_1.6) | Give permissions for sick leaves |
| Scope: | AEMS application |
| Level: | User Level |
| Intention Context: | In order not to waste time with meetings with many employees. |
| Minimum Guarantees: | Administrator just confirms/denies the request. |
| Success Guarantees: | Administrator gives an explanation and confirms/denies the request. |
| Primary Actor: | Administrator |
| Stakeholder’s interest: | To simplify the work for specific actions and save time. |
| Precondition: | At least 1 employee. |

|  |  |
| --- | --- |
| Use Case (UC\_2.1) | Fill daily reports for production |
| Scope: | AEMS application |
| Level: | User Level |
| Intention Context: | For the department header to report to the manager/administrator about the daily production. |
| Minimum Guarantees: | D.H reports shortly daily production. |
| Success Guarantees: | D.H reports a complete daily production. |
| Primary Actor: | Department Header |
| Stakeholder’s interest: | To simplify the work for specific actions and save time. |
| Precondition: | At least 1 employee and 1 product. |

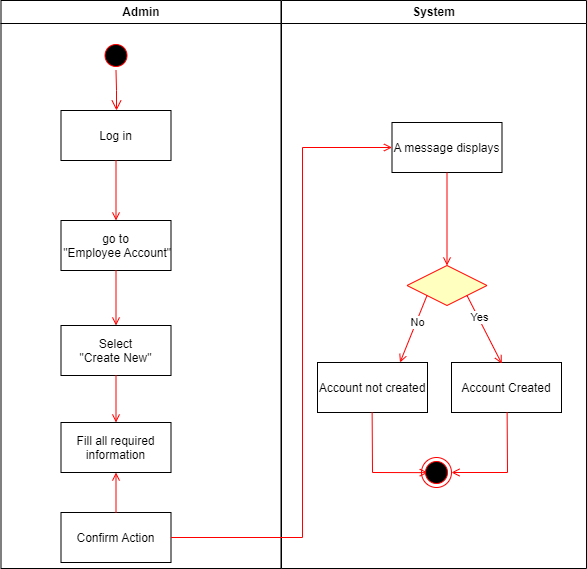
|  |  |
| --- | --- |
| Use Case (UC\_2.2) | Register workers and working-hours |
| Scope: | AEMS application |
| Level: | User Level |
| Intention Context: | In order to keep track of each employee work so at the end to be easier to find the “employee of the month”. |
| Minimum Guarantees: | D.H registers only a few. |
| Success Guarantees: | D.H registers frequently. |
| Primary Actor: | Department Header |
| Stakeholder’s interest: | To simplify the work for specific actions and save time. |
| Precondition: | At least 1 employee. |

|  |  |
| --- | --- |
| Use Case (UC\_2.3) | Asks for permissions for sick leaves |
| Scope: | AEMS application |
| Level: | User Level |
| Intention Context: | When an employee wants to leave, he doesn’t need to go to meet the manager but can ask the D.H to request a sick leave. |
| Minimum Guarantees: | D.H sends the request with minimum explanation. |
| Success Guarantees: | D.H sends the request with a complete explanation. |
| Primary Actor: | Department Header |
| Stakeholder’s interest: | To simplify the work for specific actions and save time. |
| Precondition: | At least 1 employee. |

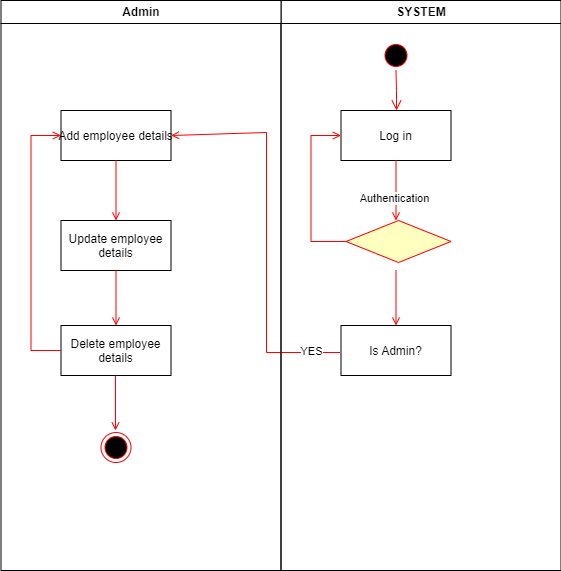
* 1. Activity Diagrams
     1. Log in



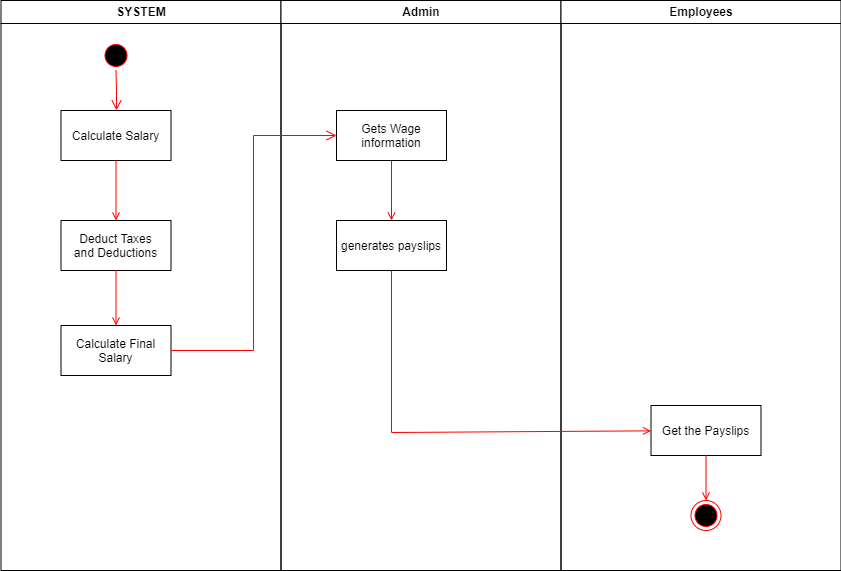
4.4.2 Creating Employee Accounts



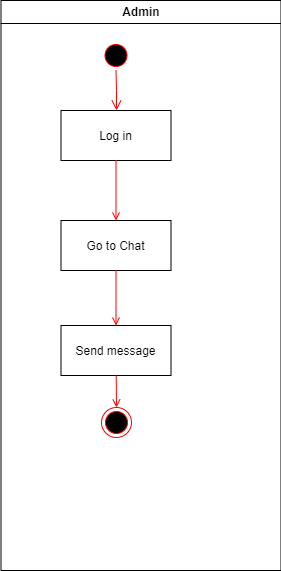
4.4.3 Deleting Employee Accounts



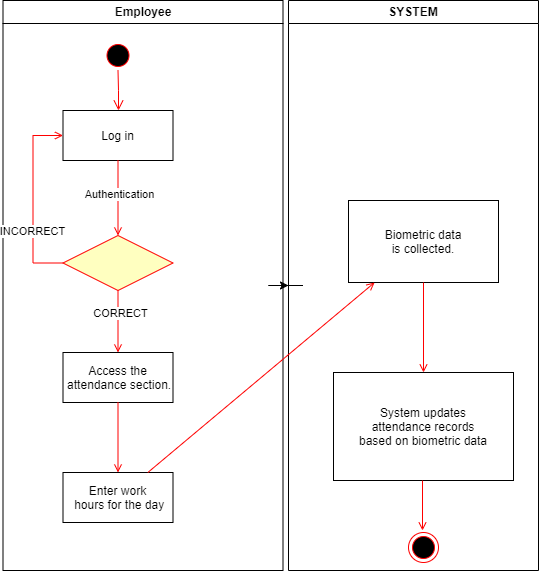
4.4.4 Payment



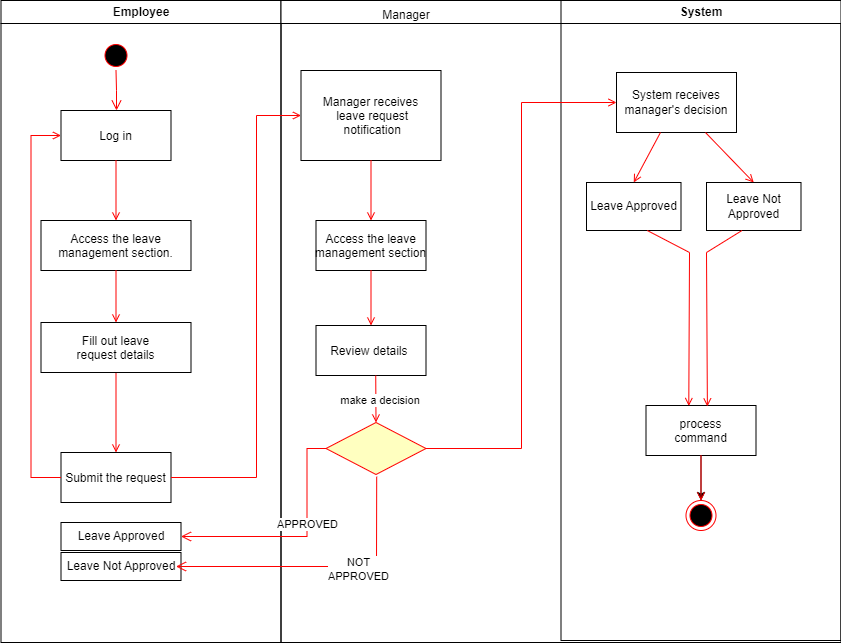
* + 1. Chat System



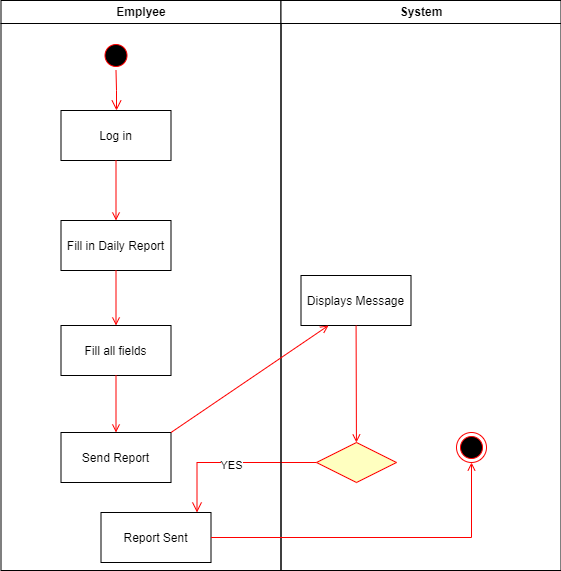
* + 1. Attendance Management



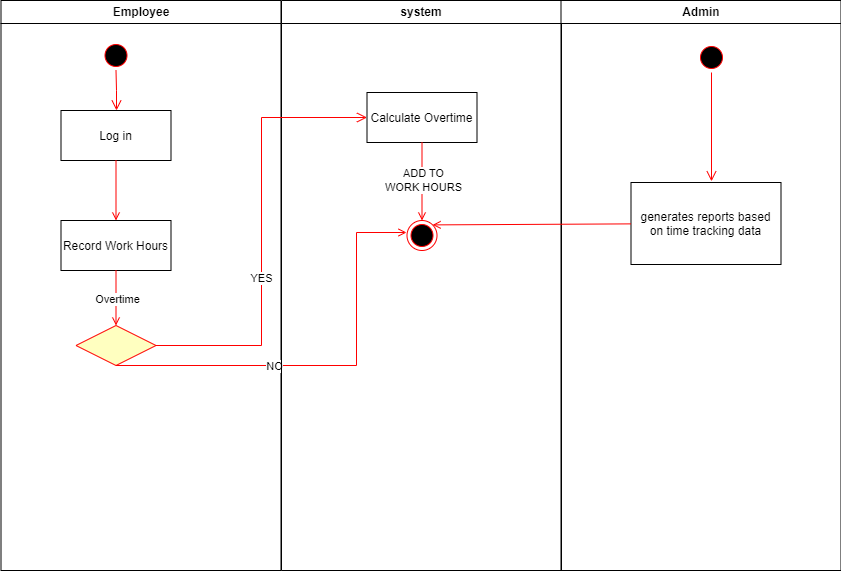
4.4.7 Permission For Sick Leave



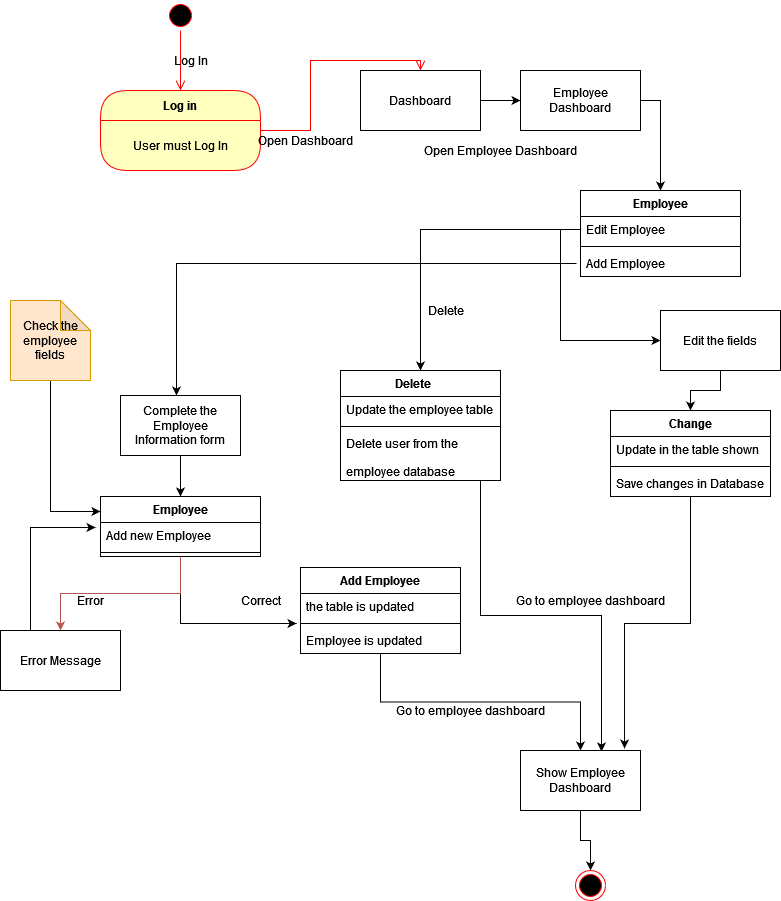
4.4.8 Daily reports



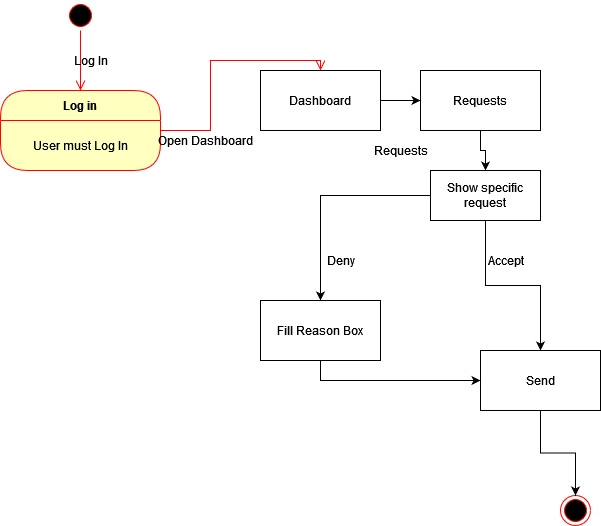
4.4.9 Register Working hours



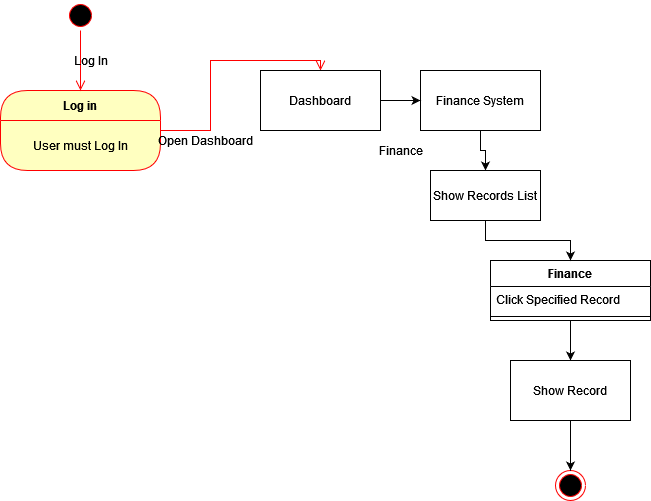
* 1. State Diagrams
     1. Employee



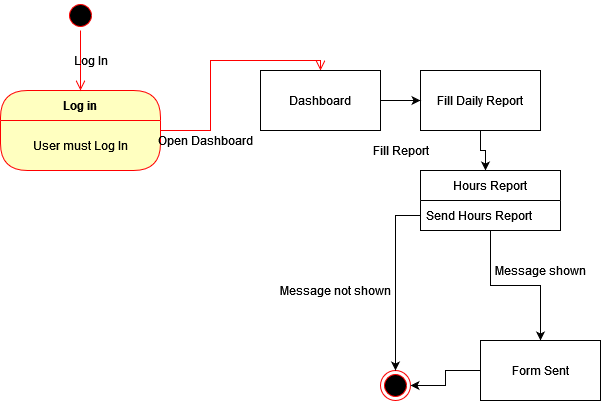
* + 1. Permission For Sick Leaves



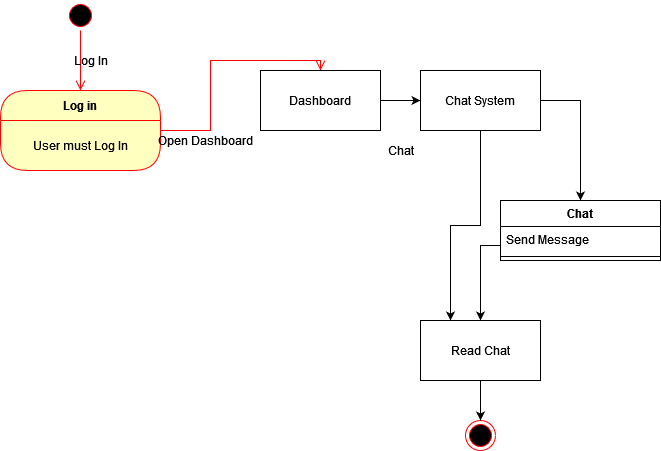
* + 1. Searching Finance Reports



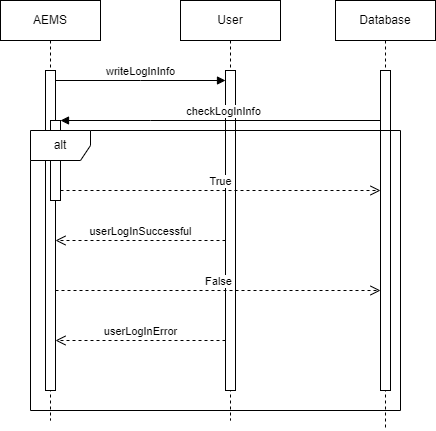
* + 1. Working Hours

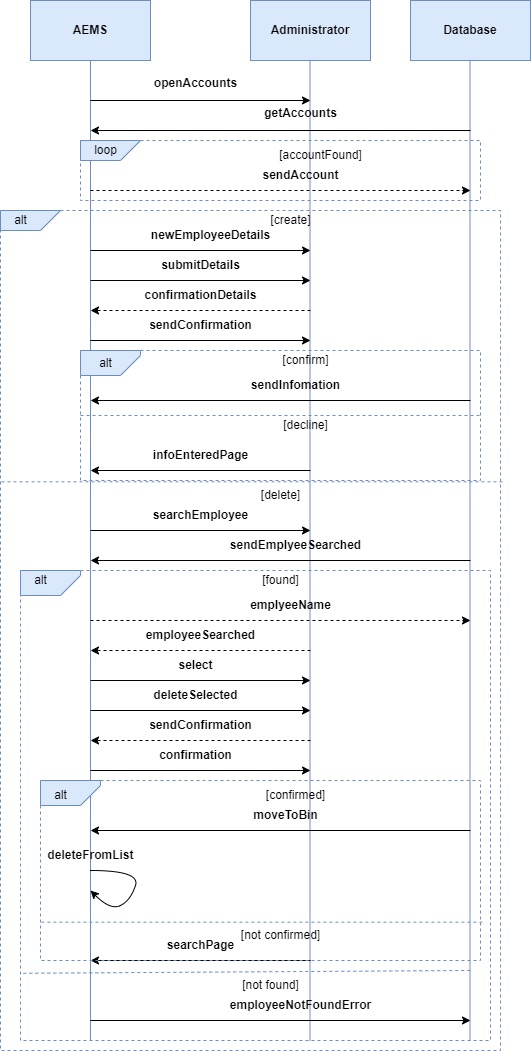


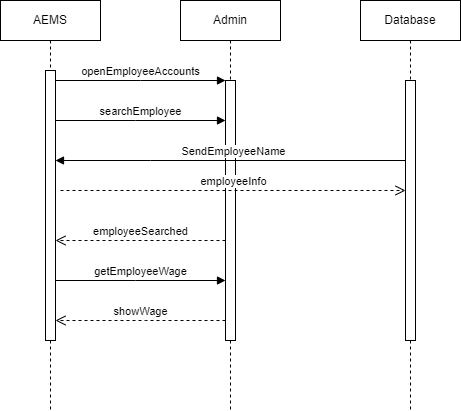
* + 1. Update Chat System



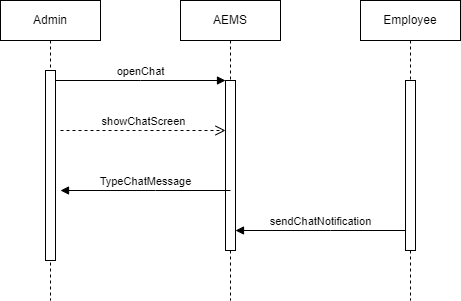
* 1. Sequence Diagrams
     1. Log In



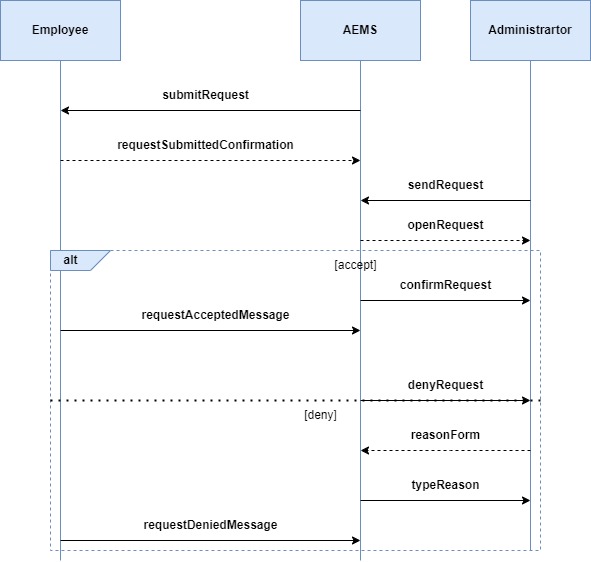
* + 1. Employees
    2. Employee Wage



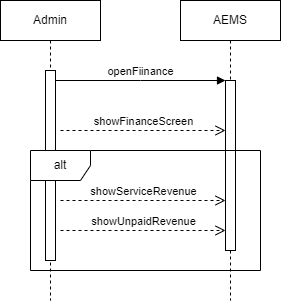
* + 1. Update Chat



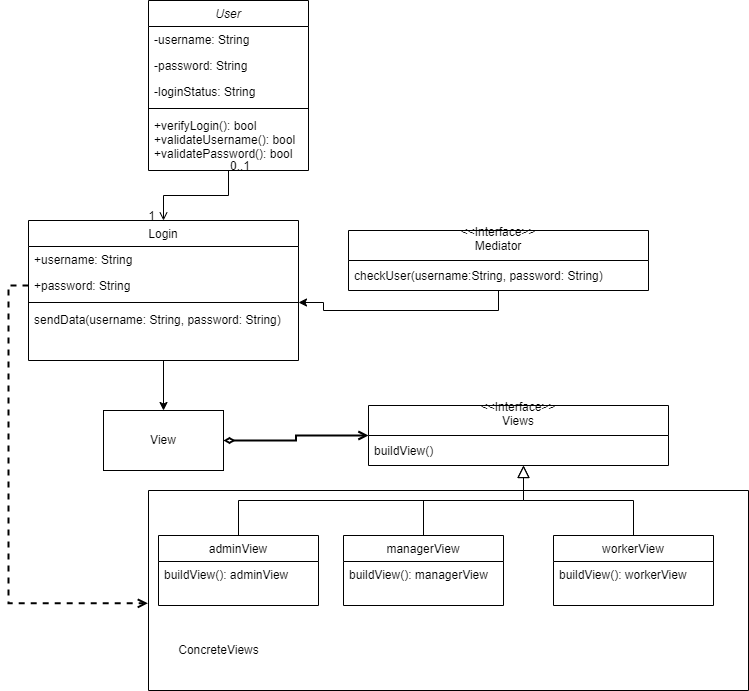
* + 1. Permission For Sick Leaves



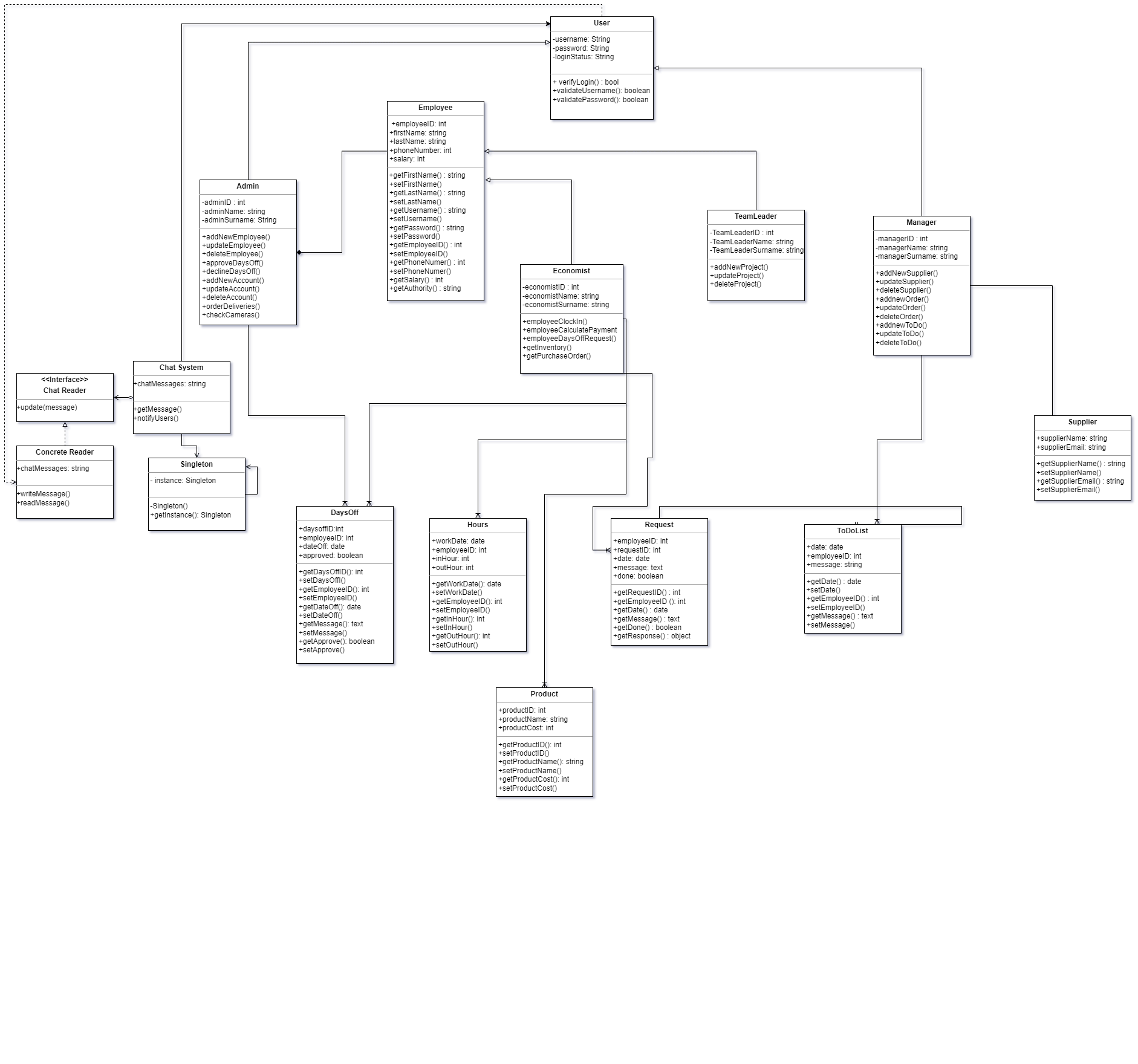
* + 1. Finance



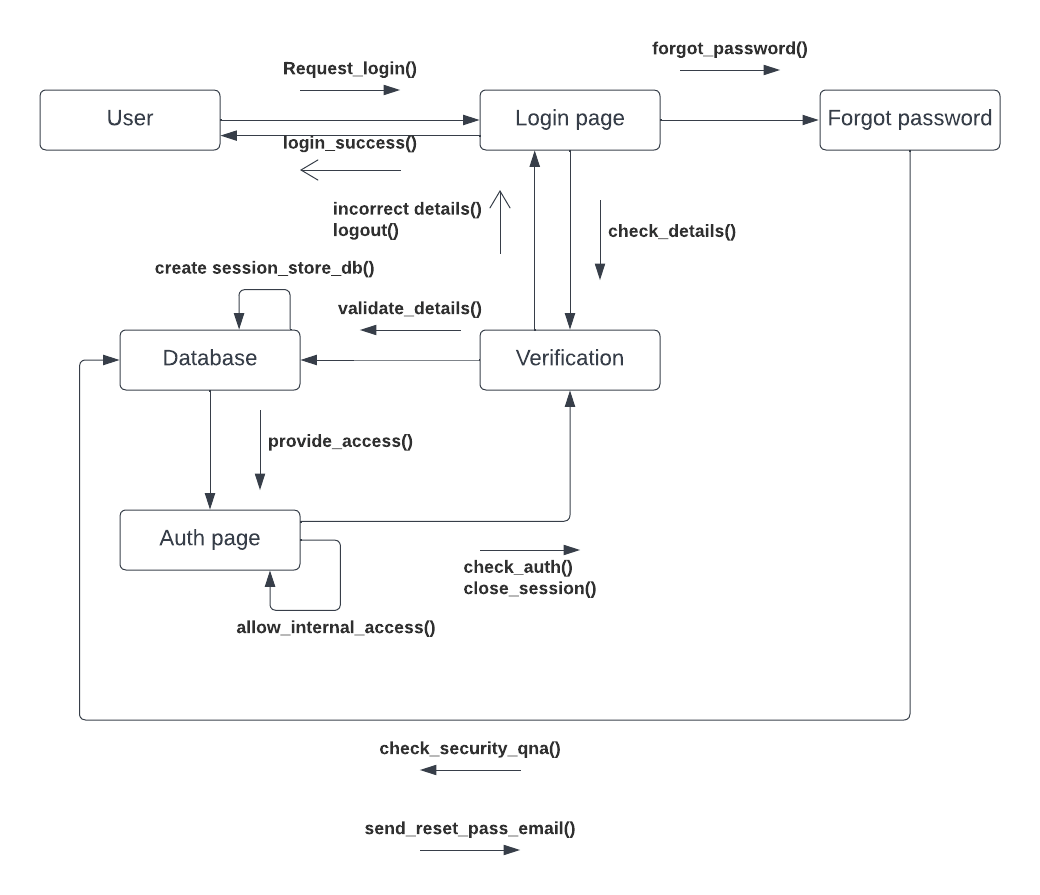
* 1. Class Diagrams
     1. Log in



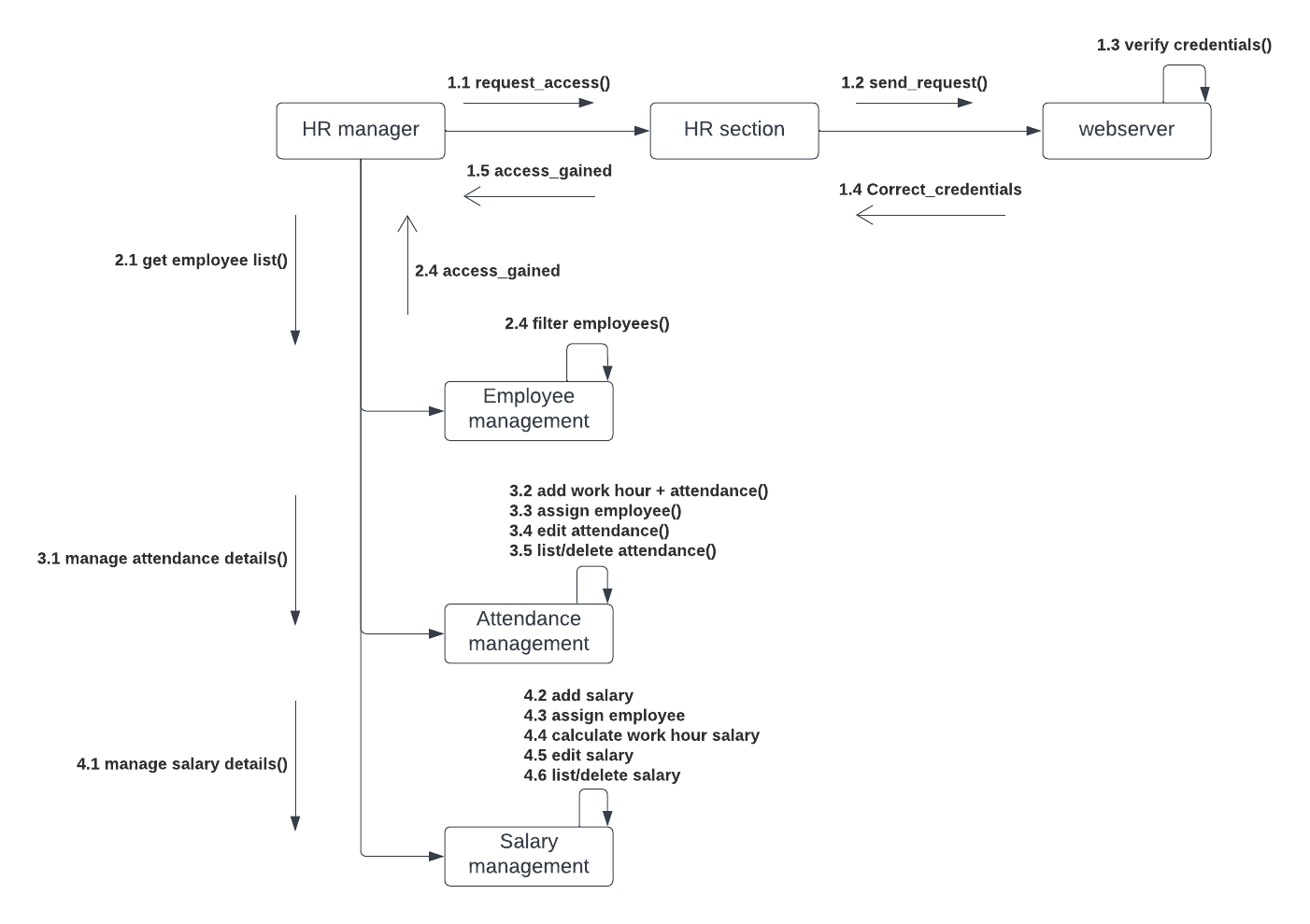
* + 1. Full Class Diagram



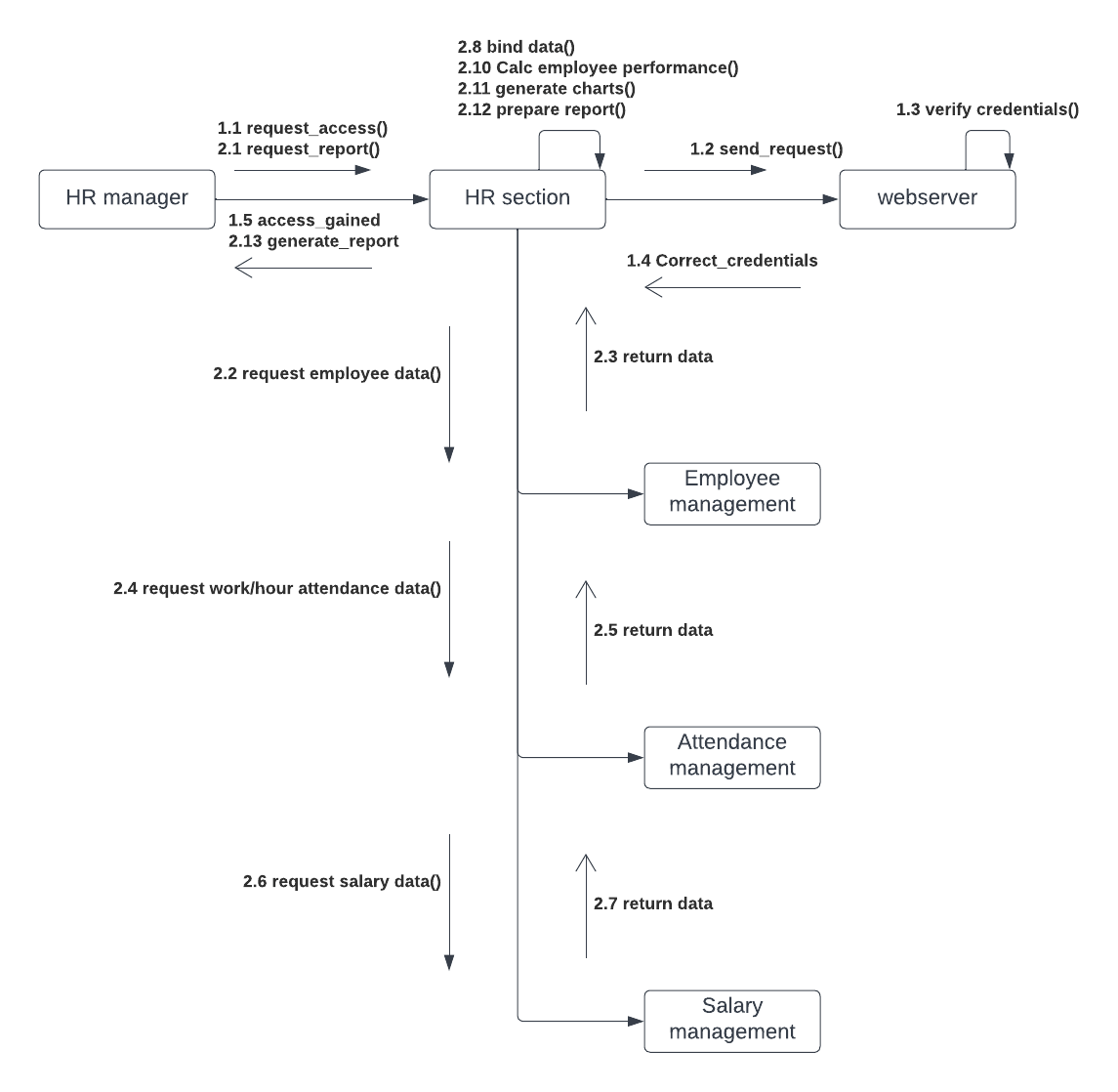
* 1. Collaboration Diagram
     1. Authentication



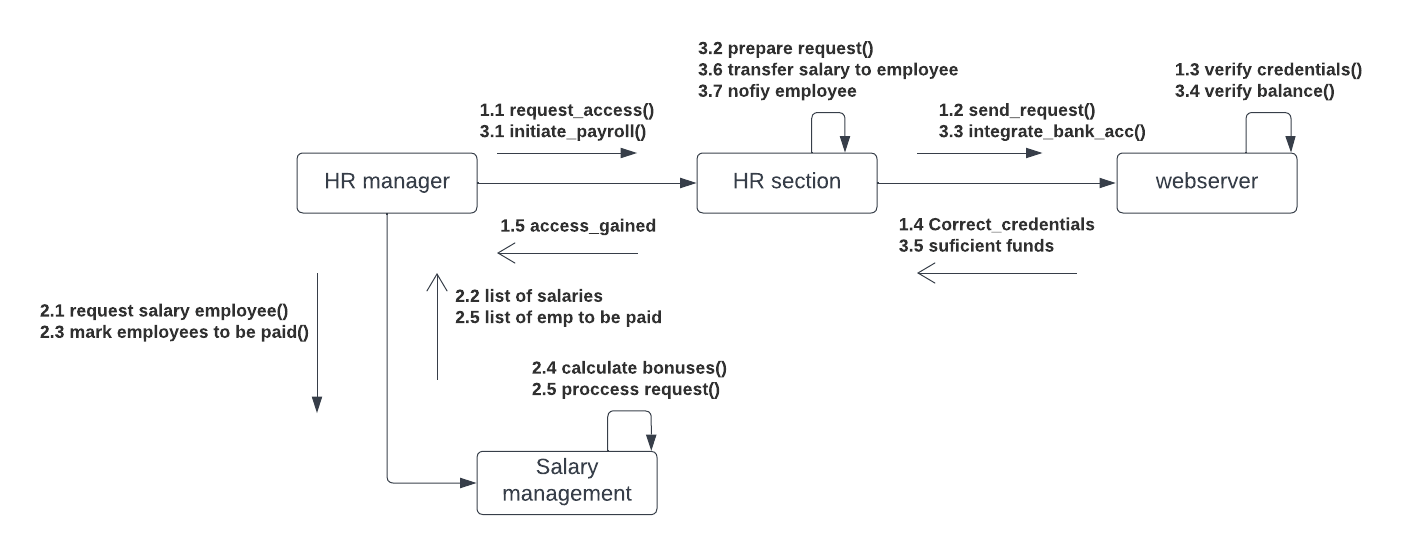
* + 1. Employee Management



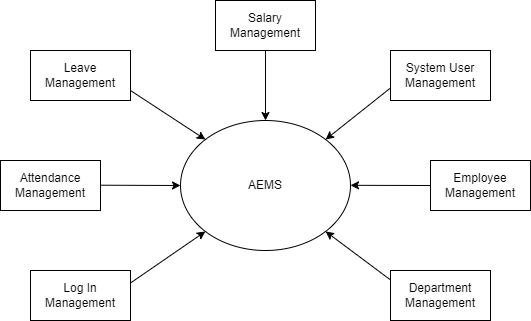
* + 1. Reports



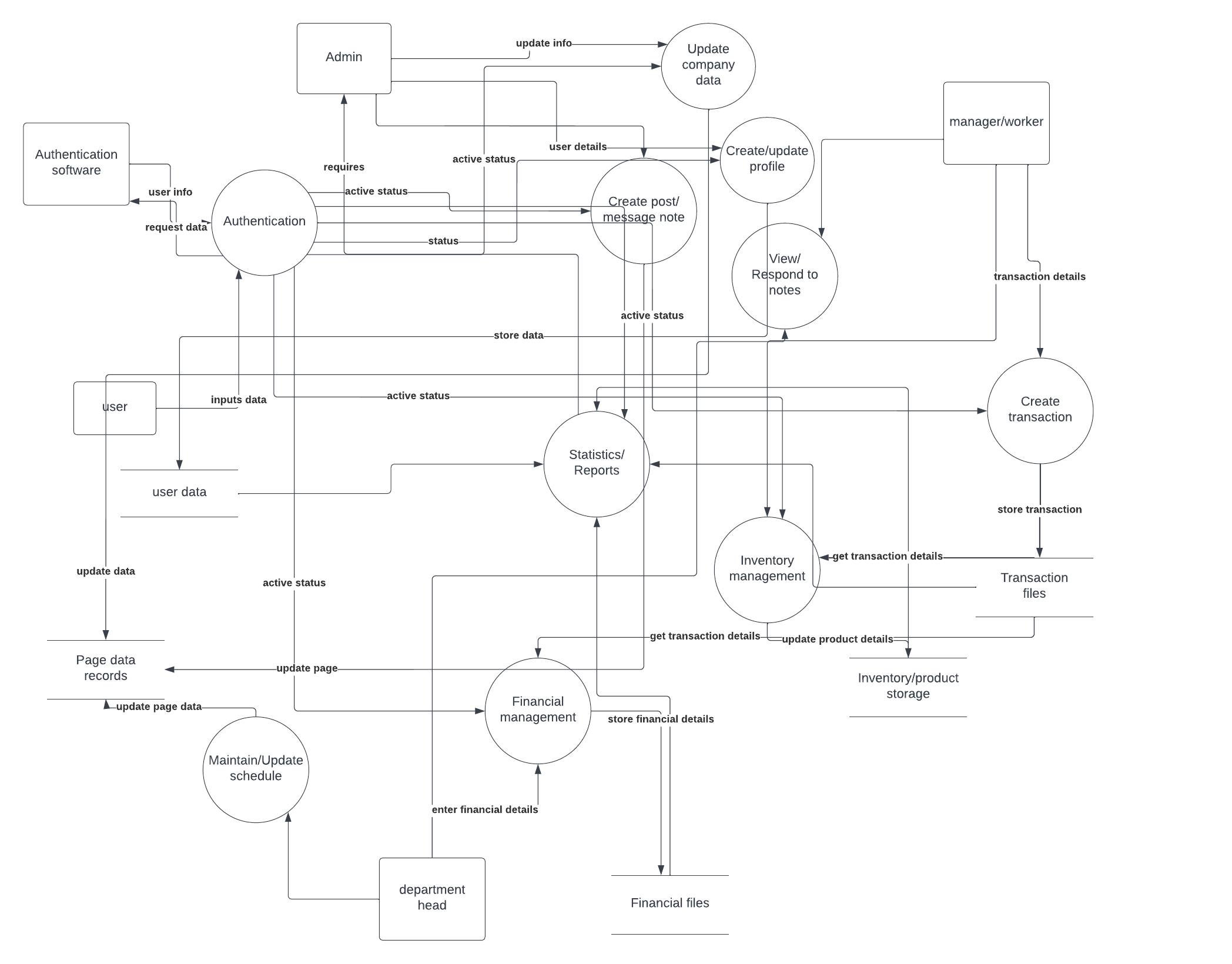
* + 1. Payrolls



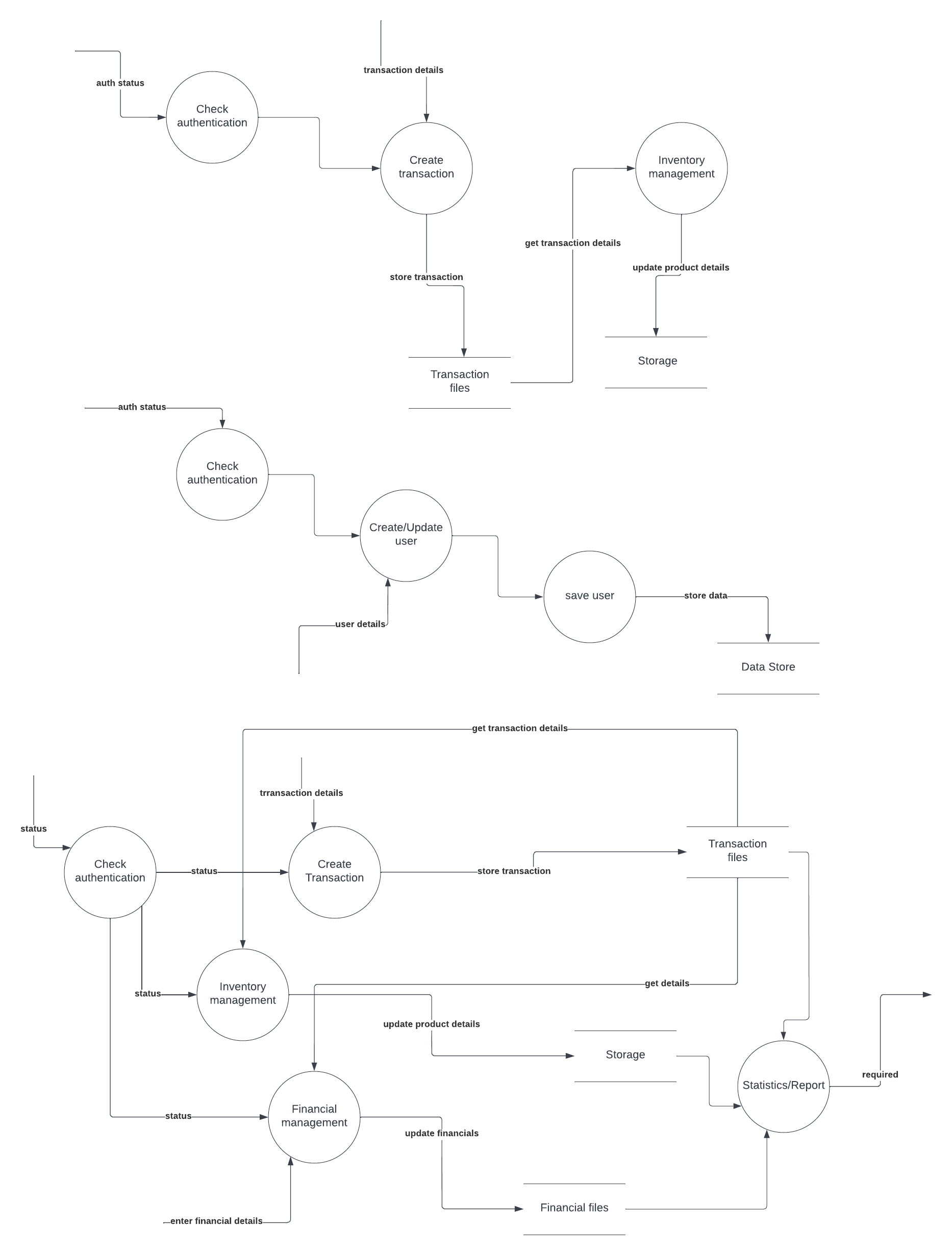
* 1. Data Flow Diagrams
     1. Level 0



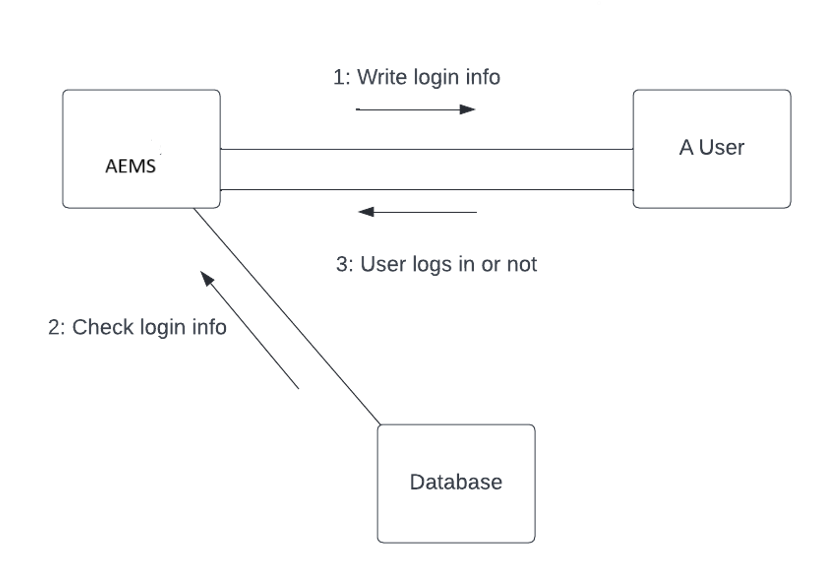
* + 1. Level 1



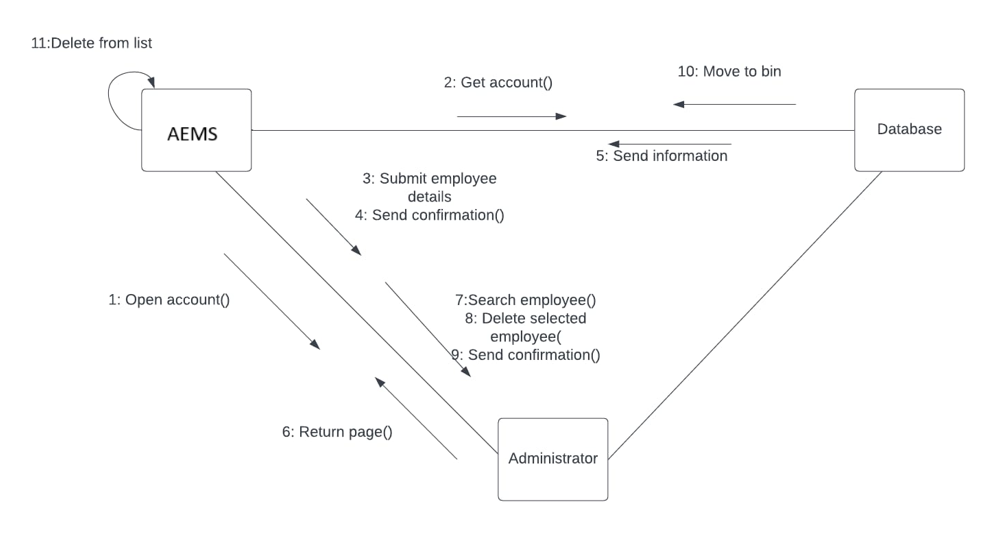
* + 1. Level 2

****

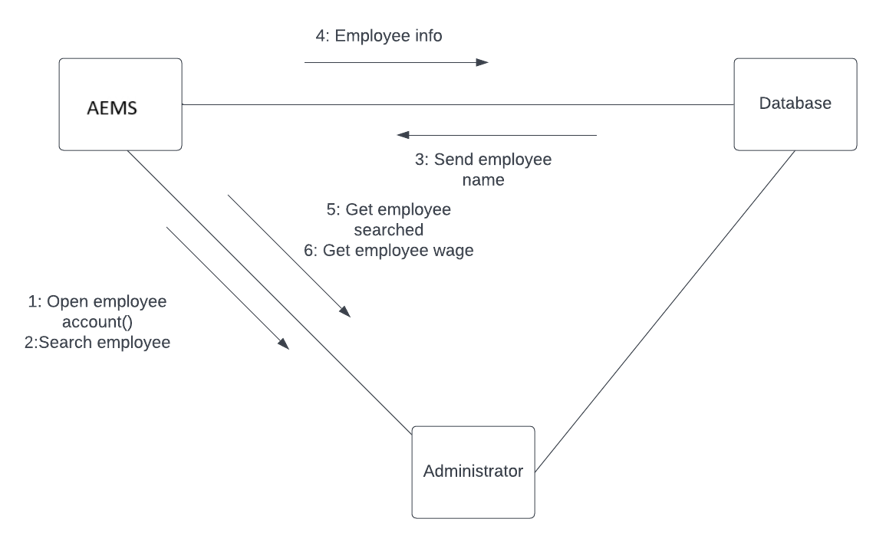
* 1. Communication Diagrams
     1. Log In



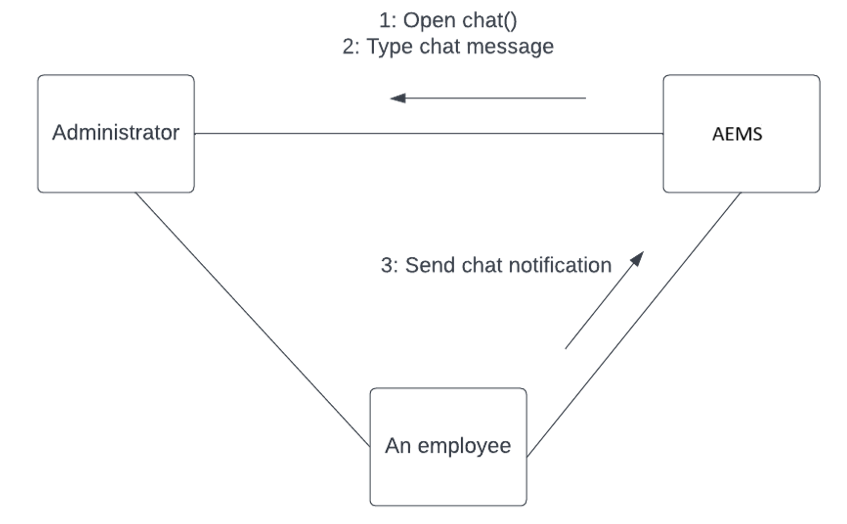
* + 1. Create or delete Employee



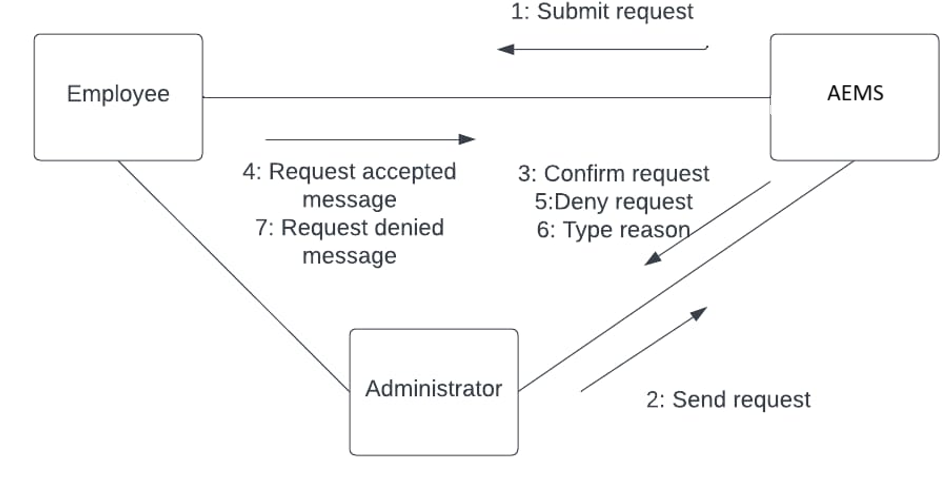
* + 1. Employee Information



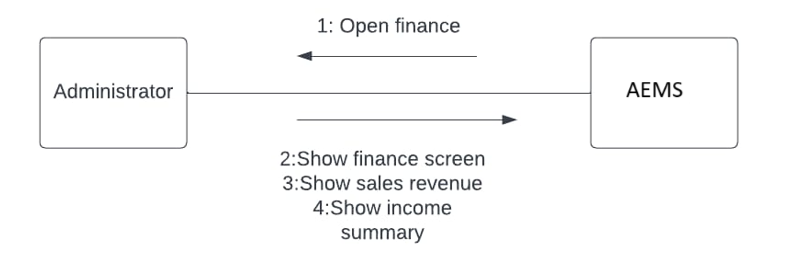
* + 1. Chat System



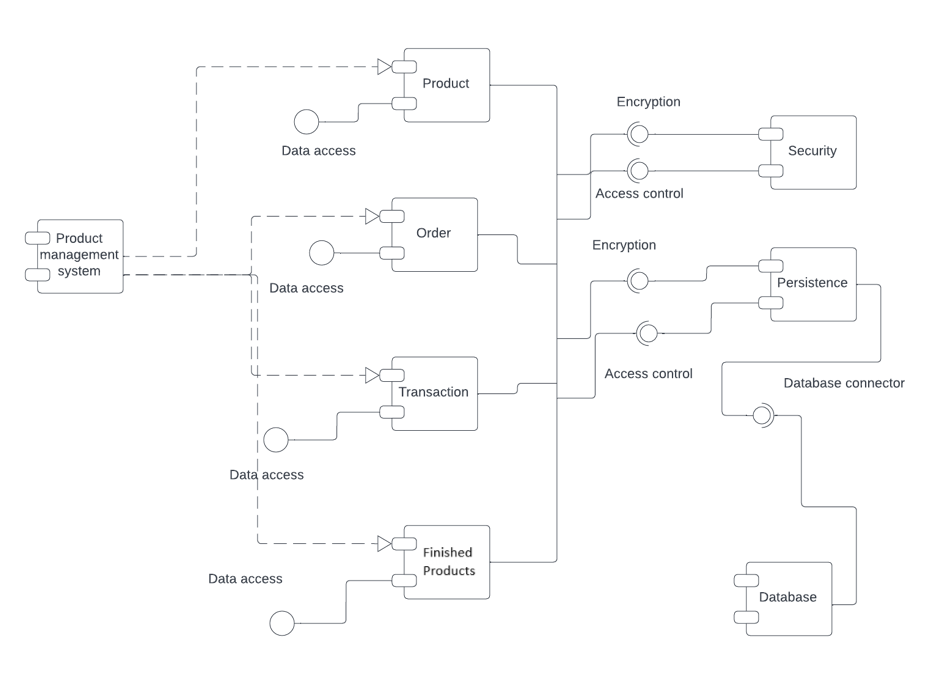
* + 1. Request Leaves



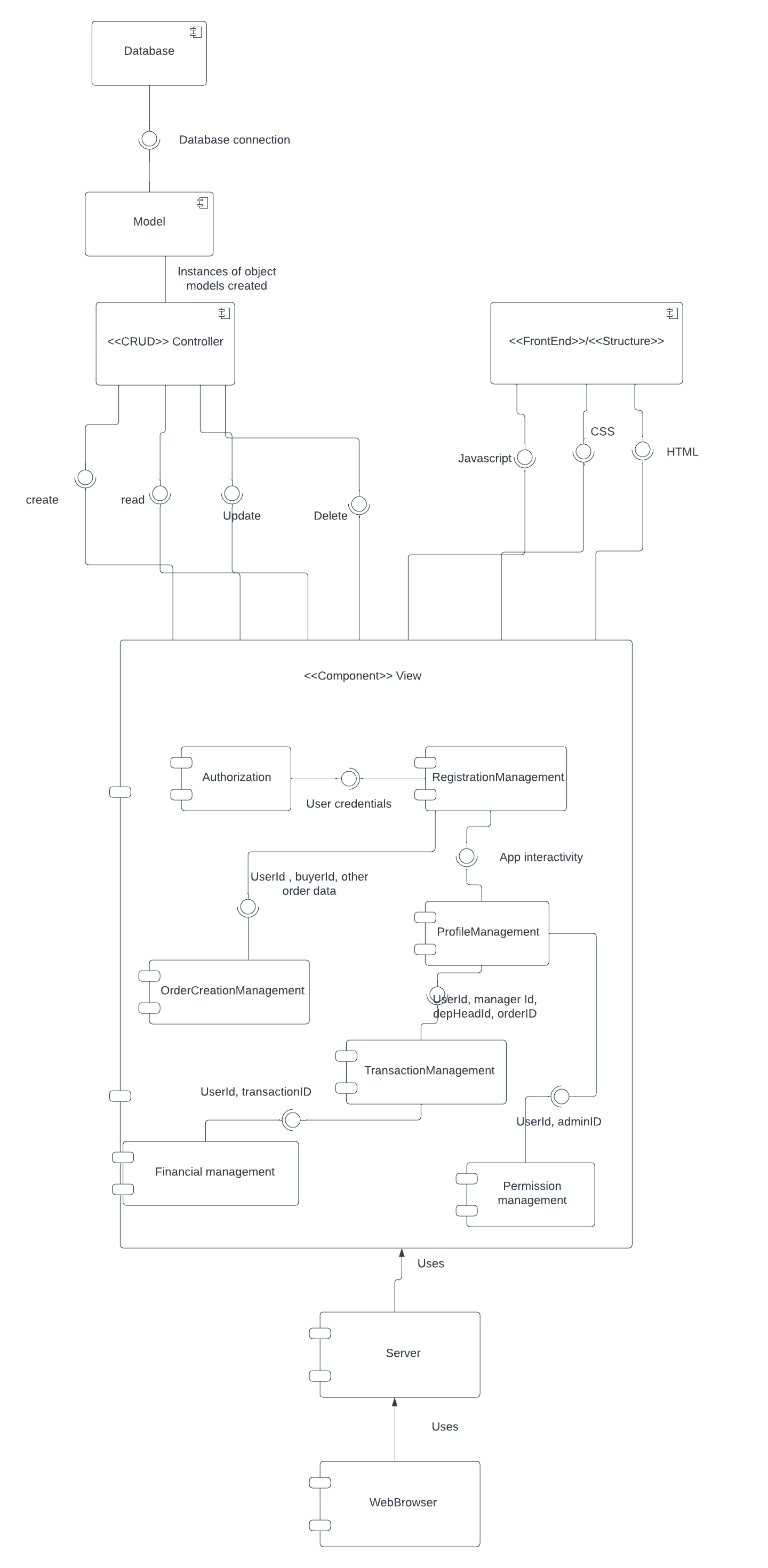
* + 1. Finance



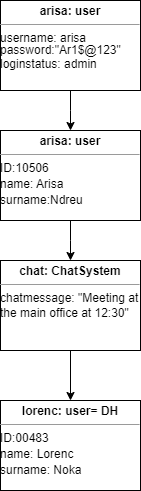
* 1. Component Diagrams
     1. Product Management



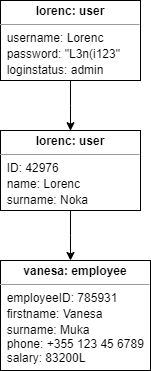
* + 1. Order Management

****

* 1. Object Diagrams
     1. Chat System



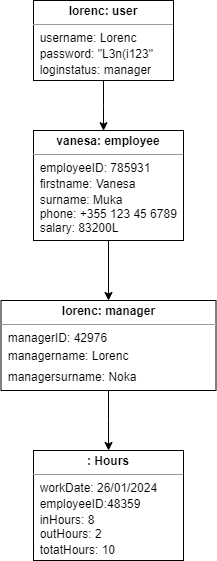
* + 1. Employee Handling



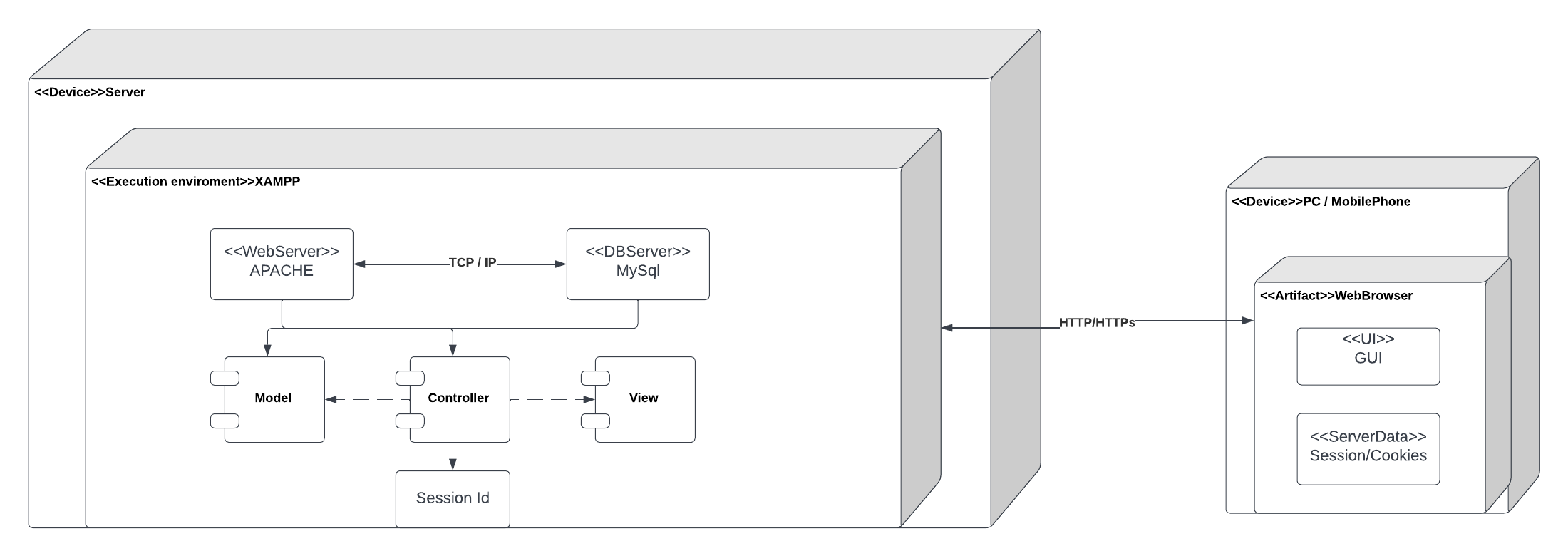
* + 1. Request Leave



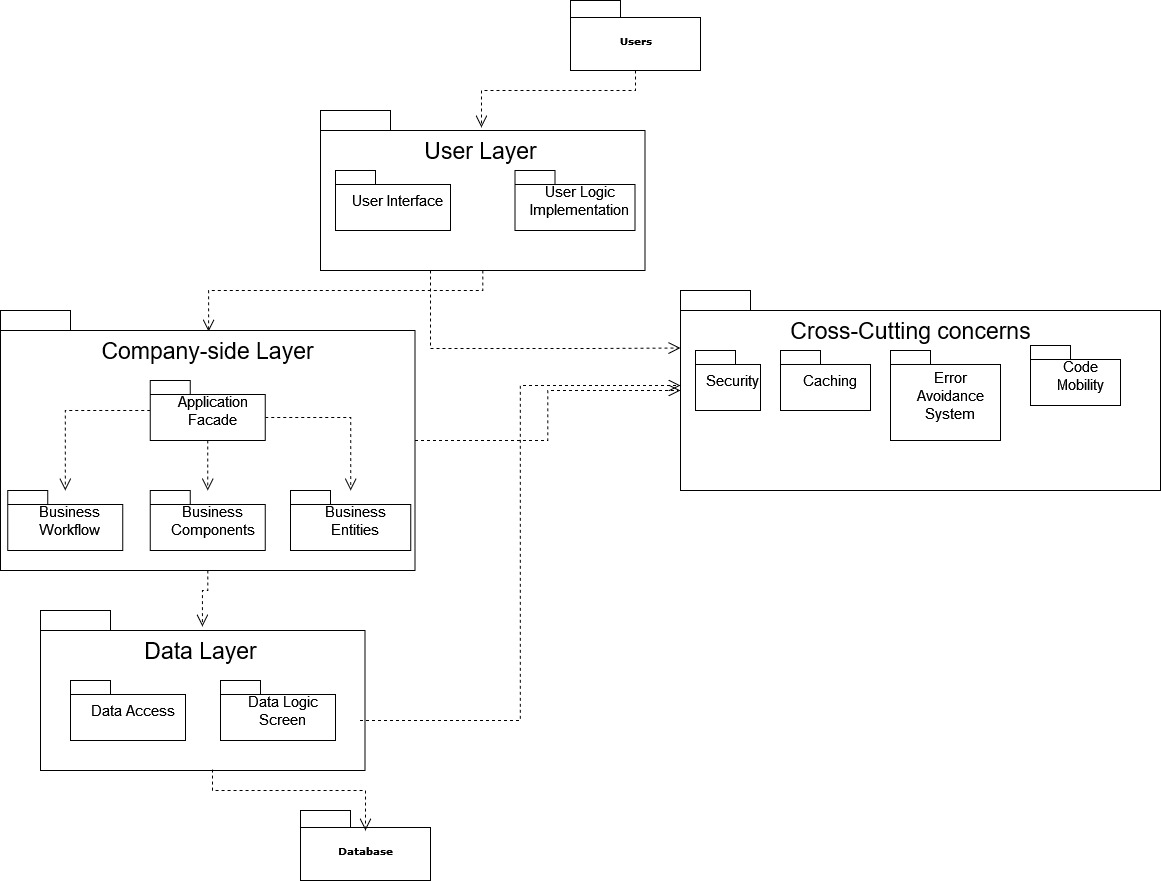
* + 1. Time Tracking



* 1. Deployment Diagrams

****

* 1. Package Diagram

****

1. Design Patterns

Singleton, Observer, and Template

In the class diagram, representing the overall structure, the design patterns of Singleton and Observer are identified, influencing the "chat system" entity. Given the requirement for a single common chat instance in the program, the Singleton design pattern is applied to ensure that only one instance of the chat class exists, preventing the initialization of additional instances. Additionally, the Observer design pattern is implemented for the chat, establishing a "subscriber" relationship between classes. In this context, whenever an announcement is made in the chat, all employees must be notified, leading to the creation of a chat reader interface implemented by concrete reader classes associated with users.

The Template method is employed in creating various user types, such as admin users, manager users, and employee users (further categorized into team leaders, basic workers, and economists). By utilizing the base user class and the employee class as templates, new classes can be generated with added features.

Mediator and Strategy

In the class diagram responsible for user login, the Mediator and Strategy design patterns are employed to ensure a user logs in and encounters the appropriate graphical user interface. The mediator interface validates user login information, checking if the entered data is acceptable. After successful login, the strategy design pattern distinguishes between different views based on user credentials, allowing the creation of admin, manager, and worker views.