Use Case Description

1. Use Case: Login as Admin

- **Primary Actor(s)**: Admin
- **Preconditions**: Admin must have valid credentials (email & password).
- **Trigger**: Admin attempts to access the system.
- **Main Flow**:
    1. Admin opens the login page.
    2. Admin enters his email (format: admin.name@company.com) and password.
    3. System triggers the "Verify login".
    4. If credentials are valid, the admin profile is accessed successfully.
- **Alternate Path**: None.
- **Exceptional Path**:
  1. Verification fails.
  2. System displays an error message for invalid credentials.
  3. Admin re-enters email and password.
  Post-conditions: Admin is redirected to his own dashboard.

2. Use Case: Login as Manager

- **Primary Actor(s)**: Manager
- **Preconditions**: Manager must have valid credentials (email & password).
- **Trigger**: Manager attempts to access the system.
- **Main Flow**:
    1. Manager opens the login page.
    2. Manager enters his email (format: department.manager.name@company.com) and password.
    3. System triggers the "Verify login".
    4. If credentials are valid, the Manager profile is accessed successfully.
- **Exceptional Path**:
  1. Verification fails.
  2. System displays an error message for invalid credentials.
  3. Manager re-enters email and password.
  Post-conditions: Manager is redirected to his own dashboard.

3. Use Case: Login as Employee

- **Primary Actor(s)**: Employee
- **Preconditions**: Employee must have valid credentials (ID & password)
- **Trigger**: Employee attempts to access the system.
- **Main Flow**:
  1. Employee opens the login page.
  2. Employee enters his ID and password.
  3. System triggers the "Verify login".
  4. If credentials are valid, the Employee profile is accessed successfully.
- **Exceptional Path**:
  1. Verification fails.
  2. System displays an error message for invalid credentials.
  3. Employee re-enters ID and password.

  Post-conditions: Employee is redirected to his own dashboard.

4. Use Case: Verify Login

- **Preconditions**:
  1. User (Admin, Manager, or Employee) has entered their credentials.
  2. The login attempt is initiated by the "Login" use case.
- **Trigger**: Login use case triggers it as included sub-use case.
- **Main Flow**:
  1. System receives the username and password from the "Login" use case.
  2. System verifies the credentials by checking them in the database.
  3. If the credentials are valid, the system returns a "successful login" response to the parent use case.
  4. If the credentials are invalid, the system extends to the "Show Error" use case.
- **Exceptional Path**: If there is a database connection error, the system may notify the parent use case that the login service is temporarily unavailable.
- **Post-conditions**: The system either grants access to the user's profile or triggers the "Show Error" use case if verification fails.

5. Use Case: Show Error

- **Preconditions**: The "Verify Login" use case has returned a failed verification result.
- **Trigger**: Triggered when the system detects an invalid username or password during login.
- **Main Flow**:
  1. System displays an error message, notifying the user of invalid credentials.
  2. System prompts the user to re-enter their login information.

- **Exceptional Paths**: If there are repeated failed login attempts (e.g., 3 consecutive attempts), the system may temporarily lock the account for security purposes and display a relevant message.
- **Post-conditions**: The user is informed of the login failure and can choose to retry the login process.

6. Use Case: Logout

- **Primary Actor(s)**: Admin, Manager, or Employee
- **Secondary Actor(s)**: None
- **Preconditions**: User must be logged in.
- **Trigger**: Actor clicks on 'Logout' button.
- **Main Flow**:
    1. User clicks on the 'Logout' button.
    2. User confirms by clicking 'Yes' to acknowledge the logout.
    3. The system logs the user out and redirects them to the login page.
- **Alternate Path**:
  - User clicks on 'Cancel' button.
  - The system does not execute the logout action, and the user remains on the current page.
  Post-conditions: User is directed to login page.

7. Use Case: Submit Employees Absence

- **Primary Actor(s)**: Manager.
- **Secondary Actor(s)**: None.
- **Preconditions**: Manager must be logged in.
- **Trigger**: Manager presses the "Mark Absence" button on the system interface.
- **Main Flow**:
    1. The system retrieves the list of employees under the manager's supervision.
    2. The manager checks the mark absence button to mark absence for employees who are absent.
    3. The system records the absent status for this employee.
- **Exceptional Path**:
  1. If the system fails to record the attendance, an error message is displayed.
  2. The manager is prompted to remark the absence.
  Post-conditions: Absence count increments and is recorded in the database.

8. Use Case: View Personal Data

- **Primary Actor(s)**: Admin, Manager, or Employee
- **Secondary Actor(s)**: None
- **Preconditions**: User must be logged in.
- **Trigger**: Actor clicks on the 'View Personal Data' tab.
- **Main Flow**:
    1. User clicks on 'View Personal Data' tab in profile.
    2. The system retrieves user data (first, last name , address, phone, job title and email) from the database.
    3. User information is displayed on the screen.
- **Exceptional:**
    1. The system encounters a database error or connection issue.
    2. System displays an error message.
- **Post-conditions**: Personal info is displayed on screen.

9. Use Case: Assign Tasks

- **Primary Actor(s)**: Manager
- **Secondary Actor(s)**: Employee
- **Preconditions**: Manager must be logged in.
- **Trigger**: Manager clicks on 'Assign Tasks' button.
- **Main Flow**:
    1. Manager fills out a form with task details (task id, title, project id, start and end date, description, employee ID/name) and submits it.
    2. System saves task details into database.
    3. System displays a confirmation message.
- **Alternate Path**:
    - Manager clicks on 'Cancel' button.
    - The system does not execute the assign task action.
- **Exceptional Paths**:

  - Path 1: Manager enters a non-existent ID/name so system displays an error message.

  - Path 2: Database error or connection issue system displays an error message.

  Post-conditions: Task is assigned to the specified employee.

10. Use Case: View Department's Employees' Data

- **Primary Actor(s):** Manager
- **Preconditions:** Manager must be logged in.
- **Trigger:** Manager clicks on the "View Employees Data" button.
- **Main Flow:**

    1. The system retrieves job details (employee id, name, phone number and job title) of employees in the department.
    2. Data is displayed in a table format.
- **Exceptional Path:**
    - If a database error or connection issue occurs, the system displays an error message.
- **Post-conditions:** Employee details are successfully displayed in a table.

11. Use Case: Request Vacation

- **Primary Actor(s):** Employee
- **Secondary Actor(s):** Manager
- **Preconditions:** Employee must be logged in.
- **Trigger:** Employee clicks on the "Request Vacation" button.
- **Main Flow:**
    1. Employee fills out a vacation form with details (e.g., employee ID, vacation type, duration, cause).
    2. The system saves the vacation request and notifies the manager.
    3. The system displays a confirmation message to the employee.
- **Alternate Path:**
    - If the employee clicks "Cancel," the system does not submit the vacation request.
- **Exceptional Path:**
    - If a database error or connection issue occurs, the system displays an error message.
- **Post-conditions:** The vacation request is sent to the manager.

12. Use Case: Generate Reports for Subordinates

- **Primary Actor(s):** Manager
- **Description:** Manager generates performance, attendance, and payroll reports for employees.
- **Preconditions:** Manager must be logged in.
- **Trigger:** Manager clicks on the "Generate Reports" button.
- **Main Flow:**
    1. Manager fills out a report form with employee details (e.g., ID, name, achievements, attendance, payroll).
    2. The system saves the report in the database.
    3. The system displays a confirmation message.
- **Alternate Path:**

    - If the manager clicks "Cancel," the system does not generate the report.
- **Exceptional Path:**
    - If an invalid employee ID is entered, the system displays an error message.
    - If a database error occurs, the system displays an error message.
- **Post-conditions:** The report is generated and stored in the database.

13. Use Case: Create Employee

- **Primary Actor(s):** Admin
- **Preconditions:**
    - Admin must be logged in.
    - Admin has permission to add employees.
- **Trigger:** Admin selects "Add Employee."
- **Main Flow:**
    1. Admin fills out employee details(fisrt name, last name, phone, ID, password, address, job title, DOB,role).
    2. The system validates and saves the new employee in the database.
- **Post-conditions:** The employee record is created.
- **Alternative Flows:**
    - If input is invalid, the system prompts for correction.
    - If the employee already exists, the system displays an error message.
- **Exceptional Path:**
    - If a database error occurs, the system displays an error message.

14. Use Case: Update Employee

- **Primary Actor(s):** Admin
- **Preconditions:**
    - Admin must be logged in.
    - Admin has permission to update employee records.
- **Trigger:** Admin selects "Update Employee."
- **Main Flow:**
    1. Admin selects an employee to update and modifies the details.
    2. The system validates the data and saves the updates in the database.
- **Post-conditions:** Employee record is updated.
- **Alternative Flows:**
    - If input is invalid, the system prompts for correction.
    - If the employee does not exist, the system displays an error message.

- **Exceptional Path:**
    - If a database error occurs, the system displays an error message.

15. Use Case: Delete Employee

- **Primary Actor(s)**: Admin
- **Preconditions**:
    - Admin must be logged in.
    - Admin has permission to delete employees.
- **Trigger**: Admin selects "Delete Employee."
  Main Flow:
  1. Admin selects the employee and presses on the delete button.
  2. The system deletes the employee record from the database.
- **Post-conditions**: Employee record is deleted.
- **Exceptional Path**:
    - If a database error occurs, the system displays an error message.

16. Use Case: Create Department

- **Primary Actor(s):** Admin
- **Preconditions:**
    - Admin must be logged in.
    - Admin has permission to create departments.
- **Trigger:** Admin selects "Create Department."
- **Main Flow:**
    1. Admin fills out department details (manager ID, name, location, number of employees).
    2. The system validates the data.
    3. The system saves the new department in the database.
- **Post-conditions:** A new department is created.
- **Alternative Flows:**
    - If input is invalid, the system prompts for correction.
    - If the department already exists, the system displays an error message.
- **Exceptional Path:**
    - If a database error occurs, the system displays an error message.

17. Use Case: Update Department

- **Primary Actor(s):** Admin

- **Preconditions:**
  - Admin must be logged in.
  - Admin has permission to update departments.
- **Trigger:** Admin selects "Update Department."
- **Main Flow:**
  1. Admin selects a department and updates its details.
  2. The system validates the data and saves the updates.
- **Post-conditions:** Department information is updated.
- **Alternative Flows:**
  - If input is invalid, the system prompts for correction.
  - If the department does not exist, the system displays an error message.
- **Exceptional Path:**
  - If a database error occurs, the system displays an error message.

18. Use Case: Delete Department

- **Primary Actor(s):** Admin
- **Preconditions:**
  - Admin must be logged in.
  - Admin has permission to delete departments.
- **Trigger:** Admin selects "Delete Department."
- **Main Flow:**
  1. Admin confirms the deletion of a department.
  2. The system deletes the department from the database.
- **Post-conditions:** Department is deleted.
- **Alternative Flows:**
  - If the department does not exist, the system displays an error message.
- **Exceptional Path:**
  - If a database error occurs, the system displays an error message.

19. Use Case: View Assigned Tasks

- **Primary Actor(s):** Employee
- **Preconditions:** Employee must be logged in.
- **Trigger:** Employee clicks on "View Tasks" button.
- **Main Flow:**
  1. The system retrieves task data assigned to the employee from the database.
  2. The system displays the task details and their current status.

- **Post-conditions:** Task details are displayed to the employee.
- **Alternative Flows:**
    - If no tasks are assigned, the system informs the employee.
- **Exceptional Path:**
    - If a database error occurs, the system displays an error message.

20. Use Case: View Requested Vacation

- **Primary Actor(s):** Employee
- **Preconditions:** Employee must be logged in.
- **Trigger:** Employee clicks on "View Vacation Requests" button.
- **Main Flow:**
    1. The system retrieves vacation request data from the database.
    2. The system displays the vacation details and their status.
- **Post-conditions:** Vacation request details are displayed.
- **Alternative Flows:**
    - If no vacation requests exist, the system informs the employee.
- **Exceptional Path:**
    - If a database error occurs, the system displays an error message.

21. Use Case: Add Manager

- **Primary Actor(s):** Admin
- **Preconditions:**
    - Admin must be logged in.
    - Admin has permission to add managers.
- **Trigger:** Admin selects "Add Manager."
- **Main Flow:**
    1. Admin fills out manager details (first name, last name, phone, ID, password, address, job title, date of birth, role).
    2. The system validates the data.
    3. The system saves the manager details in the database.
- **Post-conditions:** Manager record is created.
- **Alternative Flows:**
    - If input is invalid, the system prompts for correction.
    - If the manager already exists, the system displays an error message.
- **Exceptional Path:**
    - If a database error occurs, the system displays an error message.

22. Use Case: Update Manager

- **Primary Actor(s):** Admin
- **Preconditions:**
  - Admin must be logged in.
  - Admin has permission to update manager records.
- **Trigger:** Admin selects "Update Manager."
- **Main Flow:**
  1. Admin selects a manager and updates their details.
  2. The system validates the data and saves the updates.
- **Post-conditions:** Manager record is updated.
- **Alternative Flows:**
  - If input is invalid, the system prompts for correction.
  - If the manager does not exist, the system displays an error message.
- **Exceptional Path:**
  - If a database error occurs, the system displays an error message.

23. Use Case: Delete Manager

- **Primary Actor(s):** Admin
- **Preconditions:**
  - Admin must be logged in.
  - Admin has permission to delete managers.
- **Trigger:** Admin selects "Delete Manager."
- **Main Flow:**
  1. Admin selects the manager and presses the delete button.
  2. The system deletes the manager record from the database.
- **Post-conditions:** Manager record is deleted.
- **Exceptional Path:**
  - If a database error occurs, the system displays an error message.