

**Requirement & Design Specification**

**Workforce Management Application (WMA)**

**Version: 1.0**

| **PRM392\_SE1831\_NJ\_G7** | |
| --- | --- |
| **Group Members** | Nguyễn Quang Tú - HE150968  Nguyễn Thanh Hải - HE172169  Phạm Xuân Chinh - HE171177  Đinh Thái Huy - HE171205 |
| **Supervisor/Lecturer** | Nguyễn Thị Hải Năng |

– Hanoi, May 2025 –

# Record of Changes

| **Version** | **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- | --- |
| V1.0 | 15/5 | A |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

\*A - Added M - Modified D - Deleted

Contents

[**Record of Changes**](#_u35rft3igs8t) **2**

[**I. Overview**](#_nan1dhmtpdtb) **5**

[1. System Context](#_8sv9hkr2fzmh) 5

[2. User Requirements](#_4rshruvsbv7o) 5

[2.1 Actors](#_9catcub0q07c) 5

[2.2 Diagrams](#_nv1e8u44823u) 6

[2.3 Descriptions](#_955upnsufgif) 10

[3. System Functionalities](#_6kq42kne3dmb) 14

[3.1 Screens Flow](#_754s53s9ty3c) 14

[3.1.1: Employee](#_sf1mi3ipqkqp) 15

[3.1.2: Admin System](#_1juunqtolg6) 15

[3.1.3: Department Manager](#_atdz837oym07) 16

[3.1.4: HR Manager](#_84pq3f6rw9d0) 16

[3.2 Screen Authorization](#_3o09ihu2ck4m) 17

[3.3 Non-UI Functions](#_uxdb732z1han) 20

[**II. Functional Requirements**](#_ovfkntt2abvr) **21**

[1. User Authentication](#_uwmywjnoihej) 21

[1.1 Login](#_wzve1kc9bed) 21

[1.2 Forgot password](#_sbfok6gzo4bn) 24

[1.3 User Profile](#_8t50s02c9y8z) 29

[2. System Administration](#_wyllje292q0t) 34

[2.1 Admin Dashboard](#_iq6b6e95bd49) 34

[2.2 User Management](#_ecq9484nc262) 39

[2.2.1 User List](#) 39

[2.2.2 User Details](#) 42

[2.2.3 Create New User](#_iuygslytm2jx) 45

[2.3 Department Management](#_3850eny1azbk) 48

[2.3.1 Department List](#_vrpgygxaelba) 48

[2.3.2 Department Details](#_fre66bvfikq2) 51

[2.3.2 Create Department](#_8m5lqv9on38q) 53

[2.4 Task Management](#_pu31ry4dhgy3) 56

[2.4.1 Task List](#_ocy403q3g5i0) 57

[2.4.2 Task Details](#_238xq2sairu7) 60

[2.4.3 Create Task](#_jh3mexz69y6o) 63

[3. System HR](#_13n9dxv5s72o) 66

[4. System Dep\_Manager](#_oz1ys2e5515a) 67

[5. System Employee](#_3p3po6lqdj5z) 68

[III. System Design](#_2hzq4umnx7e3) 68

[1. Database Design](#_lhmy7udlvgfj) 68

[1.1 Database Schema](#_53r38il6zcra) 69

[1.2 Table Descriptions](#_5abwfskndilf) 70

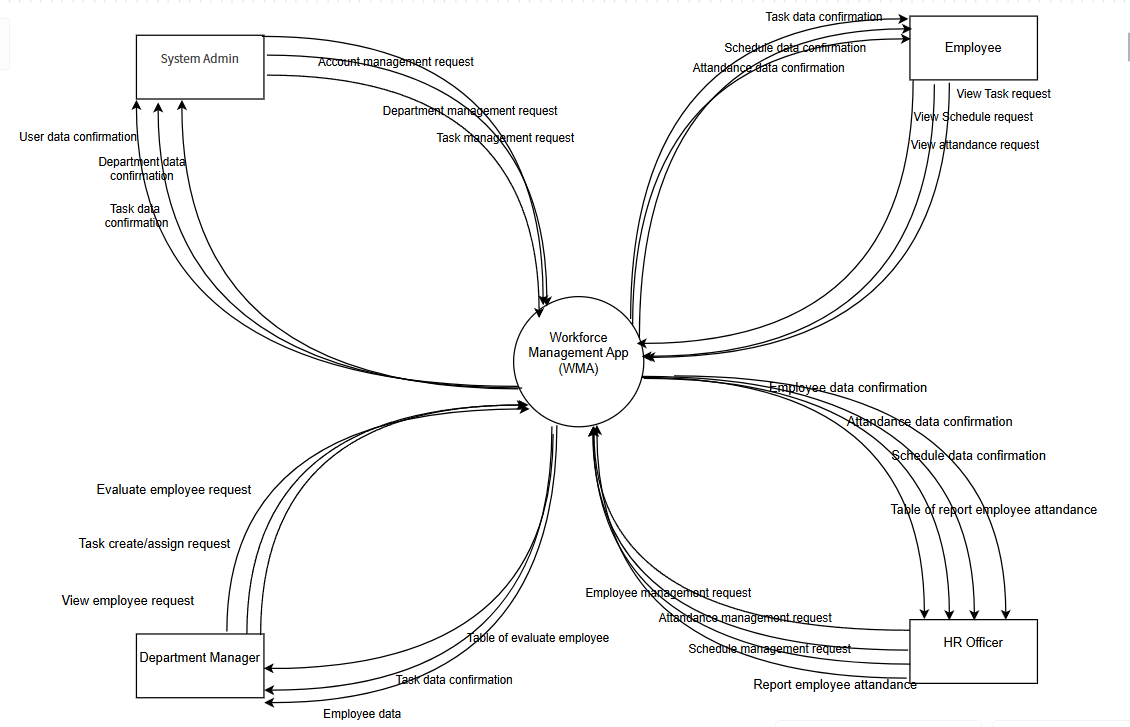
[2. Code Packages](#_jix0p8imcl50) 72

[2.1 Package Diagram](#_te3m3xzd1vfw) 72

[2.2 Package Descriptions](#_562hcphpjp79) 73

# I. Overview

## 1. System Context

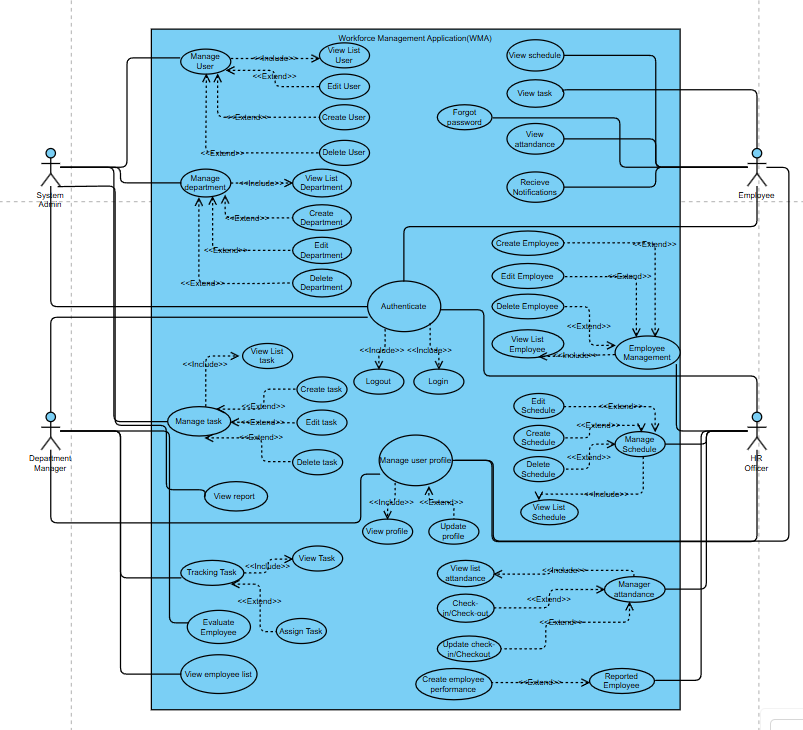


## 2. User Requirements

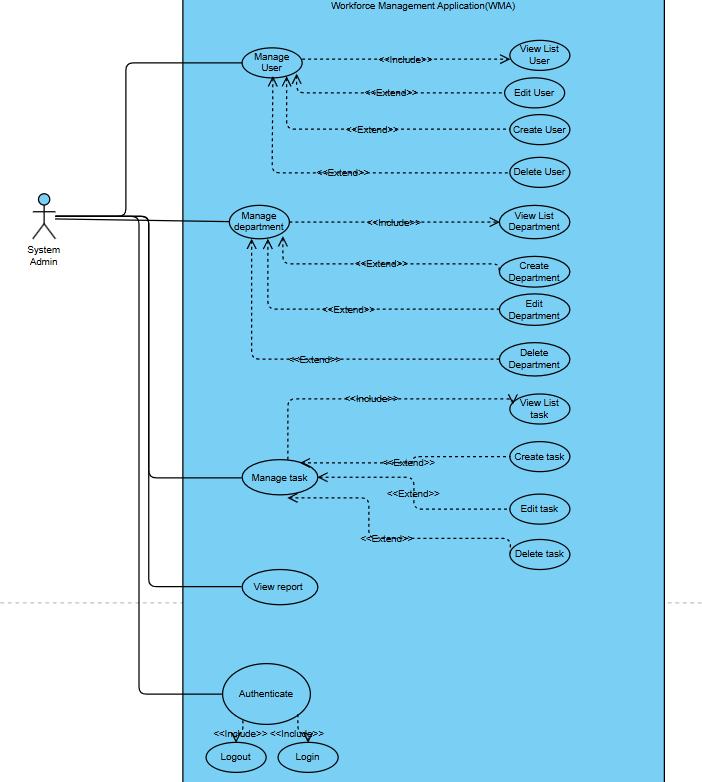
### 2.1 Actors

| **#** | **Actor** | **Description** |
| --- | --- | --- |
| 1 | System Admin | Responsible for overall system management, including user account management, department structuring, role assignments, and overseeing system configurations. |
| 2 | Employee | Regular user who interacts with the system to view tasks, submit attendance, access schedules, and receive notifications. |
| 3 | Department Manager | Oversees a specific department or team, assigns tasks, evaluates employee performance, and monitors team progress. |
| 4 | HR Officer | Manages employee records and schedule, handles recruitment processes, oversees attendance records, and ensures compliance with HR policies. |

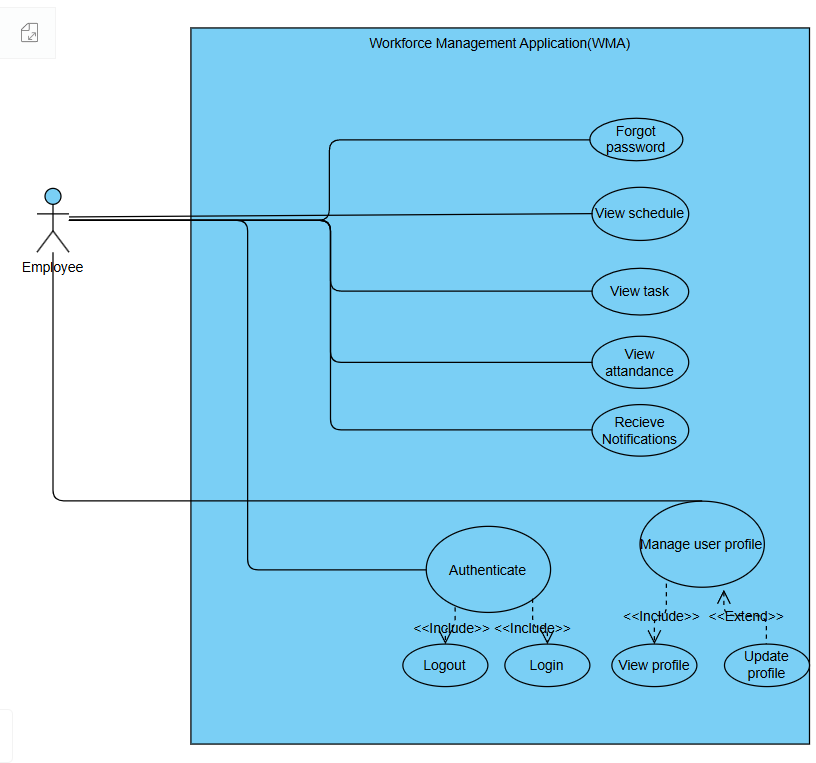
### 2.2 Diagrams

**

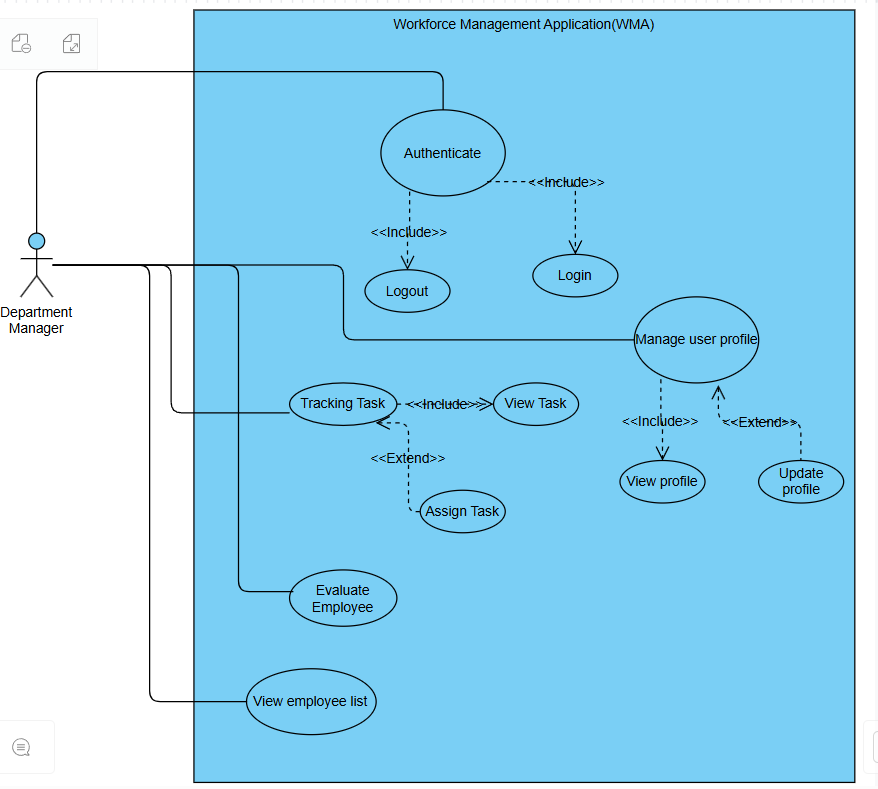
1. *Admin sysytem*

**

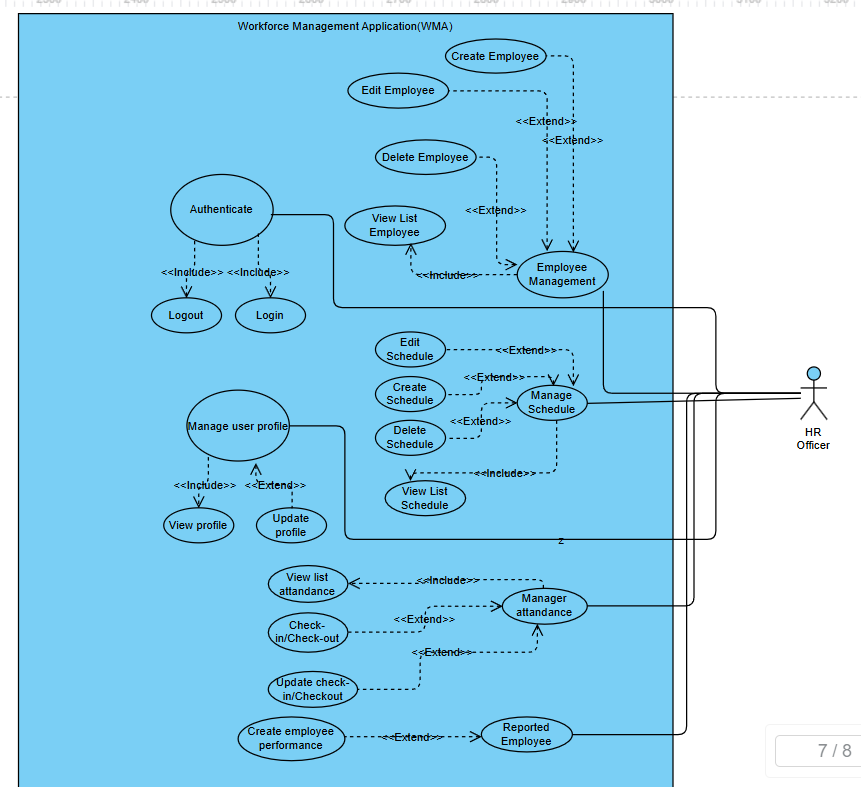
1. *Employee*

**

1. *Department Manager*

**

1. *HR Manager*

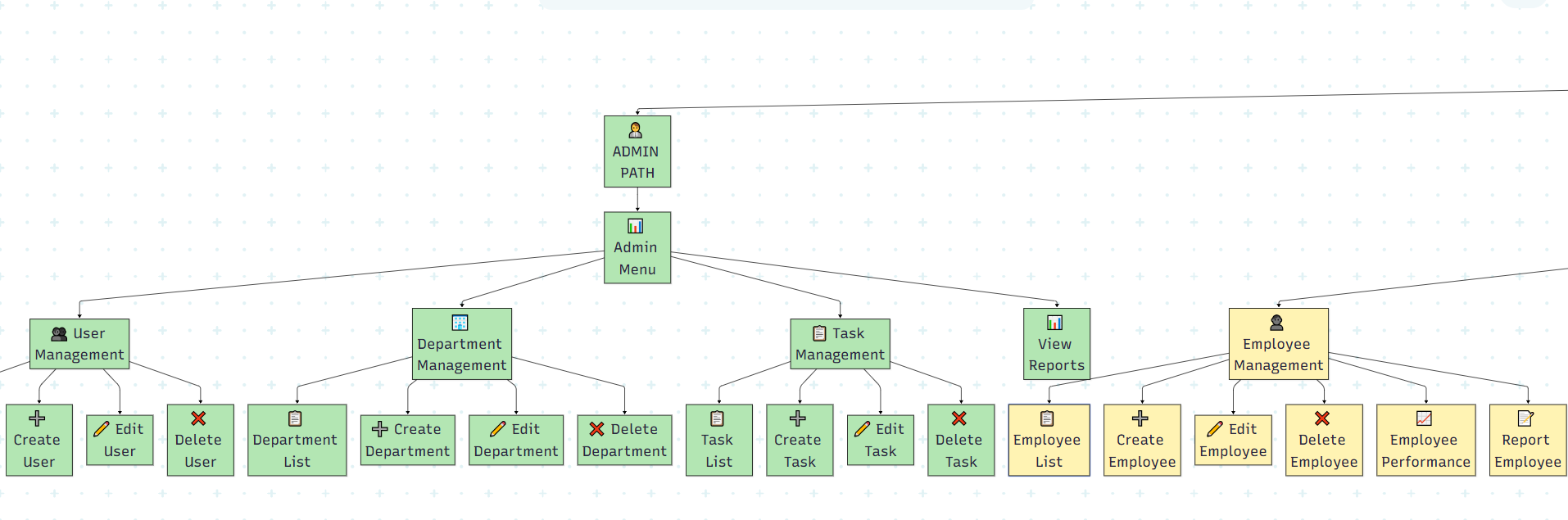
**

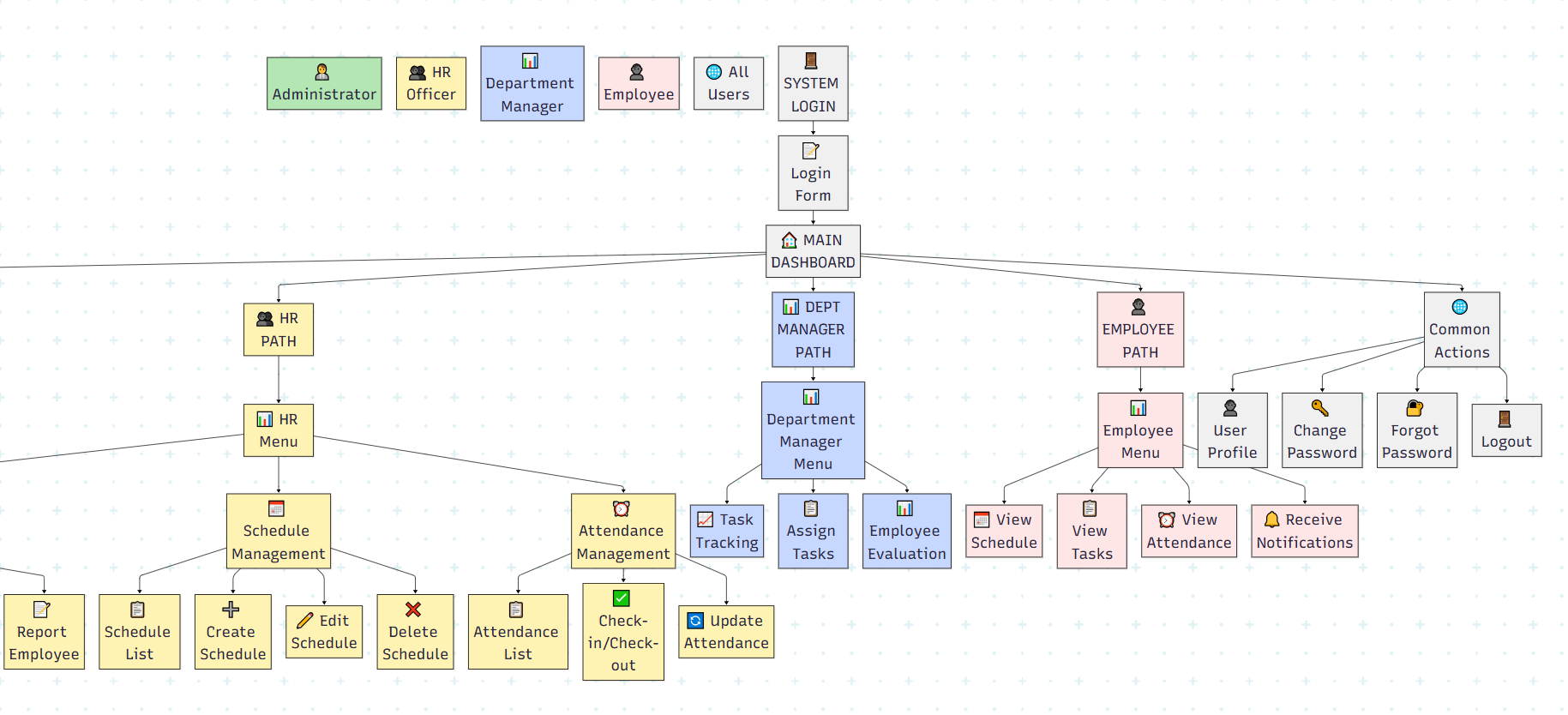
### 2.3 Descriptions

| **ID** | **Use Case** | **Actor(s)** | **Use Case Description** |
| --- | --- | --- | --- |
| UC01 | Authenticate | All Actors | Allows all users to authenticate with the system using their credentials before accessing features. |
| UC02 | Login | All Actors | Enables users to log into the system after successful authentication. |
| UC03 | Logout | All Actors | Logs the user out of the system and ends the current session securely. |
| UC04 | Forgot Password | All Actors | Allows users to reset their password if they forget it, via verification steps. |
| UC05 | Manage User Profile | All Actors | Enables users to view and update their personal information. |
| UC06 | View Profile | All Actors | Displays the profile details of the currently logged-in user. |
| UC07 | Update Profile | All Actors | Allows editing and saving updates to user profile information. |
| UC08 | Manage User | System Admin | Administers system user accounts, including creating, editing, viewing, and deleting users. |
| UC09 | View List User | System Admin | Displays a list of all users registered in the system. |
| UC10 | Create User | System Admin | Adds a new user account with required credentials and role. |
| UC11 | Edit User | System Admin | Modifies user account details such as role, name, or credentials. |
| UC12 | Delete User | System Admin | Deletes a user from the system, revoking their access. |
| UC13 | Manage Department | System Admin | Handles operations related to departments including creation, editing, and deletion. |
| UC14 | View List Department | System Admin | Lists all existing departments within the system. |
| UC15 | Create Department | System Admin | Adds a new department to the system. |
| UC16 | Edit Department | System Admin | Updates details of an existing department. |
| UC17 | Delete Department | System Admin | Removes a department and its related data. |
| UC18 | Manage Task | System Admin | Administers tasks for departments or employees including creation, editing, viewing, and deletion. |
| UC19 | View List Task | System Admin | Displays all created tasks across the system. |
| UC20 | Create Task | System Admin | Creates a new task and assigns it to a department or user. |
| UC21 | Edit Task | System Admin | Updates task information including description or assigned person. |
| UC22 | Delete Task | System Admin | Deletes an existing task from the system. |
| UC23 | View Report | Department Manager | Allows the department manager to view reports generated from tasks, schedules, or employee data. |
| UC24 | Track Task | Department Manager | Allows tracking of the status or progress of assigned tasks within the department. |
| UC25 | Assign Task | Department Manager | Assigns created tasks to specific employees. |
| UC26 | Evaluate Employee | Department Manager | Evaluates employee performance based on tracked tasks and behavior. |
| UC27 | View Employee List | Department Manager | Displays a list of all employees within the department for tracking or evaluation purposes. |
| UC28 | Employee Management | Officer | Handles the creation, editing, viewing, and deletion of employee records. |
| UC29 | View List Employee | Officer | Displays all employees managed by the Officer. |
| UC30 | Create Employee | Officer | Adds a new employee record to the system. |
| UC31 | Edit Employee | Officer | Updates information about an existing employee. |
| UC32 | Delete Employee | Officer | Deletes an employee record from the system. |
| UC33 | Manage Schedule | Officer | Manages schedule creation, editing, and deletion for employees. |
| UC34 | View List Schedule | Officer | Lists all schedules managed by the officer. |
| UC35 | Create Schedule | Officer | Adds a new work schedule for employees. |
| UC36 | Edit Schedule | Officer | Updates an existing schedule. |
| UC37 | Delete Schedule | Officer | Removes a schedule entry. |
| UC38 | Manage Attendance | Officer | Handles attendance operations such as check-in/out and editing logs. |
| UC39 | View List Attendance | Officer | Displays attendance logs for all employees. |
| UC40 | Check-in/Check-out | Officer | Records employees’ arrival and departure times. |
| UC41 | Update Check-in/Check-out | Officer | Allows corrections or updates to check-in/out records. |
| UC42 | Report Employee | Officer | Allows reporting of employee activities or issues to upper management. |
| UC43 | Create Employee Performance | Officer | Records or evaluates an employee’s performance in the system. |
| UC44 | View Schedule | Employee | Allows employees to see their assigned work schedule. |
| UC45 | View Task | Employee | Allows employees to view tasks assigned to them. |
| UC46 | View Attendance | Employee | Enables employees to check their own attendance history. |
| UC47 | Receive Notifications | Employee | Receives internal alerts and notifications regarding schedules, tasks, or messages. |

## 3. System Functionalities

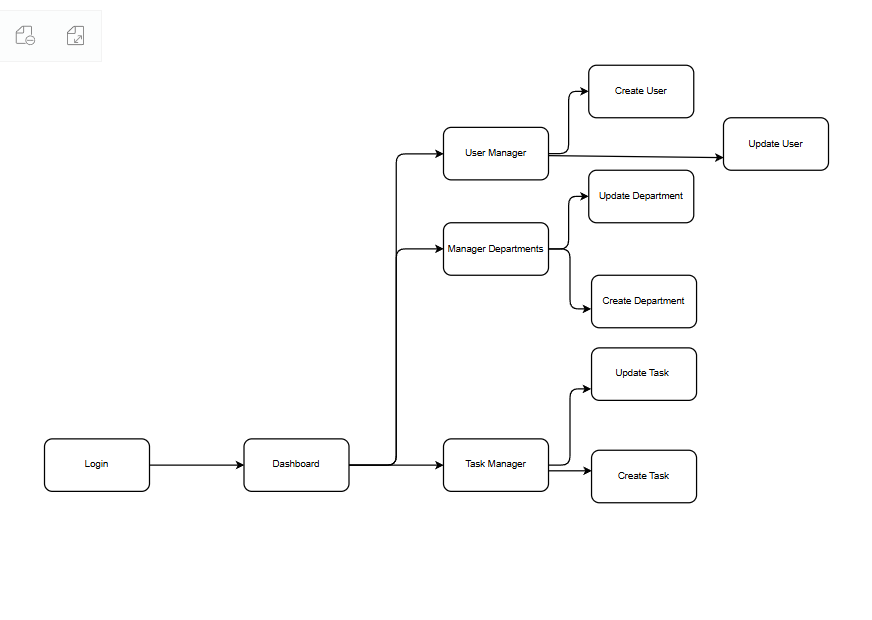
### 3.1 Screens Flow



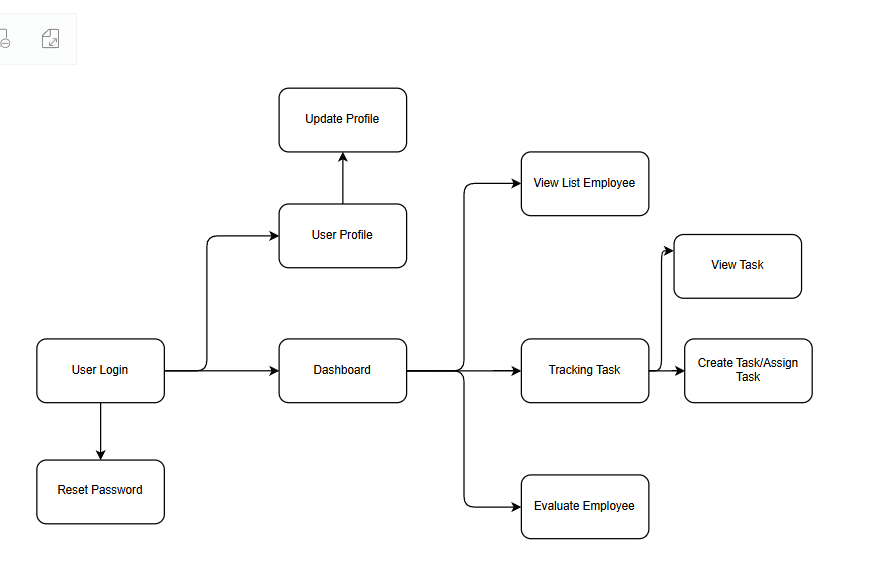


#### 3.1.1: Employee

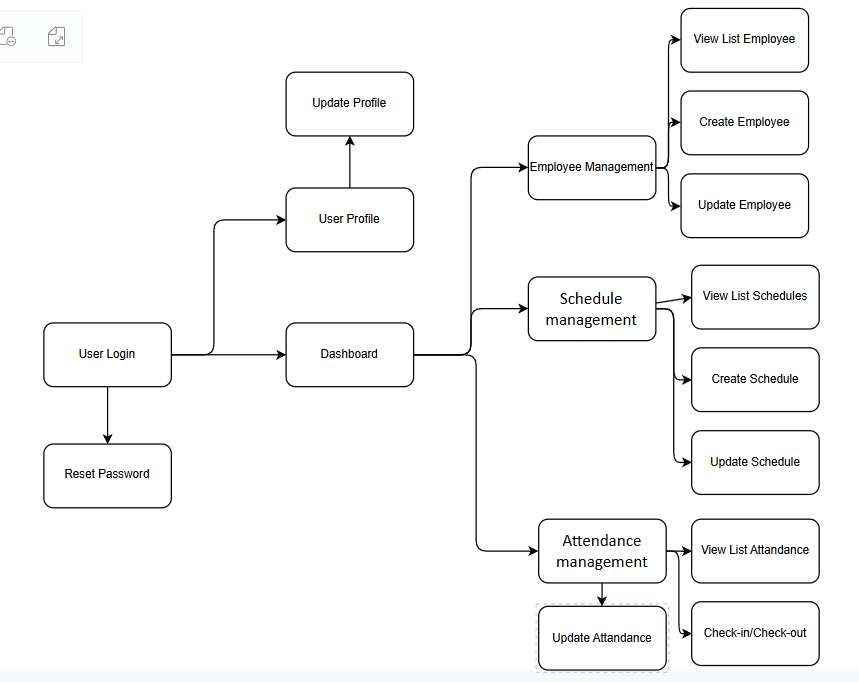
#### 3.1.2: Admin System



#### 3.1.3: Department Manager



#### 3.1.4: HR Manager



### 3.2 Screen Authorization

| **Screen / Module** | **Activity / Function** | **Administrator** | **HR Officer** | **Department Manager** | **Employee** |
| --- | --- | --- | --- | --- | --- |
| **Account and Permission Module** |  |  |  |  |  |
| Login | Login form | X | X | X | X |
| Logout | Logout action | X | X | X | X |
| Forgot Password | Password reset | X | X | X | X |
| User Profile | View and update profile | X | X | X | X |
| Manage Users | User management screen | X |  |  |  |
| User List | View list of users | X |  |  |  |
| Create User | Add new user | X |  |  |  |
| Edit User | Edit existing user | X |  |  |  |
| Delete User | Delete user | X |  |  |  |
| Manage Departments | Department management screen | X |  |  |  |
| Department List | List of departments | X |  |  |  |
| Create Department | Add new department | X |  |  |  |
| Edit Department | Edit department | X |  |  |  |
| Delete Department | Delete department | X |  |  |  |
| Manage Tasks | Task management screen | X |  |  |  |
| Task List | View all tasks | X |  |  |  |
| Create Task | Create task | X |  |  |  |
| Edit Task | Edit task | X |  |  |  |
| Delete Task | Delete task | X |  |  |  |
| **Reports** |  |  |  |  |  |
| View Reports | View reports | X |  |  |  |
| Task Tracking | Track task status |  |  | X |  |
| Assign Tasks | Assign task to employees |  |  | X |  |
| Employee Evaluation | Evaluate employee |  |  | X |  |
| **Employee Management** |  |  | X |  |  |
| Employee List | View employee list |  | X |  |  |
| Create Employee | Add employee |  | X |  |  |
| Edit Employee | Edit employee |  | X |  |  |
| Delete Employee | Delete employee |  | X |  |  |
| Manage Schedule | Schedule management |  | X |  |  |
| Schedule List | View schedules |  | X |  |  |
| Create Schedule | Add schedule |  | X |  |  |
| Edit Schedule | Edit schedule |  | X |  |  |
| Delete Schedule | Delete schedule |  | X |  |  |
| Manage Attendance | Attendance management |  | X |  |  |
| Attendance List | View attendance records |  | X |  |  |
| Check-in / Check-out | Record attendance |  | X |  |  |
| Update Attendance | Update attendance records |  | X |  |  |
| Report Employee | Report employee issues |  | X |  |  |
| Employee Performance | Record employee performance |  | X |  |  |
| **Employee Views** |  |  |  |  |  |
| View Schedule | View own schedule |  |  |  | X |
| View Task | View assigned tasks |  |  |  | X |
| View Attendance | View own attendance |  |  |  | X |
| Receive Notifications | Receive alerts |  |  |  | X |

### 

### 

### 

### 

### 

### 3.3 Non-UI Functions

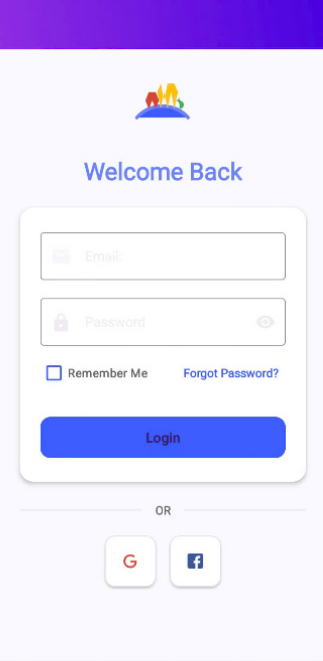
| **#** | **Feature** | **System Function** | **Description** |
| --- | --- | --- | --- |
| 1 | Notifications | Notification Service | Background service that sends automated notifications for task deadlines, meeting reminders, and system announcements to users via email |
| 2 | Data Backup | Automated Backup Job | Daily scheduled job that creates and stores backups of the system database to prevent data loss |
| 3 | UI | UI-Interface | Nice interface design and easy to use for users |

# II. Functional Requirements

## 1. User Authentication

### 1.1 Login

**Content #1: UI Layout (Mockup screen prototype)**



**Content #2: Brief description**

## **Function Description: Login**

**Actor/Role:** All Users (Admin, HR Officer, Department Manager, Employee)

**Purpose:** Allow users to log in to the system via email/password or Google Sign-In.

### **Function Interface:**

* Two text fields:  
  + Email
  + Password
* Three buttons:  
  + Login
  + Forgot Password
  + Google Login

### **Function Detail:**

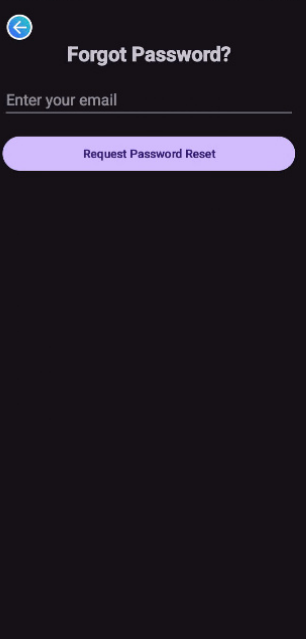
1. **Email and Password Login:**
   * Actor enters email and password.
   * The system validates the input:  
     + If email is empty → show red error: “Email is required”.
     + If email is invalid format → show red error: “Invalid email format”.
     + If password is empty → show red error: “Password is required”.
     + If password < 6 characters → show red error: “Password must be at least 6 characters”.
   * If both email and password are valid:  
     + System sends login request to backend.
     + If response is success:  
       - Navigate to correct Dashboard screen based on role (Admin → AdminDashboardActivity, HR → HRDashboardActivity…).
     + If response fails:  
       - Show message “Invalid email or password”.
       - Highlight both email and password fields in red.
2. **Forgot Password:**
   * When clicked, user is redirected to ForgotPasswordActivity.
3. **Google Sign-In:**
   * User clicks Google login button.
   * If Google account is selected:  
     + System sends ID token to backend for verification.
     + If the email is not registered → show: “This email is not registered. Please sign up.”
     + If success → redirect user to the proper Dashboard based on role.
     + If failed/canceled → show appropriate toast message.

**Content #3: Component/Field Specifications**

| **Field Name** | **Description** |
| --- | --- |
| Username | String, required, min length: 4, max length: 50. Must match registered username. |
| Password | String, required, min length: 6, max length: 50. Hidden input. |
| Login Button | Submits the credentials for validation. |
| Forgot Password? | Hyperlink to password recovery function. |

### 1.2 Forgot password

**Content #1: UI Layout (Mockup screen prototype)**



**Content #2: Brief description**

## **Function Description: Forgot Password**

**Actor/Role:** All Users (Admin, HR Officer, Department Manager, Employee)

**Purpose:** Allow users to request a password reset when they forget their login credentials.

### **Function Interface:**

* One input field:  
  + Email
* Two buttons:  
  + Request Password Reset
  + Back (← icon)

### **Function Detail:**

1. The actor enters their email address.
2. The system validates the input:  
   * If email is empty → show error message: **"Email is required"**, and highlight in red.
   * If email is not in the correct format → show error message: **"Invalid email format"**, and highlight in red.
3. If validation passes:  
   * The system sends a password reset request to the backend.
   * If successful → system displays a message: **“Reset request sent. Check your email.”** and navigates to the **Reset Password** screen, pre-filling the email field.
   * If failed → system shows appropriate error message in a toast.
4. If the back button is pressed, user returns to the login screen and all input is discarded.

## **Function Description: Reset Password**

**Actor/Role:** All Users (Admin, HR Officer, Department Manager, Employee)

**Purpose:** Allow users to reset their password using OTP verification sent via email.

### **Function Interface:**

* Three input fields:  
  + Email (pre-filled from previous screen if available)
  + OTP
  + New Password
* One button:  
  + Reset Password
* One back button:  
  + Back (← icon)

### **Function Detail:**

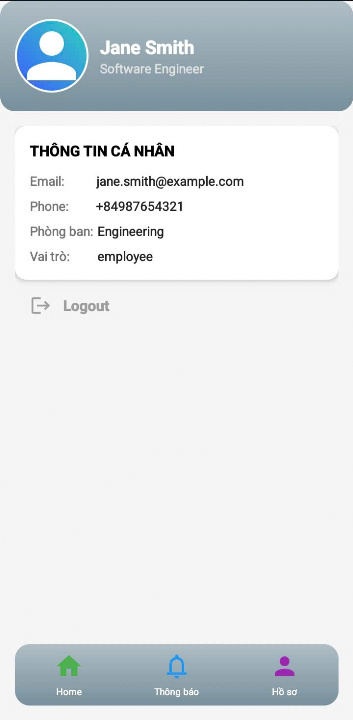
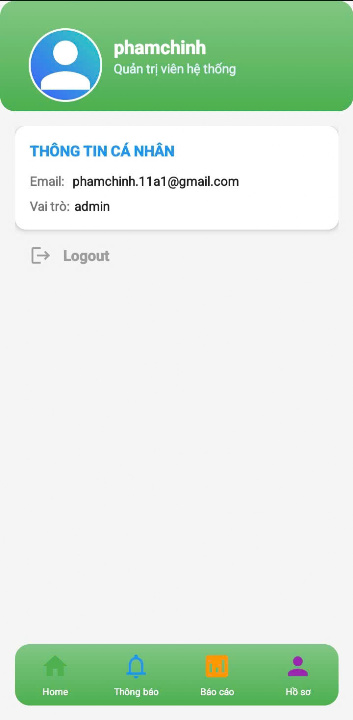
1. The actor enters:  
   * Email
   * OTP
   * New Password
2. The system validates:  
   * **Email:**
     + Empty → error: **"Email is required"**
     + Invalid format → error: **"Invalid email format"**
   * **OTP:**
     + Empty → error: **"OTP is required"**
   * **New Password:**
     + Less than 8 characters → error: **"Password must be at least 8 characters long"**
     + Missing uppercase → error: **"Must contain at least one uppercase letter"**
     + Missing lowercase → error: **"Must contain at least one lowercase letter"**
     + Missing number → error: **"Must contain at least one number"**
     + Missing special character (@$!%\*?&) → error: **"Must contain at least one special character"**
3. If all data is valid, the system calls resetPassword(email, otp, newPassword):  
   * If success → show message **"Password reset successful. Please login."** and return to login screen.
   * If error → show message with specific reason.
4. Clicking the back button will return to the **Forgot Password** screen and discard current inputs.

**Content #3: Component/Field Specifications**

| **Field Name** | **Description** |
| --- | --- |
| Email | String, must match email from OTP request. Required, valid format. |
| OTP | 6-digit number, string, required. Valid for 10 minutes from generation. |
| New Password | String, required, min 6 characters, must pass validatePassword() logic. |
| Confirm Password | Must match New Password. |
| Reset Password Button | Triggers resetPassword function. Validates and updates password. |
| Back to Login | Navigates back to the login screen. |

### 1.3 User Profile

**Content #1: UI Layout (Mockup Screen Prototype)**

****

**Content #2: Description**

## **Function Description: View Admin Profile**

**Actor/Role:** Admin (System Administrator)

**Purpose:** Display the administrator's personal and role information with the option to edit their profile and logout from the system.

### **Function Interface:**

* Displayed fields:  
  + Full Name (Username)
  + Role title
  + Email
  + Role name (admin)
* Two buttons:  
  + Edit Profile
  + Logout

### **Function Detail:**

1. When the admin logs in, the system receives the user object and loads their profile.
2. The system displays:  
   * Full name
   * System role description (e.g. "Quản trị viên hệ thống")
   * Email
   * Role
3. If data is missing → field will display "N/A"
4. The Edit Profile icon navigates to the profile editing screen (AdminEditProfile).
5. The back/home icon leads to the AdminDashboardActivity.
6. The Logout button clears all session/token data and redirects to the login screen.

**Function Description: View User Profile**

**Actor/Role:** Users(Dep Manager, HR, Employee)

**Purpose:** Allow users to view their personal and work-related information and perform actions such as editing their profile or logging out.

### **Function Interface:**

* Displayed fields:  
  + Full name
  + Position
  + Email
  + Phone number
  + Department
  + Role
* Two buttons:  
  + Edit Profile
  + Logout

### **Function Detail:**

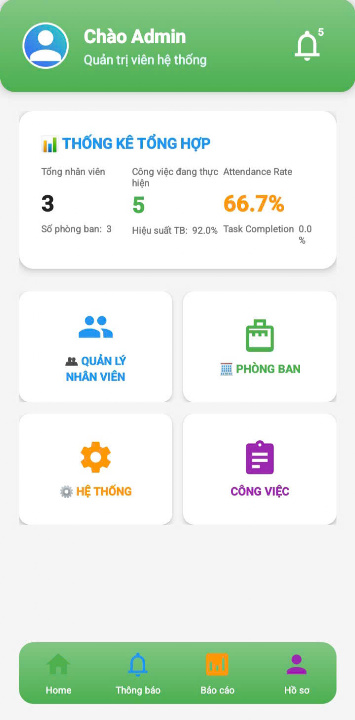
1. When the employee logs in, the system receives the user object and loads the profile.
2. The system displays:  
   * Full name
   * Position
   * Email
   * Phone number
   * Department
   * Role
3. If any data is missing → show default text “N/A”
4. The Edit Profile icon opens the corresponding editing activity (EmployeeEditProfile).
5. The Back/Home button redirects to the dashboard depending on role:  
   * Admin → AdminDashboardActivity
   * HR → HRDashboardActivity
   * Department Manager → ManagerDashboardActivity
   * Employee → EmployeeDashboardActivity
6. The Logout button clears session data and navigates to the login screen.

**Content #3: Screen Component Descriptions**

| **Field Group** | **Field Name** | **Description** |
| --- | --- | --- |
| **Common** | id | Integer, user ID, primary key |
|  | username | String, max 50 characters |
|  | email | String, valid email format |
|  | role | Enum: 'admin', 'hr', 'employee', 'dep\_manager' |
|  | profile\_type | Enum: 'admin' or 'employee', used to render correct UI layout |
| **Employee** | employee\_code | String, mã nhân viên |
|  | full\_name | String, kết hợp từ first\_name và last\_name |
|  | first\_name | String, họ |
|  | last\_name | String, tên |
|  | phone | String, định dạng số điện thoại, max 15 ký tự |
|  | position | String, chức vụ |
| **Department** | department.name | String, tên phòng ban |
|  | department.code | String, mã phòng ban |
| **Status** | hire\_date | Date (YYYY-MM-DD), ngày vào làm |
|  | status | String, trạng thái nhân viên: 'Active', 'Inactive', 'Resigned'… |

## 2. System Administration

### 2.1 Admin Dashboard



## **Function Description: Admin Dashboard**

**Actor/Role:** Admin (System Administrator)

**Purpose:** Display an overview of system statistics and provide access to management functions including employee, department, task, and system configuration.

### **Function Interface:**

* **Top Section:**
  + Avatar image & name: shows logged-in admin’s name
  + Role: “Quản trị viên hệ thống”
  + Notification bell icon with badge
* **Statistic Summary Card:**
  + Total employees
  + Number of departments
  + Active jobs
  + Average performance
  + Attendance rate
  + Task completion rate
* **4 Management Buttons:**
  + **Quản lý nhân viên**: navigate to user management
  + **Phòng ban**: navigate to department management
  + **Công việc**: navigate to task management
  + **Hệ thống**: reserved for future settings/system features
* **Bottom Navigation Menu:**
  + Home, Notification, Reports, Profile

### **Function Detail:**

1. After logging in, the system retrieves the current admin's data and statistics via StatsViewModel.
2. The dashboard loads:  
   * Employee count
   * Number of departments
   * Total ongoing tasks
   * Average performance (%)
   * Attendance rate (%)
   * Task completion rate (%)
3. Admin can:  
   * Tap avatar or profile icon to go to **AdminProfileActivity**
   * Tap “Quản lý nhân viên” → go to **UserManagementActivity**
   * Tap “Phòng ban” → go to **DepartmentManagementActivity**
   * Tap “Công việc” → go to **TaskManagementActivity**
4. If API fails to return data → Toast message appears: "Failed to load stats"
5. Notification badge shows the number of new alerts (currently hardcoded or dynamic)
6. All cards and icons use consistent color coding and rounded elevation card views

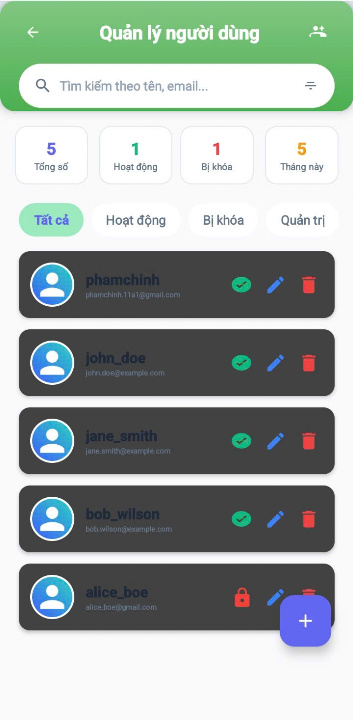
* menu.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| **User Info** | Displays the admin's name, role, and avatar at the top of the screen. |
| **Notification Icon** | Bell icon showing number of unread alerts. |
| **Summary Box** | Title “Thống Kê Tổng Hợp”, showing:  • Total Employees  • Task Count  • Attendance Rate  • Department Count  • Average Efficiency  • Task Completion Rate |
| **Module Tiles** | Four color-coded tiles:  • Employee Management  • Department Management  • System Settings  • Task Management |
| **Bottom Navigation Bar** | Fixed bar with tabs:  • Home  • Notifications  • Reports  • Profile |

### 2.2 User Management

#### *2.2.1 User List*



### **Function Description: User List Management**

**Actor/Role:** Admin System

**Purpose:** View and manage the list of users in the system, including filtering, searching, editing, locking, and deleting users.

**Function Interface:**

* One **search bar** for filtering by name or email.
* Four **filter tabs** (All, Active, Inactive, Admin).
* One **statistics bar** (Total, Active, Inactive, New this month).
* A **RecyclerView list** of user cards, each containing:  
  + Username and email
  + Role status (active, locked)
  + Icons for **edit**, **lock/unlock**, and **delete**
* A **floating action button** to create a new user.
* One **filter button** to filter by role.

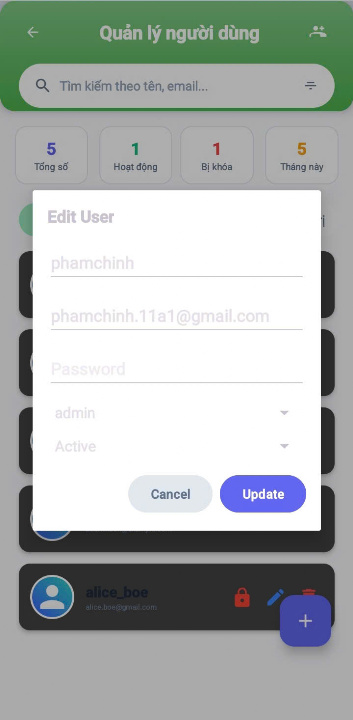
**Function Detail:**

* The actor can search for users using name or email via the search bar.
* The actor can tap filter tabs (Tất cả, Hoạt động, Bị khóa, Quản trị) to change the visible user list.
* The system updates the RecyclerView with the matching users.
* The actor can tap:  
  + (Edit): to update a user's information.
  + (Lock/Unlock): to change a user’s status between active and locked.
  + (Delete): to deactivate a user (soft delete).
* Pressing button opens a "Create User" dialog. If the form is completed and confirmed, the system creates a new user and updates the list.
* If required fields are empty or invalid during create/edit actions, the system shows validation errors with red highlights.
* Pressing back (← icon) will return to the previous screen and discard any unsaved data.
* The statistics above (Total, Active, Locked, This Month) are dynamically updated when users are modified.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| **Search Bar** | Allows input of name or email to filter the displayed user list. |
| **Summary Cards** | Small boxes showing:  • Total Users  • Active Users  • Blocked Users  • Users This Month |
| **Filter Tabs** | Quick filters:  • All (Tất cả)  • Active (Hoạt động)  • Blocked (Bị khóa)  • Admin (Quản trị) |
| **User Item** | Displays user avatar, username, email, and action icons:  • (Active)  • (Blocked)  • (Edit)  • (Delete) |
| **Floating Action Button** | The “+” button used to add a new user. Positioned at the bottom right corner. |

#### *2.2.2 User Details*



### **Function Description: Edit User Information**

**Actor/Role:** Admin System

**Purpose:** Update user information such as username, password, role, and status.

**Function Interface:**

* **Three text fields**:  
  + Username
  + Email (read-only)
  + Password
* **Two dropdowns (Spinners)**:  
  + Role
  + Active Status
* **Two buttons**:  
  + Update
  + Cancel

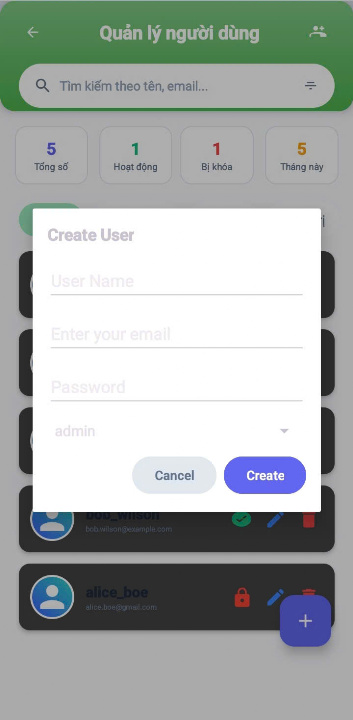
**Function Detail:**

* The actor opens the user edit dialog by clicking the edit icon from the user list.
* The system displays current values in the respective fields.
* The actor can modify the **username**, **password**, **role**, and **active status**.
* The email is displayed but **not editable**.
* When the actor clicks Update, the system performs validation:  
  + The **password** (if entered) must meet security rules (validated by PasswordValidator).
  + Any field with invalid data will **trigger an alert dialog** and prevent submission.
* If validation passes and the data has been changed, the system sends the update.
* If the actor clicks Cancel, the dialog will be dismissed without saving any changes.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Username | Text input to update the user's name. |
| Email | Non-editable field showing user email. |
| Password | Optional input to update user password. |
| Role | Dropdown to select user role. |
| Status | Dropdown to set the user as Active/Blocked. |
| Cancel Button | Discards changes and closes the form. |
| Update Button | Saves changes and updates user data. |

#### *2.2.3 Create New User*



### **Function Description: Create New User**

**Actor/Role:**Admin System

**Purpose:** Create a new user account in the system with specific role and credentials.

**Function Interface:**

* **Three text fields** to input:  
  + Username
  + Email
  + Password
* **One dropdown (spinner)** to select:  
  + Role
* **Two buttons**:  
  + Create
  + Cancel

**Function Detail:**

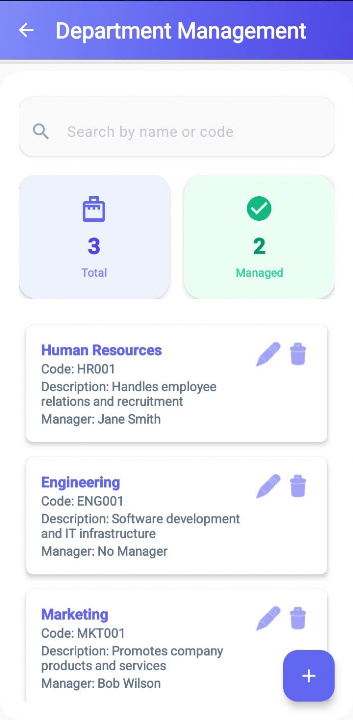
* The actor opens the Create User dialog from the user management interface.
* The actor fills in all required fields including: **username**, **email**, **password**, and selects the **role**.
* When the Create button is clicked:  
  + The system will check whether all required fields are filled.
  + If any required field is empty, the system will show a **notification**:  
     ➤ *"Please fill in all required fields"*
  + The system also validates the password using PasswordValidator.  
    - If the password is invalid, a popup message will appear showing the reason.
  + If all data is valid, the system creates a new User object and sends it to the listener for further processing.
* If the actor clicks the Cancel button, the dialog closes and **no data is saved**.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Username | Text input for entering the user's name. |
| Email | Input field for the user's email address. |
| Password | Input for setting a password. |
| Role | Dropdown to assign the user role. |
| Cancel Button | Discards input and closes the form. |
| Create Button | Saves new user data and adds to the system. |

### 2.3 Department Management

#### *2.3.1 Department List*



### **Function Description: Department List**

**Actor/Role:**Admin System

**Purpose:** Manage the list of departments in the organization.

**Function Interface:**

* **One search bar** to filter departments by name.
* **Two statistics fields** displaying:  
  + Total number of departments
  + Number of departments with assigned managers
* **One scrollable list** (RecyclerView) to show department details
* **One floating action button (FAB)** to add a new department
* **Empty state layout** for when no departments are found
* **Two dialogs**:  
  + Create Department dialog
  + Edit Department dialog

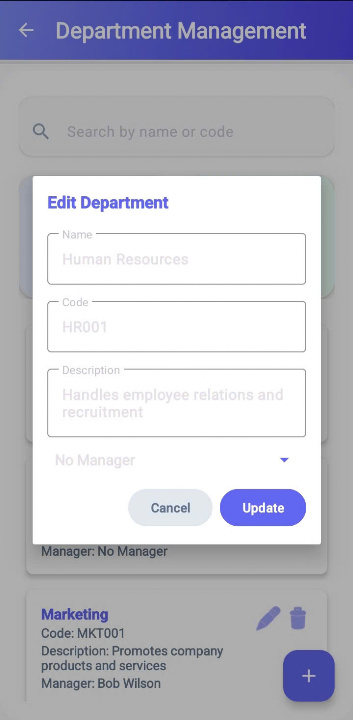
**Function Detail:**

* The actor can view the list of all departments and their assigned manager status.
* The actor can type into the **search bar** to filter the department list dynamically.
* The list shows department items retrieved from the server via ViewModel.
* The actor can:  
  + Tap the **FAB** to open the Create Department dialog
  + Tap the **edit icon** on a department to update details via Edit dialog
  + Tap the **delete icon** to remove a department (with confirmation dialog)
* When the department list is empty or no results are found, the system shows an **empty state view** with a friendly message and image.
* The UI displays statistics about:  
  + Total departments in the system
  + How many departments have been assigned a manager
* Upon successful creation, update, or deletion of a department:  
  + The system refreshes the list
  + The statistics section is updated accordingly
* If an operation fails, the system shows a toast message with the error

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Search Bar | Input to search by department name or code. |
| Total | Shows total number of departments. |
| Managed | Shows departments with an assigned manager. |
| Department Name | Display name of the department. |
| Code | Department identification code. |
| Description | Description of department responsibilities. |
| Manager | Person assigned to manage the department. |
| Edit Icon | Opens edit form for department details. |
| Delete Icon | Deletes the selected department. |
| Add Button (+) | Open form to create a new department. |

#### *2.3.2 Department Details*



### **Function Description: Edit Department**

**Actor/Role:** Admin System

**Purpose:** Update the information of an existing department in the organization.

**Function Interface:**

* Three text fields to enter/edit:  
  + Department Name
  + Department Code
  + Description
* One dropdown (spinner) to select a Manager (optional)
* Two buttons:  
  + **Update**
  + **Cancel**

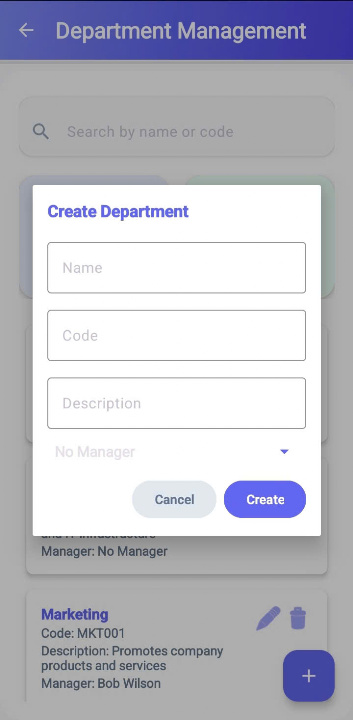
**Function Detail:**

* The actor opens the **Edit Department** dialog pre-filled with the current information of the selected department.
* The actor can edit the **Name**, **Code**, and **Description**, and optionally assign a new **Manager**.
* The system validates required fields:  
  + **Name** and **Code** are mandatory.
  + If these fields are empty, the system shows an error message and highlights the field in red.
* When the actor clicks **Update**, the system sends the updated information to the server:  
  + If the update is successful, a success message is displayed and the department list is refreshed.
  + If the update fails, an error message is shown.
* When the actor clicks **Cancel**, the dialog is closed and no changes are saved.
* The spinner loads all eligible employees that can be assigned as managers. If no manager is chosen, the department will have "No Manager".

**Field Description**

| **Field** | **Description** |
| --- | --- |
| **Name** | Editable department name. |
| **Code** | Department ID (non-editable). |
| **Description** | Editable department purpose/details. |
| **Manager** | Dropdown to assign a user as department manager. |
| **Cancel** | Close form without saving changes. |
| **Update** | Save the changes made to the department. |

#### *2.3.2 Create Department*



### **Function Description: Create Department**

**Actor/Role:**Admin System

**Purpose:** Create a new department in the center

**Function Interface:**

* Three text fields to input information about the department:  
  + Name
  + Code
  + Description (optional)
* One dropdown (spinner) to select the manager (optional)
* Two buttons:  
  + **Create**
  + **Cancel**

**Function Detail:**

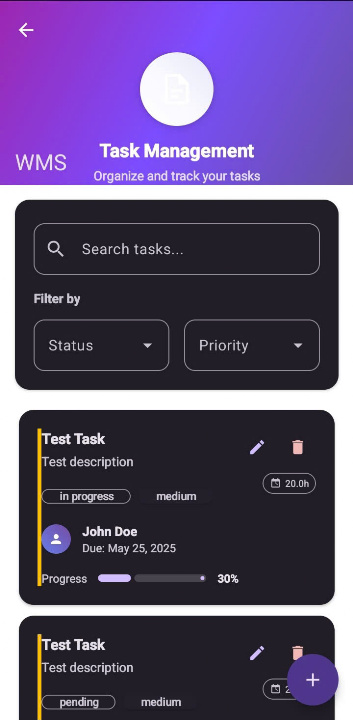
* The actor fills in the department name, code, and optional description.
* The actor can optionally assign a manager by selecting from the dropdown list.
* The system will **verify all data** that the actor fills in. Specifically:  
  + If the **Name** field is empty, the system displays a red error message: "Name is required"
  + If the **Code** field is empty, the system displays a red error message: "Code is required"
  + No validation is required for the description and manager fields.
* If all required fields are valid and the actor clicks **Create**, the system will send a request to the backend to create a new department.  
  + If the creation is **successful**, the system shows a toast notification "Department created successfully" and closes the dialog.
  + If the creation **fails**, the system shows an error toast like "Failed to create department"
* If the actor clicks **Cancel**, the dialog is closed and all unsaved data is discarded.

**Field Description**

| **Field** | **Description** |
| --- | --- |
| **Name** | Create department names. |
| **Code** | Department ID (non-editable). |
| **Description** | Create department purpose/details. |
| **Manager** | Dropdown to assign a user as department manager. |
| **Cancel** | Close form without saving changes. |
| **Create** | Create a new department. |

### 2.4 Task Management

#### *2.4.1 Task List*



### **Function Description: Task List Management**

**Actor/Role:**Admin System

**Purpose:** View, search, filter, and manage tasks within the system

**Function Interface:**

* One search field to filter tasks by keyword
* Two dropdown fields to filter by:  
  + Status (All, pending, in\_progress, review, completed, cancelled)
  + Priority (All, low, medium, high, urgent)
* One list view to display matching tasks
* One empty state layout (when no tasks found)
* One floating button (➕) to add a new task
* (Internally) buttons/actions per task to edit or delete

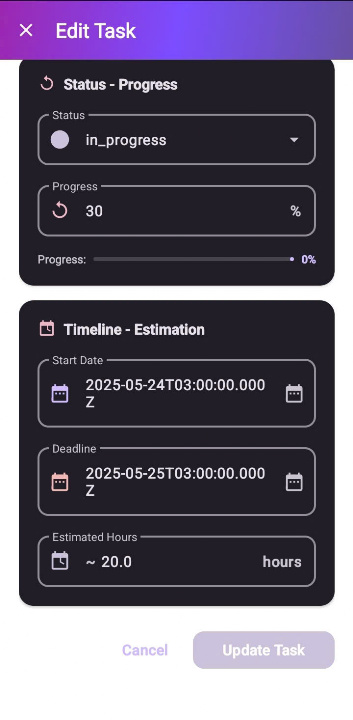
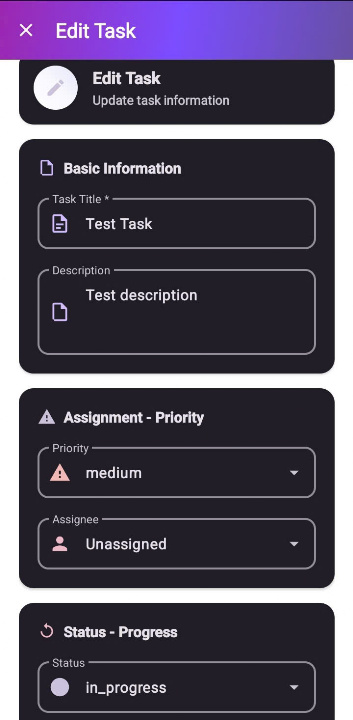
**Function Detail:**

* The actor can enter a **search keyword** to filter tasks in real-time.
* The actor can **filter tasks** using the **status** and **priority** dropdowns.
* The **task list is automatically updated** based on filters and search input.
* If there are **no matching tasks**, an empty state layout with icon and message is displayed.
* The actor can click the **Add (+) button** to navigate to the **Add Task** screen.
* The actor can click on a task in the list to:  
  + **Edit the task** (navigates to Edit screen)
  + **Delete the task** (after confirmation, task is removed from the list)
* When task data is being fetched or reloaded (after delete or change of filter), the system ensures up-to-date data is reflected.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Search Bar | Input field to search tasks by name or keywords. |
| Filter by Status | Dropdown to filter tasks by status (pending, in progress, etc.). |
| Filter by Priority | Dropdown to filter tasks by priority (low, medium, high). |
| Task Title | Display name/title of the task. |
| Task Description | Summary of what the task is about. |
| Status Label | Indicates the current task status (e.g., in progress, pending). |
| Priority Label | Indicates the task priority level. |
| Estimated Time | Shows expected effort in hours (e.g., 20.0h). |
| Assignee | Person responsible for the task (name and avatar). |
| Due Date | Deadline for task completion. |
| Progress Bar | Visual indicator of task completion percentage. |
| Edit Icon | Opens task editing form. |
| Delete Icon | Deletes the selected task. |
| Add Button (+) | Opens the form to create a new task. |

#### *2.4.2 Task Details*



### **Function Description: Edit Task**

**Actor/Role:**Admin System

**Purpose:** Update the details of an existing task in the system

**Function Interface:**

* **Six text fields** to input task information:  
  + Title
  + Description
  + Start Date
  + Deadline
  + Estimated Hours
  + Progress
* **Three dropdown fields** (AutoCompleteTextView):  
  + Priority (low, medium, high, urgent)
  + Status (pending, in\_progress, review, completed, cancelled)
  + Assignee (Unassigned or employee/user list)
* **Two buttons**:  
  + Update
  + Cancel

**Function Detail:** The actor selects a task from the task list to open the edit interface.  
 The form is pre-filled with the task's existing data.  
 The actor modifies any field such as title, description, status, etc.

When the actor clicks **Update**:

* The system validates all input fields:  
  + **Title** is mandatory. If empty, the system highlights it in red and shows an error message.
  + Other fields are optional but must be in valid format if filled (e.g., progress must be numeric).
* If validation passes:  
  + The system sends an update request.
  + If successful, a toast confirms the task was updated and the screen is closed.
  + If failed, an error message is shown.

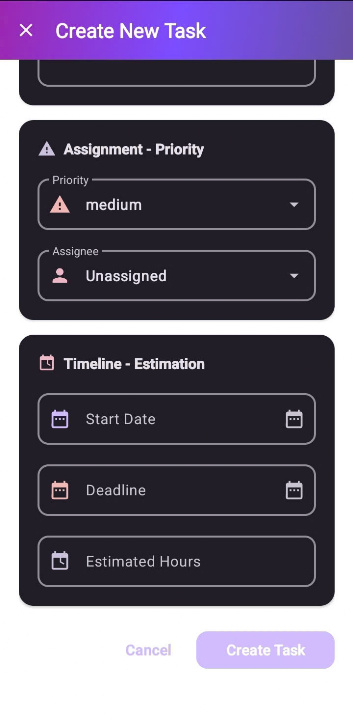
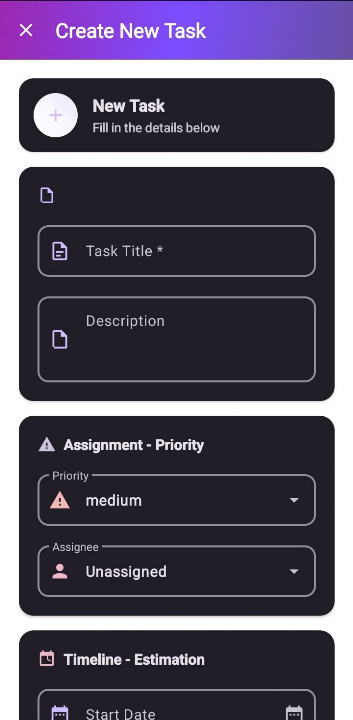
When the actor clicks **Cancel**:

* The system closes the edit screen/dialog.
* **No changes** are saved.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Task Title | Name of the task. Required. |
| Description | Summary or details of the task. |
| Priority | Priority level (e.g., low, medium, high). |
| Assignee | Person responsible for completing the task. |
| Status | Current state of the task (pending, in\_progress, done). |
| Progress (%) | Numeric field to indicate percentage completion. |
| Start Date | Date when the task begins. |
| Deadline | Due date for the task. |
| Estimated Hours | Estimated time to complete the task. |
| Cancel Button | Discards edits and returns to previous screen. |
| Update Task Button | Saves changes to the task. |

#### *2.4.3 Create Task*



### **Function Description**

**Actor/Role:**Admin System

**Purpose:** Create a new task in the project management system.

**Function Interface:**

* **Six text fields** to input task information:  
  + Title
  + Description
  + Start Date
  + Deadline
  + Estimated Hours
  + Assignee
* **Two dropdown lists**:  
  + Priority (Low, Medium, High, Urgent)
  + Assignee (List of employees)
* **Two buttons**:  
  + **Create** (to save task)
  + **Cancel** (to dismiss the form without saving)

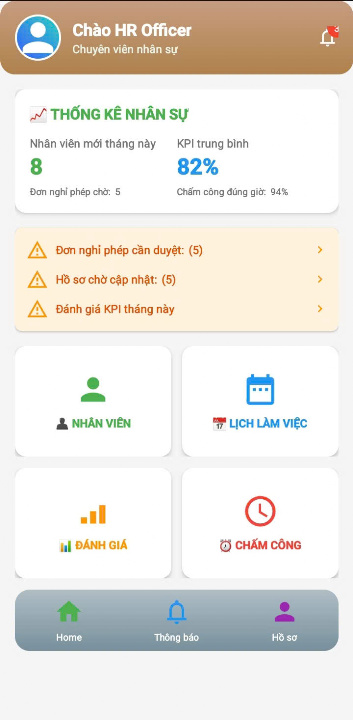
**Function Detail:**

* The actor opens the "Create Task" screen from the toolbar or dialog.
* The actor inputs or selects all necessary information for the task, including selecting the assignee and priority.
* The system performs validation:  
  + If the **Title** is empty, it shows a red error notification on the title field.
  + Other fields are optional but can be validated based on project rules (e.g., valid date format, numeric hours).
* Once all required information is correct:  
  + If the actor presses **Create**, the task is saved and a success message appears.
  + If the actor presses **Cancel** or the close icon, the screen/dialog is dismissed, and all input data is discarded without saving.

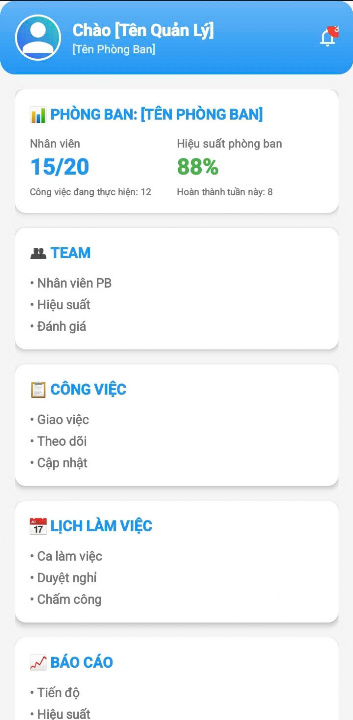
**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Task Title | Required field for the task name. |
| Description | Optional summary or task details. |
| Priority | Task importance: low, medium, or high. |
| Assignee | Person assigned to the task. |
| Status | Task status: pending, in\_progress, completed. |
| Progress (%) | Starting completion percentage. |
| Start Date | Planned start of the task. |
| Deadline | Due date for the task. |
| Estimated Hours | Expected time to finish the task. |
| Cancel Button | Cancels task creation. |
| Create Task Button | Submits and adds the new task to the list. |

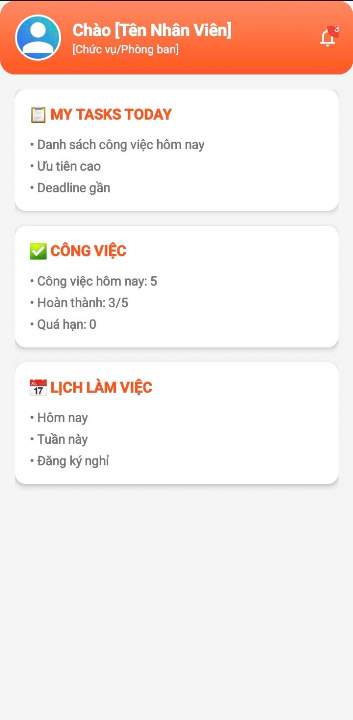
## 3. System HR



## 4. System Dep\_Manager



## 5. System Employee

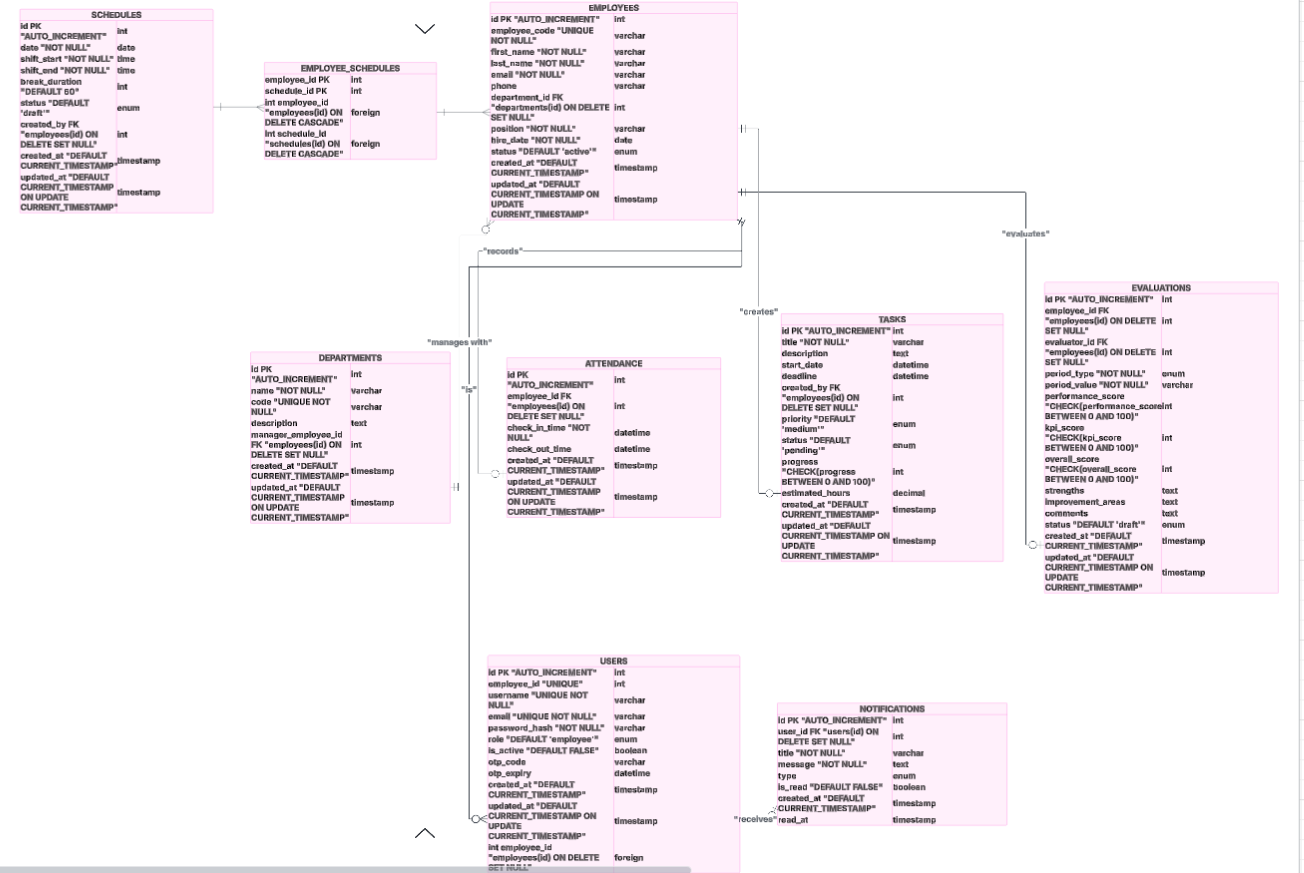


#### III. System Design

## 1. Database Design

*[Provide the tables relationship like example below]*

### 1.1 Database Schema



### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

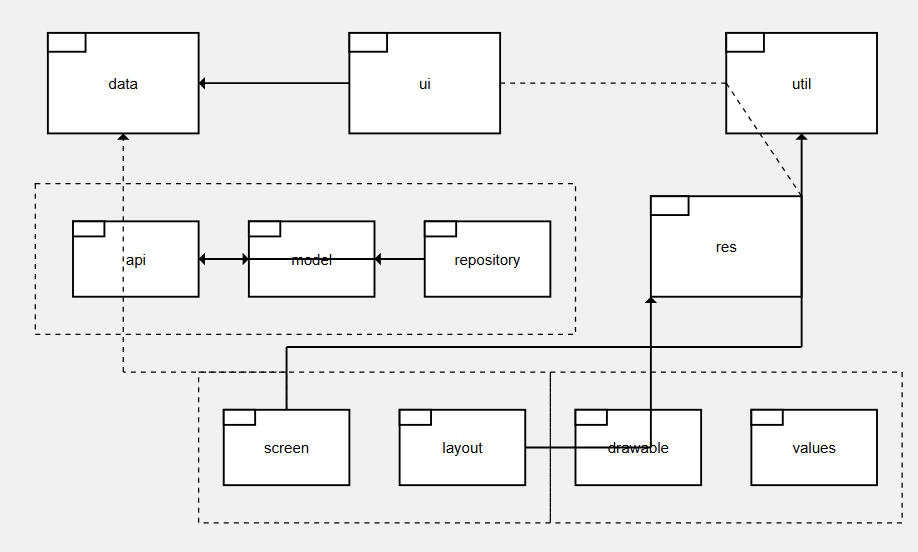
### 

### 1.2 Table Descriptions

| **No** | **Table Name** | **Description** | **Primary Keys** | **Foreign Keys** | **Unique Keys** |
| --- | --- | --- | --- | --- | --- |
| 01 | **users** | Stores user accounts for authentication and role access. | id | employee\_id ➝ employees.id (nullable, ON DELETE SET NULL) | username, email |
| 02 | **departments** | Holds department details; links to department manager (employee). | id | manager\_id ➝ employees.id (ON DELETE SET NULL) | code |
| 03 | **employees** | Employee profiles; linked to departments. | id | department\_id ➝ departments.id (ON DELETE SET NULL) | employee\_code |
| 04 | **tasks** | Manages tasks: creator, status, progress, deadlines. | id | created\_by ➝ employees.id (ON DELETE SET NULL) | – |
| 05 | **schedules** | Defines shift schedules; created by an employee. | id | created\_by ➝ employees.id (ON DELETE SET NULL) | – |
| 06 | **employee\_schedules** | Junction table linking employees ↔ schedules (Many‑to‑Many). | (employee\_id, schedule\_id) | employee\_id ➝ employees.id (CASCADE), schedule\_id ➝ schedules.id (CASCADE) | Composite PK |
| 07 | **attendance** | Records employee check‑in/check‑out times. | id | employee\_id ➝ employees.id (ON DELETE SET NULL) | – |
| 08 | **evaluations** | Stores performance reviews with period info (year/quarter/month/custom). | id | employee\_id ➝ employees.id, evaluator\_id ➝ employees.id (both ON DELETE SET NULL) | – |
| 09 | **notifications** | Tracks in-app notifications per user. | id | user\_id ➝ users.id (ON DELETE SET NULL) | – |

## 2. Code Packages

### 2.1 Package Diagram



Personnel Management App

├── java/

│ ├── com.company.personnelapp/

│ │ ├── data/

│ │ │ ├── api/

│ │ │ ├── model/

│ │ │ └── repository/

│ │ ├── ui/

│ │ │ └── layout/

│ │ │ ├── screen/

│ │ │ ├── adapter/

│ │ │ └── fragment/

│ │ └── util/

│ └── MainActivity.java

└── res/

├── drawable/

├── layout/

└── values/

### 2.2 Package Descriptions

| **Package Name** | **Sub-Package** | **Description** | **Key Components** |
| --- | --- | --- | --- |
| **data** |  | Root package for data management and business logic | Contains all data-related operations |
|  | **api** | Handles REST API communications and network operations | - ApiService.java<br>- RetrofitClient.java<br>- ApiResponse.java<br>- NetworkCallback.java |
|  | **model** | Contains data models and entity classes | - Employee.java<br>- Department.java<br>- Attendance.java<br>- Payroll.java<br>- User.java<br>- WorkSchedule.java |
|  | **repository** | Implements Repository pattern for data access | - EmployeeRepository.java<br>- AttendanceRepository.java<br>- PayrollRepository.java<br>- DatabaseHelper.java |
| **ui** |  | User Interface components and presentation layer | Contains all UI-related classes |
|  | **layout** | UI layout management and screen organization | Contains screen-specific UI components |
|  | **screen** | Individual screen activities and fragments | - LoginActivity.java<br>- DashboardActivity.java<br>- EmployeeListActivity.java<br>- AddEmployeeActivity.java<br>- AttendanceActivity.java<br>- PayrollActivity.java<br>- ProfileActivity.java |
|  | **adapter** | RecyclerView adapters for list displays | - EmployeeAdapter.java<br>- AttendanceAdapter.java<br>- DepartmentAdapter.java |
|  | **fragment** | Reusable UI fragments | - EmployeeFragment.java<br>- AttendanceFragment.java<br>- ReportsFragment.java |
| **util** |  | Utility classes and helper functions | - DateUtils.java<br>- ValidationUtils.java<br>- PreferenceManager.java<br>- Constants.java<br>- Logger.java |
| **res** |  | Android resource files | Contains all application resources |
|  | **drawable** | Image and vector drawable resources | - ic\_employee.xml<br>- ic\_attendance.xml<br>- ic\_payroll.xml<br>- app\_logo.png<br>- background\_gradient.xml |
|  | **layout** | XML layout files for UI screens | - activity\_main.xml<br>- activity\_login.xml<br>- activity\_dashboard.xml<br>- fragment\_employee\_list.xml<br>- item\_employee.xml<br>- dialog\_add\_employee.xml |
|  | **values** | Application values and configurations | - strings.xml<br>- colors.xml<br>- dimens.xml<br>- styles.xml<br>- arrays.xml |