

## Đề tài 7

Xây dựng hệ thống quản lý thời gian làm việc và báo cáo công việc theo tuần được miêu tả như sau:

Hệ thống quản lý các đối tượng sau: Project, Task, User; cụ thể là:

- User: quản lý đăng ký/ đăng nhập. User có chức năng CRUD thông tin cá nhân, tạo Project (trở thành LEADER của Project), tham gia Project bằng mã mời ,tạo Task trong Project
- Project: CRUD thông tin Project, phân quyền cho các thành viên tham gia trong Project gồm có
  - + Leader: có quyền tạo Task và gán trách nhiệm cho thành viên, UPDATE thông tin Project, trao quyền VICE\_LEADER cho thành viên trong nhóm, trao lại quyền LEADER, thay đổi visibility của Project, giải tán Project, quản lý thành viên trong Project
  - + Vice\_leader: được LEADER chọn từ danh sách thành viên và có các quyền giống LEADER nhưng không được phép giải tán Project
  - + Member: tạo Task do cá nhân Member đảm nhiệm (có khả năng UPDATE thông tin Task), không được UPDATE thông tin Task được LEADER và VICE\_LEADER giao, UPDATE trạng thái Task, gửi kết quả của Task (chức năng upload file)
- Task: được chia làm 2 loại, Task được quản lý giao và Task bản thân thành viên tạo. Có các chức năng CRUD thông tin Task, có chức năng upload file

Đồng thời hệ thống có chức năng thông báo cho thành viên trong Project khi được giao Task, khi sắp đến deadline

## Mục lục

Mục lục.....	1
Chương I: Giới thiệu chung.....	2
1.1. Lý do chọn đề tài.....	2
1.2. Mục tiêu đề tài.....	2
1.3. Đối tượng và phạm vi áp dụng.....	2
1.3.1. Đối tượng áp dụng.....	2
1.3.2. Phạm vi áp dụng.....	3
1.4. Bố cục báo cáo.....	4
Chương II: Phân tích yêu cầu.....	5
2.1. Yêu cầu chức năng.....	5
2.1.1. Quản lý người dùng (User Management).....	5
2.1.2. Quản lý dự án (Project Management).....	5
2.1.3. Quản lý công việc (Task Management).....	6
2.1.4. Hệ thống thông báo (Notification System).....	7
2.1.5. Chức năng lọc (Filtering & Searching).....	7
2.1.6. Xuất báo cáo (Export Report).....	8
2.2. Yêu cầu phi chức năng.....	9
2.2.1. Hiệu năng (Performance).....	9
2.2.2. Bảo mật (Security).....	9
2.2.3. Khả năng mở rộng (Scalability).....	9
2.2.4. Tính ổn định (Reliability).....	10
2.2.5. Dễ sử dụng (Usability).....	10
2.2.6. Khả dụng cao (Availability).....	10
2.2.7. Khả năng bảo trì (Maintainability).....	10
2.3. Đặc tả các trường hợp sử dụng.....	11
2.3.1. Use Case: Đăng ký.....	11
2.3.2. Use Case: Đăng nhập.....	12

2.3.3. Use Case: Chỉnh sửa thông tin cá nhân.....	12
2.3.4. Use Case: Tham gia dự án.....	13
2.3.5. Use Case: Tạo nhiệm vụ.....	13
2.3.6. Use Case: Xem thông tin nhiệm vụ.....	14
2.3.7. Use Case: Xem thông tin dự án.....	14
2.3.8. Use Case: Xóa dự án.....	14
2.3.9. Use Case: Chỉnh sửa dự án.....	15
2.3.10. Use Case: Chỉnh sửa nhiệm vụ.....	15
2.3.11. Use Case: Xem báo cáo.....	16
Chương III: Thiết kế hệ thống.....	17
3.1 Kiến trúc tổng thể.....	17
3.2. Mô hình triển khai.....	18
3.3. Thiết kế cơ sở dữ liệu.....	19
3.4. Thiết kế giao diện.....	29
3.5. Công cụ và môi trường phát triển.....	34
3.5.1. IDE (Integrated Development Environment).....	34
3.5.2. Version Control.....	34
3.5.3. Công cụ xây dựng & quản lý phụ thuộc.....	34
3.5.4. Cơ sở dữ liệu.....	34
3.5.5. Môi trường chạy ứng dụng.....	34
3.5.6. Công cụ hỗ trợ nhóm.....	35
Chương IV: Kế hoạch thực hiện.....	36
Giai đoạn 1 – Khởi động & phân tích yêu cầu.....	36
Giai đoạn 2 – Thiết kế hệ thống & lập trình backend cơ bản.....	36
Giai đoạn 3 – Hoàn thiện backend & điều chỉnh.....	36
Giai đoạn 4 – Phát triển frontend & tích hợp.....	36
Giai đoạn 5 – Hoàn thiện sản phẩm & bàn giao.....	37
Chương V: Kết luận và định hướng phát triển trong tương lai.....	38
5.1. Kết luận.....	38

<u>5.2. Hướng phát triển.....</u>	<u>38</u>
<u>Tài liệu tham khảo.....</u>	<u>40</u>

## **Chương I: Giới thiệu chung**

### **1.1. Lý do chọn đề tài**

Trong môi trường doanh nghiệp hiện đại, việc quản lý công việc hiệu quả và minh bạch đóng vai trò then chốt trong việc đảm bảo tiến độ dự án và đánh giá hiệu suất nhân sự. Tuy nhiên, nhiều nhóm làm việc vẫn sử dụng phương pháp thủ công như file Excel hoặc báo cáo giấy, dẫn đến nhiều bất cập như:

- Thiếu tính tập trung và hệ thống trong quản lý đầu việc.
- Khó theo dõi tiến độ và hiệu quả công việc theo thời gian.
- Mất thời gian tổng hợp báo cáo thủ công.
- Thiếu công cụ hỗ trợ trực quan cho cả nhân viên lẫn người quản lý.

Trong bối cảnh đó, việc xây dựng một hệ thống phần mềm hỗ trợ quản lý thời gian làm việc và báo cáo công việc giúp giải quyết các vấn đề trên là hết sức cần thiết. Đặc biệt, đề tài phù hợp với sinh viên đang thực tập tại doanh nghiệp, giúp rèn luyện kỹ năng lập trình, phân tích hệ thống, đồng thời hiểu rõ quy trình làm việc thực tế trong doanh nghiệp.

### **1.2. Mục tiêu đề tài**

Mục tiêu chính của đề tài là xây dựng một ứng dụng web đơn giản, dễ sử dụng hỗ trợ:

- Nhân viên/sinh viên thực tập: ghi nhận công việc hàng ngày, theo dõi tiến độ công việc theo trạng thái, quản lý đầu việc theo ngày, tuần
- Quản lý/ trưởng nhóm: xem tổng hợp công việc của từng người theo tuần, theo dõi hiệu suất, khối lượng công việc, xuất báo cáo công việc dưới dạng Excel và PDF để phục vụ đánh giá

### **1.3. Đối tượng và phạm vi áp dụng**

#### **1.3.1. Đối tượng áp dụng**

Hệ thống hướng đến các đối tượng sau:

- Sinh viên thực tập

- Có nhu cầu ghi nhận công việc hàng ngày trong quá trình thực tập tại doanh nghiệp
- Cần báo cáo công việc hàng tuần cho người hướng dẫn hoặc quản lý
- Nhân viên trong doanh nghiệp
  - Làm việc theo nhóm hoặc dự án, cần cập nhật và theo dõi đầu việc cá nhân
  - Muốn có công cụ đơn giản để quản lý tiến độ và hiệu suất làm việc cá nhân
- Trưởng nhóm/quản lý
  - Quản lý dự án, nhóm nhân viên, sinh viên thực tập
  - Cần theo dõi tổng quan khối lượng công việc của từng người, hỗ trợ đánh giá định kỳ

### **1.3.2. Phạm vi áp dụng**

Phạm vi áp dụng của hệ thống bao gồm:

- Chức năng chính:
  - Quản lý người dùng (User) và phân quyền cơ bản
  - Quản lý dự án (Project)
  - Quản lý công việc (Task)
  - Theo dõi tiến độ công việc, dự án theo trạng thái (status), tiến độ (progress)
  - Thống kê công việc theo tuần, theo thành viên
  - Xuất file báo cáo dưới dạng PDF hoặc Excel
- Phạm vi kỹ thuật:
  - Triển khai nội bộ, sử dụng localhost hoặc mạng LAN
  - Thiết kế REST API theo mô hình client-server
  - Sử dụng các công nghệ web phổ biến như Java(Spring Boot), Python(Flask), hoặc NodeJS(Javascript)

- Lưu trữ bằng hệ quản trị cơ sở dữ liệu quan hệ (MySQL hoặc PostgreSQL)
- Có thể chạy bằng Docker (không bắt buộc)

#### **1.4. Bố cục báo cáo**

Báo cáo được chia thành 5 chương, với nội dung cụ thể như sau:

##### **Chương I – Giới thiệu chung**

Trình bày tổng quan về đề tài, bao gồm lý do chọn đề tài, mục tiêu thực hiện, đối tượng và phạm vi áp dụng, cùng phương pháp nghiên cứu. Chương này giúp người đọc hiểu được bối cảnh và ý nghĩa thực tiễn của đề tài.

##### **Chương II – Phân tích yêu cầu**

Tập trung vào việc khảo sát hiện trạng, xác định các yêu cầu chức năng và phi chức năng của hệ thống. Đây là cơ sở để đảm bảo hệ thống được xây dựng đúng mục tiêu và phù hợp với nhu cầu thực tế.

##### **Chương III – Thiết kế hệ thống**

Trình bày chi tiết các thiết kế về mặt chức năng và kiến trúc của hệ thống, bao gồm sơ đồ use case, sơ đồ cơ sở dữ liệu, kiến trúc tổng thể và giao diện người dùng. Chương này thể hiện tư duy thiết kế và khả năng hiện thực hóa các yêu cầu đã phân tích.

##### **Chương IV – Kế hoạch thực hiện**

Mô tả tiến trình triển khai đề tài theo các giai đoạn cụ thể, các công nghệ được lựa chọn, công cụ hỗ trợ, và cách thức quản lý mã nguồn. Chương này nhằm chứng minh khả năng tổ chức và quản lý dự án phần mềm thực tế.

##### **Chương V – Kết luận và định hướng phát triển trong tương lai**

Tổng kết các kết quả đã đạt được, đánh giá những hạn chế còn tồn tại và đưa ra những đề xuất phát triển hệ thống trong tương lai. Chương này khép lại báo cáo với cái nhìn tổng thể và định hướng tiếp theo cho đề tài.

## Chương II: Phân tích yêu cầu

### 2.1. Yêu cầu chức năng

#### 2.1.1. Quản lý người dùng (User Management)

- **Đăng ký (Register):**
  - Người dùng cung cấp email, mật khẩu và thông tin cá nhân để đăng ký.
  - Kiểm tra trùng email.
  - Mật khẩu phải được mã hóa.
- **Đăng nhập/Đăng xuất (Login/Logout):**
  - Hệ thống xác thực bằng email + mật khẩu.
  - Sau khi xác thực, hệ thống trả JWT token để truy cập API.
  - Logout có thể là xoá token phía client (stateless).
- **Cập nhật thông tin cá nhân (CRUD Profile):**
  - Cho phép thay đổi avatar, tên hiển thị, thông tin liên hệ cá nhân.
  - Không cho phép chỉnh sửa email nếu không xác thực lại.
- **Tạo Project mới:**
  - Người dùng tạo Project sẽ trở thành **LEADER** mặc định.
  - Hệ thống sinh mã mời để mời người khác tham gia.
- **Tham gia Project bằng mã mời:**
  - Nhập mã hợp lệ → được thêm vào Project với vai trò mặc định là **MEMBER**.

#### 2.1.2. Quản lý dự án (Project Management)

- **Tạo/đọc/sửa/xoá Project:**
  - LEADER có thể chỉnh sửa các thuộc tính: tên, mô tả, trạng thái, thời gian.
  - LEADER có thể xoá hoặc giải tán Project.



- **Phân quyền trong Project:**

- **LEADER:**

- Gán quyền VICE\_LEADER cho thành viên.
    - Trao lại quyền LEADER cho người khác.
    - Gỡ bỏ thành viên khỏi Project.
    - Thay đổi trạng thái công khai/riêng tư (visibility).
    - Giải tán Project.

- **VICE\_LEADER:**

- Tạo/giao Task cho các thành viên.
    - Cập nhật thông tin Project.
    - Không được giải tán Project.

- **MEMBER:**

- Chỉ có thể tạo Task cá nhân.
    - Không được giao việc cho người khác.

- **Quản lý thành viên:**

- Hiện thị danh sách thành viên và vai trò.
  - Thêm thành viên bằng mã mời.
  - Gỡ thành viên (chỉ LEADER và VICE\_LEADER được làm).

### **2.1.3. Quản lý công việc (Task Management)**

#### **a. Tạo và gán Task**

- **LEADER/VICE\_LEADER:**

- Tạo Task mới và gán người phụ trách.
  - Nhập tên Task, mô tả, deadline, file đính kèm (nếu có).
  - Gửi thông báo đến người phụ trách.

- **MEMBER:**

- Tạo Task cá nhân.
  - Chỉ sửa/xoá các Task do chính họ tạo ra.

### b. Quản lý Task

- Xem danh sách Task theo tuần/Project/người thực hiện.
- Cập nhật các trạng thái:
  - To Do
  - In Progress
  - Done
- Gửi báo cáo Task:
  - Thành viên upload file (PDF, DOCX, hình ảnh, v.v.) làm kết quả hoàn thành Task.
  - Có thể thêm nhận xét nội dung hoàn thành.

### c. Hạn chế quyền sửa Task

- Thành viên chỉ có thể chỉnh sửa các Task do chính họ tạo.
- Task do LEADER/VICE\_LEADER giao → thành viên chỉ được cập nhật trạng thái, không được sửa nội dung Task.

### 2.1.4. Hệ thống thông báo (Notification System)

- Gửi thông báo real-time (WebSocket hoặc push notification/email) cho người dùng khi:
  - Được giao Task mới.
  - Có thay đổi quyền trong Project (được gán làm VICE\_LEADER, LEADER).
  - Task sắp đến deadline (ví dụ trước 24h, trước 1h).
- Có thể xem lịch sử thông báo trong giao diện người dùng.

### 2.1.5. Chức năng lọc (Filtering & Searching)

#### a. Lọc Task theo thời gian

- Cho phép lọc Task trong khoảng thời gian nhất định (từ ngày → đến ngày).
- Có thể lọc theo:
  - Deadline: các Task có deadline nằm trong khoảng.
  - Ngày tạo Task.
  - Ngày hoàn thành Task.

#### b. Lọc Task theo thành viên

- Lọc các Task được giao cho một hoặc nhiều thành viên cụ thể.
- Phân biệt rõ:
  - Task do thành viên tự tạo.
  - Task do LEADER/VICE\_LEADER giao cho thành viên.

#### c. Kết hợp điều kiện lọc

- Hỗ trợ lọc kết hợp điều kiện: theo thời gian + thành viên + trạng thái (To Do/In Progress/Done).
- Hỗ trợ tìm kiếm theo tên Task, từ khóa mô tả.

### 2.1.6. Xuất báo cáo (Export Report)

#### a. Xuất báo cáo tổng quan Project

- Xuất dưới định dạng Excel hoặc PDF.
- Nội dung bao gồm:
  - Thông tin chung của Project (tên, mô tả, thời gian, LEADER).
  - Danh sách Task (tên, người thực hiện, trạng thái, deadline).
  - Tổng số Task theo trạng thái.
  - Thống kê hiệu suất: phần trăm Task hoàn thành đúng hạn / trễ hạn.

#### b. Xuất báo cáo hiệu suất theo thành viên

- Tập trung phân tích hiệu suất làm việc cá nhân, gồm:
  - Tổng số Task được giao.
  - Tỷ lệ hoàn thành đúng hạn / trễ hạn.
  - Tỷ lệ bị từ chối
  - Đánh giá thành viên
- Có thể chọn lọc theo khoảng thời gian và Project cụ thể.

#### c. Tùy chọn định dạng và tải về

- Người dùng chọn định dạng: Excel (XLSX) hoặc PDF.
- Các báo cáo có thể được tải về từ UI (giao diện).

## **2.2. Yêu cầu phi chức năng**

### **2.2.1. Hiệu năng (Performance)**

- Truy xuất danh sách Project, Task, User trong vòng  $< 1s$ .
- Tạo, cập nhật, xóa dữ liệu với độ trễ tối đa  $< 2s$ .
- Hệ thống hoạt động trơn tru với  $> 1000$  người dùng đồng thời.

### **2.2.2. Bảo mật (Security)**

- Xác thực người dùng bằng JWT/OAuth2.
- Phân quyền rõ ràng theo vai trò (Role-based Access Control - RBAC).
- Mã hóa mật khẩu bằng BCrypt hoặc Argon2.
- Giới hạn quyền upload file (kích thước, định dạng).
- Xác thực người dùng trước khi thay đổi dữ liệu quan trọng (chuyển quyền, xóa project...).

### **2.2.3. Khả năng mở rộng (Scalability)**

- Thiết kế kiến trúc dễ dàng mở rộng theo chiều ngang (scale-out).
- Tách biệt các module (Authentication, Project, Task, Notification) nếu dùng Microservices.

- Có thể tích hợp thêm các chức năng như báo cáo biểu đồ, thống kê, API bên thứ ba trong tương lai.

#### **2.2.4. Tính ổn định (Reliability)**

- Tự động retry nếu mất kết nối tạm thời.
- Cơ chế backup dữ liệu định kỳ.
- Ghi log toàn bộ thao tác quan trọng (audit log): tạo/xóa Task, chuyển quyền, upload file.

#### **2.2.5. Dễ sử dụng (Usability)**

- Giao diện thân thiện, trực quan, phù hợp với người không chuyên IT.
- Có giao diện hiển thị Task theo dạng lịch (calendar view hoặc kanban board).
- Tối ưu hiển thị cho desktop, tablet và mobile.

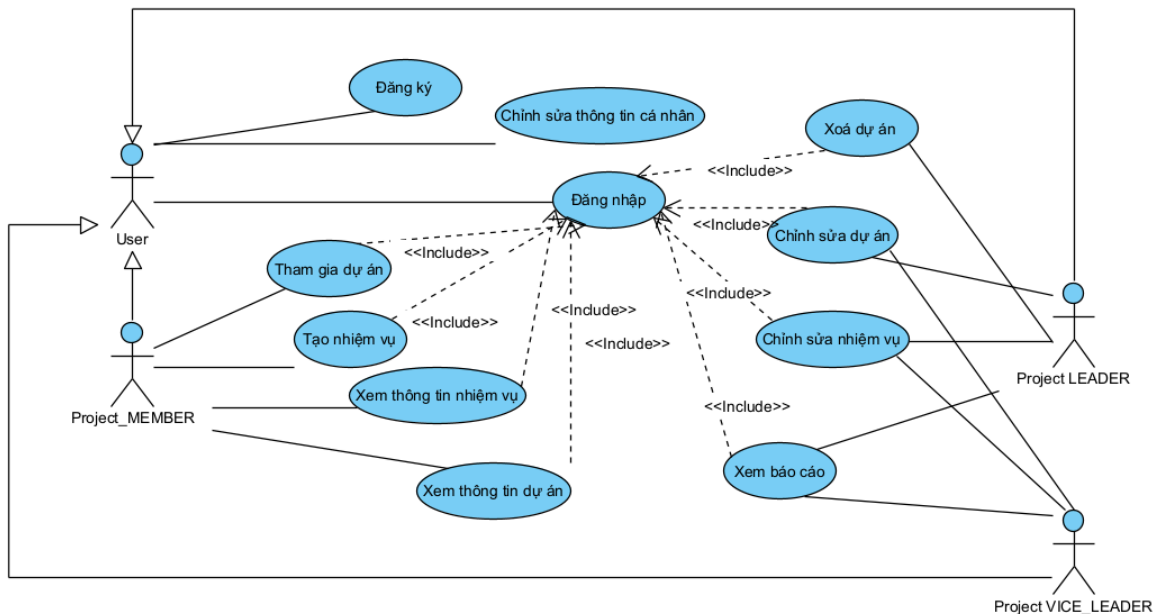
#### **2.2.6. Khả dụng cao (Availability)**

- Hệ thống có thể hoạt động liên tục 24/7 với thời gian downtime < 0.01%.
- Các dịch vụ quan trọng (login, phân quyền, upload file) phải có cơ chế dự phòng.

#### **2.2.7. Khả năng bảo trì (Maintainability)**

- Mã nguồn được tổ chức rõ ràng theo mô hình MVC hoặc Service-Oriented Architecture.
- Tài liệu hướng dẫn API đầy đủ.

## 2.3. Đặc tả các trường hợp sử dụng



### 2.3.1. Use Case: Đăng ký

Tác nhân: User

Mô tả: Cho phép người dùng tạo tài khoản mới.

Bao gồm:

Điều kiện tiên quyết: Người dùng chưa có tài khoản.

Luồng chính:

- Người dùng nhập thông tin đăng ký (họ tên, email, mật khẩu...).
- Hệ thống kiểm tra thông tin hợp lệ.
- Hệ thống tạo tài khoản mới và thông báo thành công.

Luồng phụ:

- Thông tin không hợp lệ → thông báo lỗi.
- Email đã tồn tại → yêu cầu nhập email khác.

### **2.3.2. Use Case: Đăng nhập**

Tác nhân: User, Project\_MEMBER, Project\_LEADER, Project\_VICE\_LEADER

Mô tả: Cho phép người dùng truy cập hệ thống với tài khoản đã có.

Bao gồm:

Điều kiện tiên quyết: Có tài khoản hợp lệ.

Luồng chính:

- Người dùng nhập email và mật khẩu.
- Hệ thống kiểm tra thông tin.
- Hệ thống xác thực và chuyển đến trang chính.

Luồng phụ:

- Email hoặc mật khẩu sai → báo lỗi.

### **2.3.3. Use Case: Chỉnh sửa thông tin cá nhân**

Tác nhân: User

Mô tả: Cho phép người dùng thay đổi thông tin cá nhân.

Bao gồm: Đăng nhập

Điều kiện tiên quyết: Đã đăng nhập.

Luồng chính:

- Người dùng truy cập trang thông tin cá nhân.
- Chỉnh sửa các trường thông tin.
- Hệ thống lưu thay đổi.

Luồng phụ:

- Thông tin không hợp lệ → báo lỗi.

#### **2.3.4. Use Case: Tham gia dự án**

Tác nhân: Project\_MEMBER

Mô tả: Thành viên tham gia vào một dự án đang có.

Bao gồm: Đăng nhập

Điều kiện tiên quyết: Đã đăng nhập và chưa thuộc dự án.

Luồng chính:

- Thành viên chọn dự án muốn tham gia.
- Hệ thống thêm thành viên vào danh sách dự án.

#### **2.3.5. Use Case: Tạo nhiệm vụ**

Tác nhân: Project\_MEMBER, Project\_LEADER, Project\_VICE\_LEADER

Mô tả: Tạo nhiệm vụ mới trong dự án.

Bao gồm: Đăng nhập

Điều kiện tiên quyết: Đã đăng nhập và có quyền tạo nhiệm vụ.

Luồng chính:

- Chọn dự án → Tạo nhiệm vụ.
- Nhập thông tin nhiệm vụ (tiêu đề, mô tả, deadline).
- Lưu nhiệm vụ.

Luồng phụ:



- Thông tin không hợp lệ → báo lỗi.

#### **2.3.6. Use Case: Xem thông tin nhiệm vụ**

Tác nhân: Tất cả vai trò dự án

Mô tả: Hiển thị chi tiết nhiệm vụ.

Bao gồm: Đăng nhập

Điều kiện tiên quyết: Đã đăng nhập.

Luồng chính:

- Người dùng chọn nhiệm vụ.
- Hệ thống hiển thị thông tin.

#### **2.3.7. Use Case: Xem thông tin dự án**

Tác nhân: Tất cả vai trò dự án

Mô tả: Hiển thị chi tiết dự án.

Bao gồm: Đăng nhập

Điều kiện tiên quyết: Đã đăng nhập.

Luồng chính:

- Người dùng chọn dự án.
- Hệ thống hiển thị thông tin.

#### **2.3.8. Use Case: Xóa dự án**

Tác nhân: Project\_LEADER

Mô tả: Xóa dự án khỏi hệ thống.

Bao gồm: Đăng nhập

Điều kiện tiên quyết: Đã đăng nhập, có quyền leader và dự án không có nhiệm vụ hoạt động.

Luồng chính:

- Leader chọn dự án cần xóa.
- Hệ thống xác nhận và xóa.

### **2.3.9. Use Case: Chỉnh sửa dự án**

Tác nhân: Project\_LEADER, Project\_VICE\_LEADER

Mô tả: Cập nhật thông tin dự án.

Bao gồm: Đăng nhập

Điều kiện tiên quyết: Đã đăng nhập và có quyền chỉnh sửa.

Luồng chính:

- Chọn dự án → Chỉnh sửa.
- Cập nhật thông tin.
- Lưu thay đổi.

### **2.3.10. Use Case: Chỉnh sửa nhiệm vụ**

Tác nhân: Project\_LEADER, Project\_VICE\_LEADER

Mô tả: Cập nhật thông tin nhiệm vụ.

Bao gồm: Đăng nhập

Điều kiện tiên quyết: Đã đăng nhập và có quyền chỉnh sửa.

Luồng chính:

- Chọn nhiệm vụ → Chính sửa.
- Cập nhật thông tin.
- Lưu thay đổi.

#### **2.3.11. Use Case: Xem báo cáo**

Tác nhân: Project\_LEADER, Project\_VICE\_LEADER

Mô tả: Xem báo cáo tiến độ, kết quả dự án.

Bao gồm: Đăng nhập

Điều kiện tiên quyết: Đã đăng nhập và có quyền xem báo cáo.

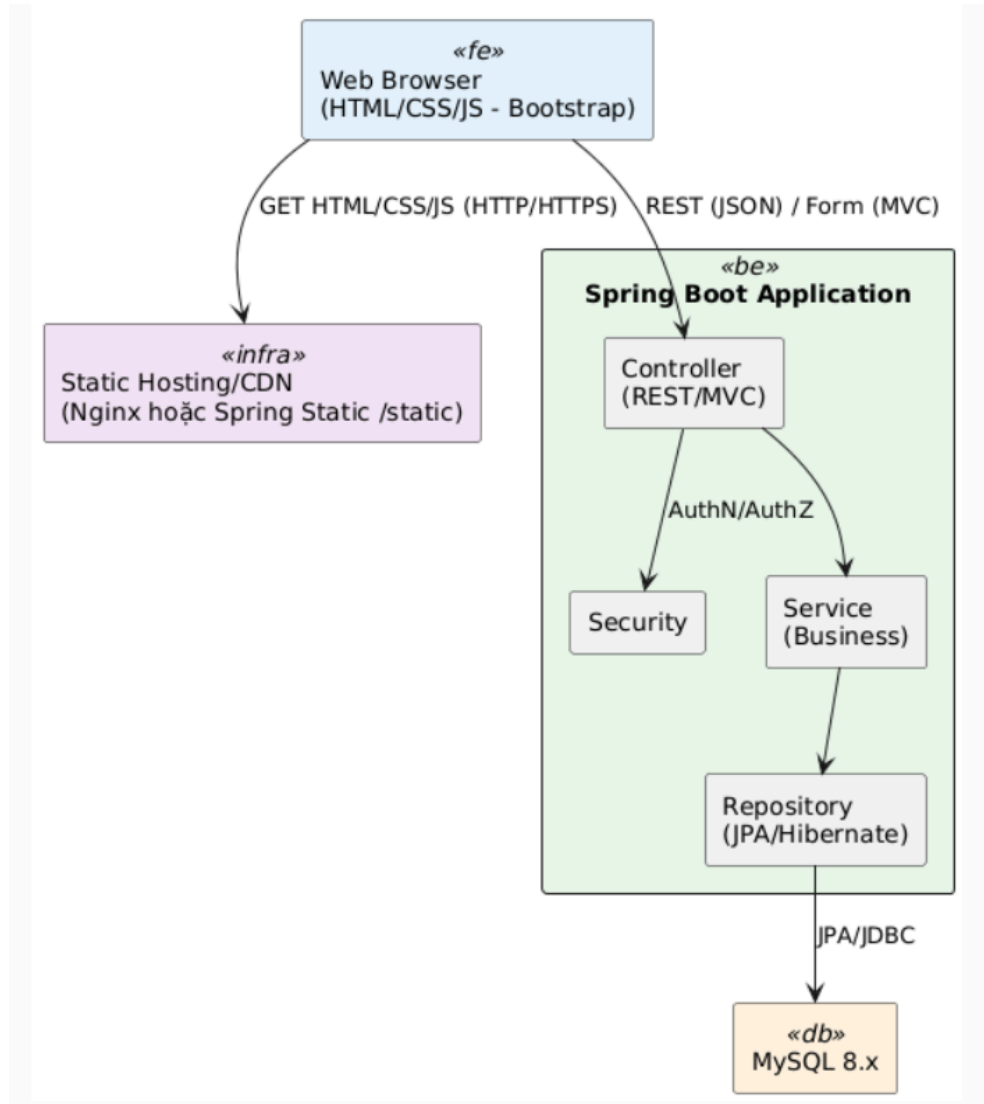
Luồng chính:

- Chọn dự án → Xem báo cáo.
- Hệ thống hiển thị báo cáo.

## Chương III: Thiết kế hệ thống

### 3.1 Kiến trúc tổng thể

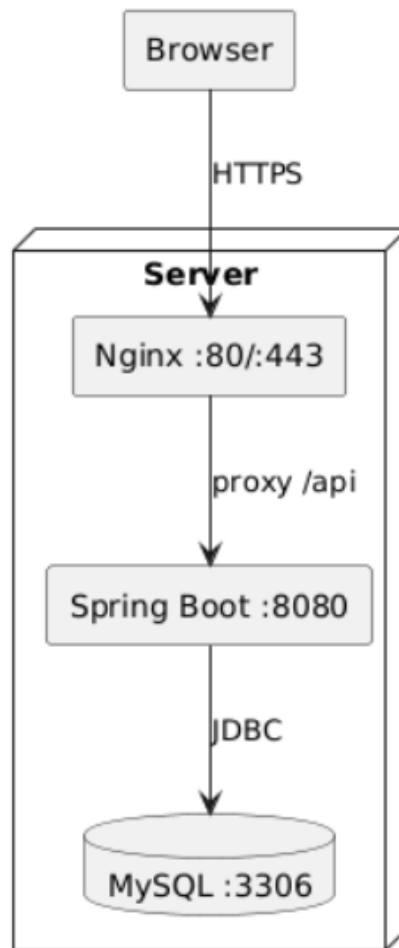
Sử dụng Kiến trúc Client-Server để phát triển hệ thống.



- Client (HTML + Bootstrap + JS):
  - Giao diện hiển thị ở trình duyệt.
  - Gửi request thông qua form hoặc fetch/ajax tới Spring Boot (HTTP POST/GET).
- Server (Spring Boot):
  - Nhận request từ Client.
  - Xử lý logic nghiệp vụ ở tầng Service.
  - Giao tiếp với MySQL qua JPA/Hibernate.

- Database (MySQL):
  - Lưu thông tin người dùng, project, task, phân quyền, lịch sử task, file metadata...

### 3.2. Mô hình triển khai

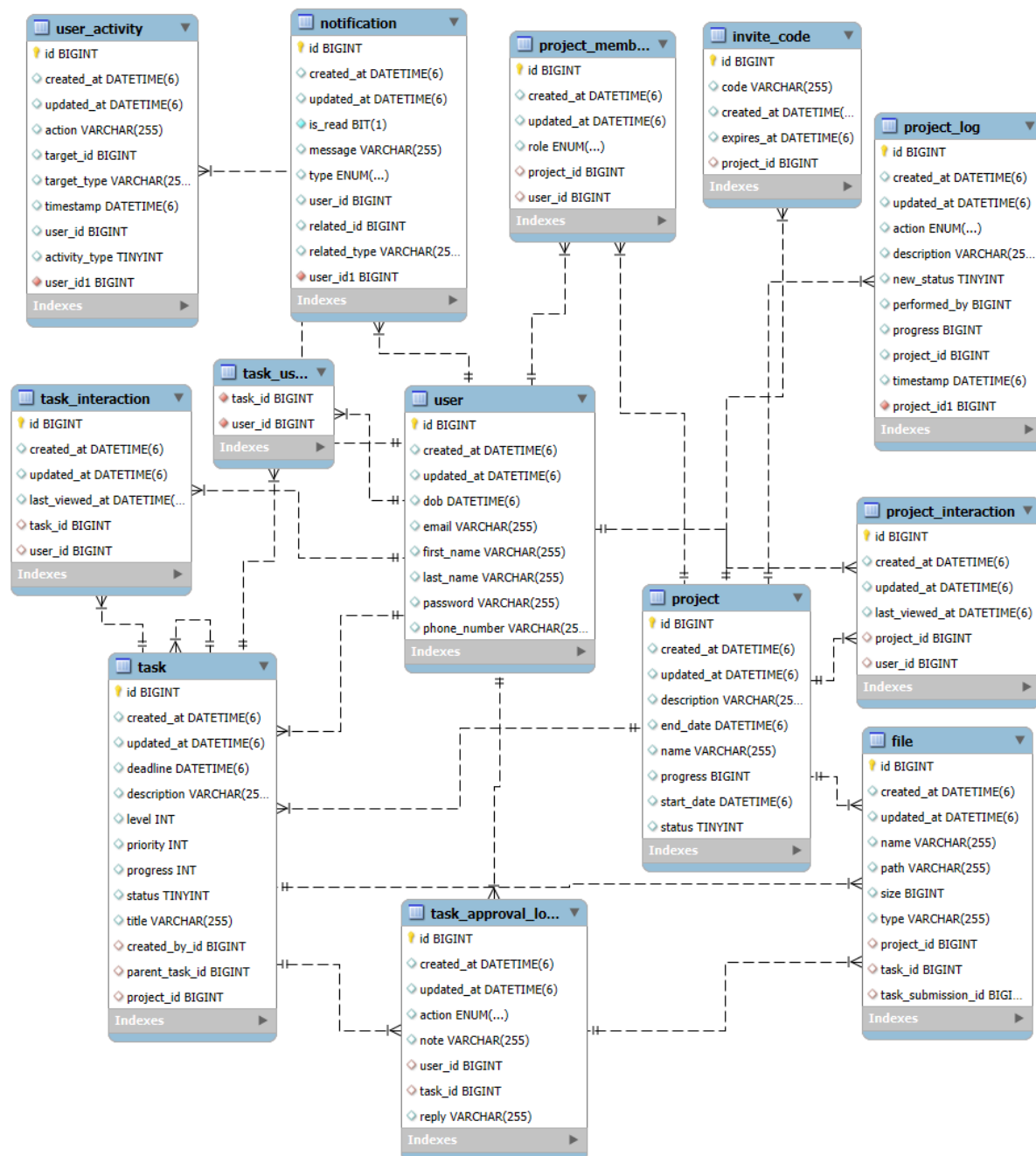


Mô hình triển khai tối giản gồm 3 thành phần chính:

- Browser (Client)
  - Người dùng truy cập ứng dụng thông qua trình duyệt.
  - Gửi yêu cầu HTTP/HTTPS tới máy chủ.
  - Nhận phản hồi là dữ liệu (JSON, HTML) hoặc file tĩnh (CSS, JS, hình ảnh).
- Nginx (Reverse Proxy - tùy chọn)

- Đứng giữa Browser và Spring Boot.
- Nhiệm vụ:
  - Chuyển tiếp request từ client tới Spring Boot qua HTTP nội bộ.
  - Phục vụ file tĩnh (HTML/CSS/JS, hình ảnh) nhanh hơn so với Java.
  - Bảo mật: hỗ trợ SSL/TLS, chống truy cập trái phép, giới hạn request.
  - Tối ưu hiệu năng: cache file tĩnh, nén dữ liệu (gzip, brotli).
- Spring Boot Application (Port 8080)
  - Xử lý toàn bộ nghiệp vụ của hệ thống.
  - Tầng Controller nhận request → Service xử lý → Repository truy vấn MySQL.
  - Trả dữ liệu JSON hoặc HTML (nếu dùng MVC).
  - Có thể tự phục vụ file tĩnh nếu bỏ qua Nginx.
- MySQL Database (Port 3306)
  - Lưu trữ dữ liệu của hệ thống (người dùng, sản phẩm, đơn hàng...).
  - Spring Boot kết nối qua JDBC/HikariCP.

### **3.3. Thiết kế cơ sở dữ liệu**



- Bảng: user\_activity

Cột	Kiểu dữ liệu	Ràng buộc
id	BIGINT	PK, AUTO_INCREMENT

created_at	DATETIME(6)	
updated_at	DATETIME(6)	
action	VARCHAR(255)	
target_id	BIGINT	
target_type	VARCHAR(255)	
timestamp	DATETIME(6)	
user_id	BIGINT	FK → user(id)
activity_type	TINYINT	CHECK (0–9)

- Bảng: notification

Cột	Kiểu dữ liệu	Ràng buộc
id	BIGINT	PK, AUTO_INCREMENT
created_at	DATETIME(6)	
updated_at	DATETIME(6)	
is_read	BIT(1)	NOT NULL
message	VARCHAR(255)	



type	ENUM(...)	
user_id	BIGINT	
related_id	BIGINT	
related_type	VARCHAR(255)	

- Bảng: project\_member

Cột	Kiểu dữ liệu	Ràng buộc
id	BIGINT	PK, AUTO_INCREMENT
created_at	DATETIME(6)	
updated_at	DATETIME(6)	
role	ENUM(...)	
project_id	BIGINT	FK → project(id)
user_id	BIGINT	FK → user(id)

- Bảng: invite\_code

Cột	Kiểu dữ liệu	Ràng buộc
id	BIGINT	PK, AUTO_INCREMENT

code	VARCHAR(255)	
created_at	DATETIME(6)	
expires_at	DATETIME(6)	
project_id	BIGINT	FK → project(id)

- Bảng: project\_log

Cột	Kiểu dữ liệu	Ràng buộc
id	BIGINT	PK, AUTO_INCREMENT
created_at	DATETIME(6)	
updated_at	DATETIME(6)	
action	ENUM(...)	
description	VARCHAR(255)	
new_status	TINYINT	CHECK (0–5)
performed_by	BIGINT	
progress	BIGINT	
project_id	BIGINT	FK → project(id)

timestamp	DATETIME(6)	
-----------	-------------	--

- Bảng: project\_interaction

Cột	Kiểu dữ liệu	Ràng buộc
id	BIGINT	PK, AUTO_INCREMENT
created_at	DATETIME(6)	
updated_at	DATETIME(6)	
last_viewed_at	DATETIME(6)	
project_id	BIGINT	FK → project(id)
user_id	BIGINT	FK → user(id)

- Bảng: project

Cột	Kiểu dữ liệu	Ràng buộc
id	BIGINT	PK, AUTO_INCREMENT
created_at	DATETIME(6)	
updated_at	DATETIME(6)	
description	VARCHAR(255)	

end_date	DATETIME(6)	
name	VARCHAR(255)	
progress	BIGINT	
start_date	DATETIME(6)	
status	TINYINT	CHECK (0–5)

- Bảng: file

Cột	Kiểu dữ liệu	Ràng buộc
id	BIGINT	PK, AUTO_INCREMENT
created_at	DATETIME(6)	
updated_at	DATETIME(6)	
name	VARCHAR(255)	
path	VARCHAR(255)	
size	BIGINT	
type	VARCHAR(255)	
project_id	BIGINT	FK → project(id)

task_id	BIGINT	FK → task(id)
task_submission_id	BIGINT	FK → task_approval_logs(id)

- Bảng: task\_interaction

Cột	Kiểu dữ liệu	Ràng buộc
id	BIGINT	PK, AUTO_INCREMENT
created_at	DATETIME(6)	
updated_at	DATETIME(6)	
last_viewed_at	DATETIME(6)	
task_id	BIGINT	FK → task(id)
user_id	BIGINT	FK → user(id)

- Bảng: task\_user

Cột	Kiểu dữ liệu	Ràng buộc
task_id	BIGINT	FK → task(id)
user_id	BIGINT	FK → user(id)

- Bảng: user

Cột	Kiểu dữ liệu	Ràng buộc
id	BIGINT	PK, AUTO_INCREMENT
created_at	DATETIME(6)	
updated_at	DATETIME(6)	
dob	DATETIME(6)	
email	VARCHAR(255)	UNIQUE
first_name	VARCHAR(255)	
last_name	VARCHAR(255)	
password	VARCHAR(255)	
phone_number	VARCHAR(255)	

- Bảng: task

Cột	Kiểu dữ liệu	Ràng buộc
id	BIGINT	PK, AUTO_INCREMENT
created_at	DATETIME(6)	
updated_at	DATETIME(6)	

deadline	DATETIME(6)	
description	VARCHAR(255)	
level	INT	
priority	INT	
progress	INT	
status	TINYINT	CHECK (0–5)
title	VARCHAR(255)	
created_by_id	BIGINT	FK → user(id)
parent_task_id	BIGINT	FK → task(id)
project_id	BIGINT	FK → project(id)


- Bảng: task\_approval\_logs

Cột	Kiểu dữ liệu	Ràng buộc
id	BIGINT	PK, AUTO_INCREMENT
created_at	DATETIME(6)	
updated_at	DATETIME(6)	

action	ENUM(...)	
note	VARCHAR(255)	
user_id	BIGINT	FK → user(id)
task_id	BIGINT	FK → task(id)

### 3.4. Thiết kế giao diện

- Giao diện đăng nhập

 TEAManage

Đăng nhập
Đăng ký

Email

Mật khẩu

Quên mật khẩu

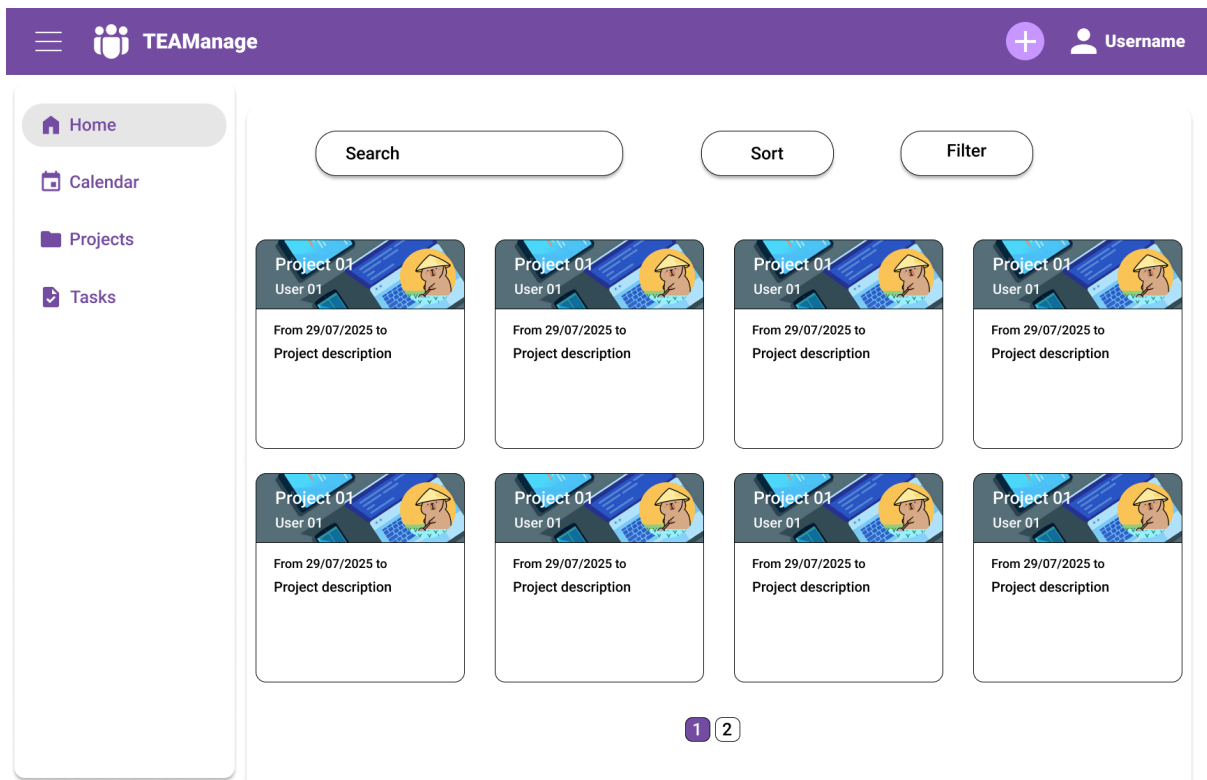
Đăng nhập

- Giao diện đăng ký

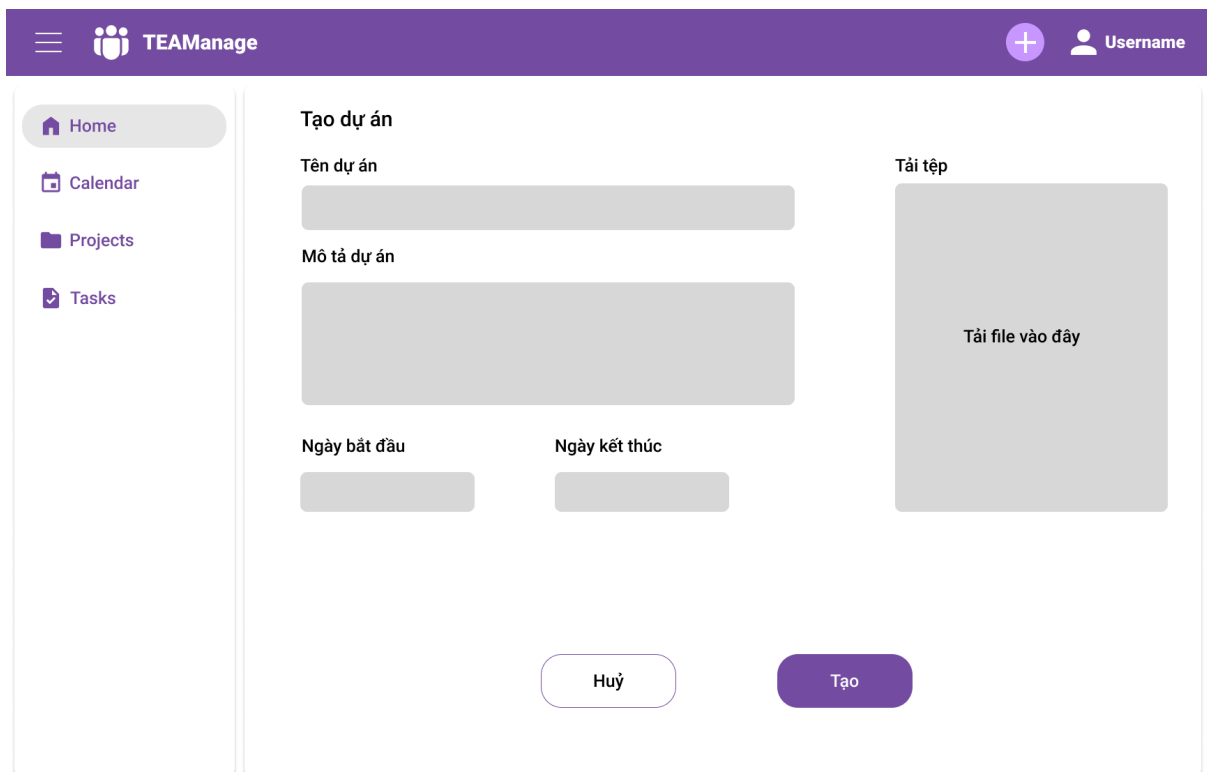


- Giao diện index

- Giao diện danh sách dự án



- Giao diện tạo dự án



- Giao diện thông tin dự án

TEAManage

+

Username

Dự án TEAManage

Mô tả dự án

Dự án xây dựng và phát triển hệ thống TEAManage

Ngày bắt đầu: 01/01/2025

Ngày kết thúc: 01/06/2025

Trạng thái dự án: Đang hoạt động

Hoạt động của dự án

Mô phỏng github

Nhiệm vụ	Thành viên	Tệp tin	Cài đặt
----------	------------	---------	---------

- Giao diện danh sách nhiệm vụ

TEAManage

+

Username

Home

Calendar

Projects

Tasks

Search

Sort

Filter

Task 1

Task 1.1

Task 1.1.1

Task 2

- Giao diện tạo dự án

TEAManage

+

Username

Tạo nhiệm vụ

Chọn dự án

Chọn nhiệm vụ lớn

Tiêu đề

Mô tả

Hạn chót

Chọn thành viên

Hủy

Tạo

Tải tệp

Tải file vào đây

- Giao diện thông tin nhiệm vụ

TEAManage

+

Username

Nhiệm vụ TEAManage

Mô tả Nhiệm vụ

Hạn chót: 01/01/2025

Trạng thái dự án: Đang hoạt động

Nhiệm vụ liên quan

Thành viên

Tệp tin

Cài đặt

### **3.5. Công cụ và môi trường phát triển**

#### **3.5.1. IDE (Integrated Development Environment)**

- IntelliJ IDEA Ultimate / Community – IDE chính để lập trình Java/Spring Boot.
- Visual Studio Code – Hỗ trợ chỉnh sửa file HTML, CSS, JavaScript.
- MySQL Workbench / HeidiSQL – Quản lý và thiết kế cơ sở dữ liệu MySQL.

#### **3.5.2. Version Control**

- Git – Hệ thống quản lý phiên bản phân tán.

#### **3.5.3. Công cụ xây dựng & quản lý phụ thuộc**

- Maven hoặc Gradle – Build project Spring Boot, quản lý dependencies.

#### **3.5.4. Cơ sở dữ liệu**

- MySQL 8.x – Hệ quản trị cơ sở dữ liệu quan hệ.
- Công cụ quản lý:
  - HeidiSQL: kết nối và quản trị DB.

#### **3.5.5. Môi trường chạy ứng dụng**

- Java JDK 21 – Phiên bản Java tương thích Spring Boot.
- Spring Boot 3.x – Backend framework chính.
- Bootstrap 5.x – CSS framework cho frontend.
- Docker – Đóng gói ứng dụng và database.
- Nginx – Reverse proxy và phục vụ file tĩnh.

### **3.5.6. Công cụ hỗ trợ nhóm**

- Postman – Test API.
- Figma – Thiết kế mockup giao diện.

## **Chương IV: Kế hoạch thực hiện**

### **Giai đoạn 1 – Khởi động & phân tích yêu cầu**

**Thời gian:** 04/07 – 10/07

**Nhiệm vụ:**

- Nghiên cứu và phân tích yêu cầu đề tài.
- Viết và nộp đề cương đề tài.
- Trao đổi với mentor để thống nhất phạm vi và mục tiêu.

### **Giai đoạn 2 – Thiết kế hệ thống & lập trình backend cơ bản**

**Thời gian:** 10/07 – 17/07

**Nhiệm vụ:**

- Thiết kế CSDL (ERD, bảng, ràng buộc).
- Thiết kế REST API endpoints.
- Lập trình backend cho các chức năng chính (Spring Boot + MySQL).
- Test đơn giản các API.
- Báo cáo tiến độ lần 1.

### **Giai đoạn 3 – Hoàn thiện backend & điều chỉnh**

**Thời gian:** 17/07 – 24/07

**Nhiệm vụ:**

- Chỉnh sửa backend theo phản hồi mentor.
- Hoàn thiện API, xử lý bảo mật, phân quyền.
- Bổ sung unit test hoặc test thủ công.
- Báo cáo tiến độ lần 2.

### **Giai đoạn 4 – Phát triển frontend & tích hợp**

**Thời gian:** 24/07 – 31/07

**Nhiệm vụ:**

- Kiểm tra lại toàn bộ API.
- Lập trình frontend (HTML, CSS, Bootstrap, JS).
- Tích hợp frontend với backend qua API.
- Báo cáo tiến độ lần 3.

### **Giai đoạn 5 – Hoàn thiện sản phẩm & bàn giao**

**Thời gian:** 31/07 – 11/08

**Nhiệm vụ:**

- Chỉnh sửa, tối ưu sản phẩm theo góp ý mentor.
- Hoàn thiện giao diện và tính năng.
- Kiểm thử toàn bộ hệ thống.
- Viết báo cáo tổng kết.
- Nộp sản phẩm cuối cùng.



## **Chương V: Kết luận và định hướng phát triển trong tương lai**

### **5.1. Kết luận**

Trong quá trình thực hiện đề tài, tôi đã:

- Phát triển đầy đủ Backend và Frontend cho hệ thống quản lý thời gian làm việc và báo cáo công việc theo tuần.
- Triển khai ứng dụng thành công bằng Docker, giúp dễ dàng đóng gói và triển khai trên nhiều môi trường.
- Hoàn thiện các chức năng chính đáp ứng yêu cầu cơ bản của đề tài.

Tuy nhiên, sản phẩm vẫn còn một số hạn chế:

- Giao diện chưa hoàn toàn đúng với thiết kế ban đầu, một số phần hiển thị chưa đồng nhất.
- API chưa được tối ưu về hiệu năng và chưa chuẩn REST ở một số điểm.
- Chưa có cơ chế tự động hóa báo cáo hoặc tích hợp công nghệ AI để hỗ trợ lập kế hoạch.

### **5.2. Hướng phát triển**

Trong thời gian tới, tôi dự định:

- 1. Hoàn thiện giao diện người dùng (UI/UX)**
  - Sửa các lỗi giao diện còn tồn tại.
  - Đồng bộ toàn bộ giao diện với thiết kế mockup.
  - Cải thiện khả năng responsive trên thiết bị di động.
- 2. Tối ưu Backend**
  - Chuẩn hóa API theo RESTful.
  - Tối ưu truy vấn cơ sở dữ liệu để tăng hiệu năng.
  - Tích hợp cache (Redis) cho các API có tần suất gọi cao.

### **3. Tích hợp quy trình báo cáo tự động**

- Phát triển chức năng tự động tổng hợp và gửi báo cáo tuần qua email/PDF.
- Cho phép cấu hình thời gian gửi báo cáo theo nhu cầu.

### **4. Ứng dụng AI để lập kế hoạch**

- Sử dụng AI để phân tích dữ liệu công việc và đề xuất kế hoạch làm việc hợp lý.
- Gợi ý ưu tiên công việc dựa trên deadline, độ khó và khối lượng.
- Tự động tạo kế hoạch tuần cho từng người dùng dựa trên dữ liệu lịch sử.

### **5. Triển khai hạ tầng và CI/CD**

- Hoàn thiện kịch bản triển khai bằng Docker Compose/Kubernetes cho môi trường production.
- Tích hợp CI/CD để tự động build, test và deploy sản phẩm.

## **Tài liệu tham khảo**

## **Phụ lục**

Toàn bộ mã nguồn của hệ thống được lưu trữ tại:

<https://github.com/haianh3623/TEAManage.git>