

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO NGHIÊN CỨU TÍNH KHẢ THI CỦA DỰ ÁN

Môn học: Nhập môn công nghệ phần mềm

Giáo viên hướng dẫn:	TS Bùi Thị Mai Anh
Nhóm sinh viên thực hiện:	Lê Viết Đức Anh 20226075
	Hồ Tuấn Anh 20226100
	Đào Kiên Cường 20226104
	Nguyễn Tài Hoan 20226106
	Đỗ Đình Trung 20226128

Hà Nội - 2024

MỤC LỤC

I. TÓM TẮT CHUNG VỀ YÊU CẦU CỦA PROJECT	4
II. PHÂN TÍCH YÊU CẦU SƠ BỘ	4
III. QUY TRÌNH PHÁT TRIỂN	6
IV. MỘT SỐ KẾT QUẢ DỰ KIẾN	8
V. TÍNH KHẢ THI VỀ MẶT KỸ THUẬT	10
VI. PHÂN TÍCH RỦI RO	12
VIII. CONCLUSION/ KẾT LUẬN	14

I. TÓM TẮT CHUNG VỀ YÊU CẦU CỦA PROJECT

Ứng dụng quản lý dự án nhóm được đề xuất thiết kế dành cho các bạn sinh viên, nhân viên, quản lý, freelancer muốn làm dự án và muốn quản lý dự án một cách hiệu quả nhất, cũng như có thể xem được hiệu suất, tiến độ của dự án trong suốt thời gian làm dự án. User của ứng dụng sẽ là sinh viên, nhân viên, quản lý, freelancer, hoặc những người dùng cần quản lý và theo dõi tiến độ dự án cũng như tham gia và nhận các nhiệm vụ trong dự án. Client sẽ là các doanh nghiệp và tổ chức.

Mục đích phát triển nhằm đáp ứng nhu cầu, phân chia công việc và theo dõi tiến độ trong các dự án. Giúp người đứng đầu có thể giám sát tổng quan được hiệu suất và tiến độ của các nhân viên trong dự án.

Hỗ trợ thực hiện nghiệp vụ lập kế hoạch dự án, theo dõi tiến độ và hiệu suất của dự án, báo cáo tiến độ và kết quả của dự án.

Mục tiêu sau khi phát triển cải thiện quy trình quản lý dự án, giúp các nhóm dự án làm việc hiệu quả hơn. Tăng cường khả năng giám sát tiến độ, hiệu suất từ phía quản lý.

II. PHÂN TÍCH YÊU CẦU SƠ BỘ

A. Tổng quan về ứng dụng:

1. Mục tiêu:

- Tạo một trang web với hệ thống quản lý dự án để hỗ trợ các nhóm dự án trong việc lập kế hoạch, phân phối công việc, theo dõi tiến độ và hiệu suất một cách hiệu quả.
- Những mục tiêu về chức năng của hệ thống có thể kể đến như:
 - o Người quản lý dự án có thể tạo và quản lý các dự án, đồng thời quản lý các thành viên tham gia dự án cũng như nhiệm vụ của từng thành viên.
 - o Các thành viên trong dự án có thể cùng nhau theo dõi tiến độ dự án, theo dõi và cập nhật các nội dung công việc cũng như cập nhật tiến độ của cá nhân.
 - o Các thành viên sẽ được nhắc nhở về các mốc deadline quan trọng cũng như xem được lịch trình làm việc cụ thể của bản thân.

2. Quy trình nghiệp vụ:

- Trước đây: Nhóm quản lý dự án nếu muốn quản lý dự án thì dùng các công cụ thủ công như Excel, Sheet để phân chia các công việc cho từng người khác nhau, gây khó khăn trong việc tạo bảng quản lý cũng như trong việc theo dõi tiến độ và hiệu suất của các thành viên trong nhóm dự án.
- Sau khi có ứng dụng: Quản lý dự án chỉ cần tạo dự án trên ứng dụng, và phân chia công việc cho các thành viên trong nhóm dự án. Sau đó, các thành viên trong nhóm dự án sẽ báo cáo tiến độ tương ứng với từng nhiệm vụ mà quản lý đã giao. Cuối cùng, những thông tin như tiến độ và hiệu suất sẽ được hiển thị cho quản lý dự án.

3. Vai trò và trách nhiệm của người dùng:

- Hệ thống sẽ có 2 loại người dùng chính:
 - o Quản lý dự án
 - o Thành viên dự án

- Quyền của các người dùng:
 - o Quản lý dự án: Tạo và xoá các dự án, thêm thành viên cho dự án, tạo các nhiệm vụ cho thành viên của dự án, xem được báo cáo hiệu suất và tiến độ của dự án.
 - o Thành viên dự án: Xem được các nhiệm vụ, xem được hiệu suất và tiến độ của bản thân, báo cáo về các nhiệm vụ đã được giao, deadline của từng nhiệm vụ.
- 4. Tương tác với hệ thống khác:
 - Mailservice API: Để thông báo tới thành viên của dự án khi đã sắp hết hạn của nhiệm vụ nhưng vẫn không có báo cáo.
- 5. Những cân nhắc khi triển khai hệ thống:
 - Đảm bảo tính bảo mật của dự án, bảo mật thông tin của khách hàng.
 - Xử lý số liệu một cách chính xác, không để xảy ra tình trạng tính toán sai trong báo cáo tiến độ và hiệu suất.
 - Thiết kế hệ thống phù hợp để có thể scale up sau này khi có nhiều người sử dụng hơn.

B. Yêu cầu chức năng:

1. Tuyên bố về chức năng:
 - Nhóm chức năng cho người quản lý dự án:
 - o Quản lý dự án: Xem/thêm/sửa/xoá dự án.
 - o Quản lý nhiệm vụ: Xem/thêm/sửa/xoá nhiệm vụ.
 - o Lưu trữ file cho từng dự án.
 - o Các thao tác về nhiệm vụ: Phân loại nhiệm vụ theo tag, gán nhiệm vụ cho thành viên, đặt deadline cho nhiệm vụ, sao chép nhiệm vụ, chọn mức ưu tiên cho nhiệm vụ, thay đổi các trạng thái của nhiệm vụ,....
 - Nhóm chức năng cho thành viên tham gia dự án:
 - o Xem dự án, xem nhiệm vụ.
 - o Báo cáo tiến độ: Cập nhật tiến độ công việc đã làm xong.
 - o Thông báo khi gần tới hạn và quá hạn của nhiệm vụ.
 - o Lưu trữ file cho từng dự án.
2. Khả năng bảo mật và người dùng:
 - Phân quyền thành 2 loại người dùng chính:
 - o Quản lý dự án.
 - o Thành viên dự án.
 - Bảo mật:
 - o Mã hoá mật khẩu của khách hàng khi lưu trữ trong database.
 - o Chống lại tấn công SQL Injection
3. Tính năng tùy chọn:
 - Tìm kiếm nhiệm vụ theo tên, theo tag, theo từ khóa hoặc theo người được phân công, lọc nhiệm vụ,...
 - Quản lý các file được lưu trữ
4. Phạm vi dự án:
 - Mục tiêu dự án:

- Xây dựng một website quản lý dự án cho phép người dùng tạo, theo dõi, và quản lý các dự án một cách hiệu quả.
- Cung cấp các công cụ để phân công nhiệm vụ, theo dõi tiến độ, và báo cáo kết quả.
- Sản phẩm đầu ra:
 - Giao diện người dùng (UI) thân thiện và dễ sử dụng.
 - Chức năng đăng nhập và quản lý người dùng (admin, thành viên dự án).
 - Chức năng tạo và quản lý dự án (thêm, sửa, xóa).
 - Chức năng phân công nhiệm vụ cho các thành viên trong dự án.
 - Các chức năng về thao tác với nhiệm vụ như phân loại, đặt mức ưu tiên, đặt trạng thái tùy chỉnh, đặt deadline,...
 - Hệ thống theo dõi tiến độ và trạng thái của các nhiệm vụ.
 - Tính năng thông báo (notification) qua email khi có thay đổi trạng thái nhiệm vụ hoặc dự án.
 - Lưu trữ file trong từng các dự án
- Giới hạn dự án:
 - Dự án không bao gồm việc phát triển ứng dụng di động.
 - Không bao gồm các tính năng quản lý tài chính hoặc kế toán.
 - Không xử lý các dữ liệu từ bên thứ ba ngoài hệ thống quản lý dự án.
- Yêu cầu dự án:
 - Website cần phải được phát triển bằng HTML, CSS, JavaScript cho phần frontend và Java Spring Framework cho phần backend.
 - Cơ sở dữ liệu sử dụng PostgreSQL.
- Giả định dự án:
 - Người dùng có kiến thức cơ bản về công nghệ thông tin và internet.
 - Hệ thống có thể hoạt động ổn định trên các trình duyệt phổ biến (Chrome, Firefox, Safari).

III. QUY TRÌNH PHÁT TRIỂN

Đối với dự án này, mô hình phát triển mà nhóm sẽ tuân theo là mô hình phát triển phần mềm nhanh (AGILE). Cụ thể, nhóm sẽ sử dụng Scrum management framework. Nhóm đã chọn phương pháp cụ thể này do thời gian để hoàn thiện dự án ngắn (chỉ trong vòng 3,5 tháng) và muốn đưa sản phẩm đến tay người dùng càng nhanh càng tốt. Với phương pháp Agile, dự án sẽ được thực hiện trong các phân đoạn lặp đi lặp lại (Sprint) có khung thời gian ngắn. Trong mỗi phân đoạn này, nhóm sẽ thực hiện đầy đủ các công việc cần thiết như lập kế hoạch, phân tích yêu cầu, thiết kế, triển khai, kiểm thử để cho ra các phần nhỏ của sản phẩm. Product Backlog và Sprint Backlog sẽ được nhóm thiết kế và triển khai, cập nhật thường xuyên trong các Sprint Planning Meeting. Daily Scrum cũng được nhóm thường xuyên tổ chức do các thành viên đã làm việc với nhau qua nhiều dự án. Việc kiểm thử được thực hiện liên tục và thường xuyên trong dự án và do đó, giúp phát hiện và khắc phục sớm các vấn đề. Cuối các phân đoạn, nhóm cho ra các phần nhỏ của sản phẩm cuối cùng, thường là đầy đủ, có khả năng chạy tốt, được kiểm thử cẩn thận và có thể sử dụng, đồng thời là các Sprint Review và Sprint Restrospective để định hướng phát triển cho các Sprint tiếp theo. Theo thời gian,

phân đoạn này tiếp nối phân đoạn kia, các phần chạy được này sẽ được tích lũy, lớn dần lên cho tới khi thỏa mãn toàn bộ yêu cầu của khách hàng. Vì các phân đoạn chỉ kéo dài trong thời gian ngắn nên có thể đáp ứng các thay đổi trong quá trình phát triển dựa theo các phản hồi của khách hàng. Về yêu cầu của khách hàng, nhóm sẽ trực tiếp trao đổi, làm việc với khách hàng để hiểu rõ hơn về cái họ thực sự cần, và để biết yêu cầu nào có độ ưu tiên cao hơn, mang giá trị hơn sớm nhất cho dự án.

Lý do lựa chọn mô hình:

- Cải thiện khả năng hợp tác và giao tiếp giữa các thành viên.
- Tăng cường khả năng làm việc nhóm thông qua các Daily Scrum.
- Giúp các thành viên trong nhóm phản ứng nhanh chóng với những thay đổi và thách thức bất ngờ trong dự án.
- Quy trình lặp đi lặp lại và phản hồi thường xuyên cho phép nhóm thực hiện các điều chỉnh nhỏ thường xuyên thay vì phải đại tu lớn khi kết thúc dự án.
- Sử dụng tài nguyên một cách hiệu quả hơn.
- Phương pháp Scrum áp dụng cách tiếp cận lập kế hoạch đúng lúc giúp các nhóm tránh lãng phí thời gian vào những nhiệm vụ không quan trọng, bám sát các ưu tiên của họ và tạo ra các kết quả nhanh chóng trong suốt dự án.

Dưới đây là phác thảo được đề xuất về quá trình thực hiện và các cột mốc bao gồm những gì nhóm dự kiến hoàn thành ở mỗi giai đoạn.

Process Outline:

Quá trình thực hiện chia làm 3 cột mốc (milestone). Tại mỗi cột mốc, chúng tôi chia nhỏ thành các phân đoạn phát triển (sprint), mỗi sprint kéo dài 2 tuần. Mỗi milestone phải phát hành được sản phẩm để bảo đảm lấy được ý kiến khách hàng, qua được các quy trình phát triển của sản phẩm nhằm rút kinh nghiệm và tránh sự cố sau này.

Milestone 1 (08/10/2024 – 03/11/2024)

Lần Sprint thứ 1:

- Thiết kế database cơ bản.
- Mockups: Giao diện người dùng.

Lần Sprint thứ 2:

- Hoàn thiện chức năng đăng ký và đăng nhập, phân quyền được người dùng.
- Hoàn thiện chức năng quản lý dự án (Xem/thêm/sửa/xóa dự án).

Ở thời gian này, nhóm sẽ chia thành 2 Sprint, nhóm sẽ hoàn thiện mockup để đưa cho khách hàng xem và nhận xét, nếu khách chưa hài lòng hoặc có ý kiến thì sẽ chỉnh sửa kịp thời để không làm tiến độ của khách hàng bị ảnh hưởng.

Sau khi hoàn thiện được 2 chức năng cơ bản của hệ thống (Đăng ký, đăng nhập và quản lý dự án), chúng tôi sẽ tiến hành kiểm tra hệ thống để đảm bảo hệ thống hoạt động tốt, sau đó sẽ phát hành sản phẩm và gửi tới khách hàng. Nhóm sẽ cùng khách hàng trao đổi và tiếp nhận những yêu cầu từ khách hàng nếu có.

Milestone 2 (04/11/2024 – 01/12/2024)

Lần Sprint thứ 1:

- Hoàn thiện chức năng quản lý nhiệm vụ (Xem/thêm/sửa/xoá nhiệm vụ).
- Hoàn thành các nhóm chức năng liên quan đến thao tác với nhiệm vụ (phân loại, đặt mức ưu tiên, đặt trạng thái tùy chỉnh, đặt deadline,...)

Lần Sprint thứ 2:

- Hoàn thiện chức năng cập nhật tiến độ công việc.
- Hoàn thiện chức năng gửi mail để xử lý các nhiệm vụ: thông báo khi sắp đến hạn nhiệm vụ, quá hạn nhiệm vụ, lấy lại mật khẩu.

Sau lần phát hành đầu tiên với khách hàng ở Milestone 1, chúng tôi sẽ sửa đổi những điều khách hàng đã phản hồi. Sau khi sửa đổi, nhóm tiếp tục thực hiện những chức năng đã đề ra ở Milestone 2.

Ở Milestone này, chúng tôi chủ yếu tập trung vào phát triển thêm những tính năng cơ bản của hệ thống bao gồm các chức năng quản lý về nhiệm vụ, chức năng cập nhật tiến độ công việc cho người tham gia dự án, các chức năng liên quan tới mail. Ở Milestone này chúng tôi sẽ chia thành 2 Sprint.

Sau khi hoàn thiện, chúng tôi sẽ release sản phẩm cho khách hàng và nhận phản hồi từ khách hàng.

Milestone 3 (02/12/2024 – 15/12/2024)

- Hoàn thiện chức năng lưu trữ file cho từng dự án.
- Kiểm thử toàn bộ hệ thống.
- Hoàn thiện hệ thống cuối cùng và các tài liệu liên quan.

Sau lần phát hành sản phẩm thứ 2 với khách hàng ở Milestone 2, chúng tôi sẽ chỉnh sửa hệ thống cho phù hợp với yêu cầu của họ. Sau đó nhóm tiếp tục thực hiện những chức năng đã đề ra ở Milestone 3 này.

Ở Milestone này, nhóm sẽ chỉ dùng 1 Sprint, ở giai đoạn này nhóm sẽ hoàn thiện các chức năng còn lại của hệ thống. Đây là các chức năng bổ sung, giúp cho người dùng thuận tiện hơn khi sử dụng.

Cuối cùng, nhóm sẽ tiến hành kiểm thử toàn bộ hệ thống và gửi cho khách hàng. Khách hàng sẽ phản hồi và chúng tôi sẽ sửa đổi để có sản phẩm hoàn thiện nhất bản cho người dùng.

IV. MỘT SỐ KẾT QUẢ DỰ KIẾN

Tài liệu đặc tả (Documents):

- Tài liệu SRS (Software Requirement Specification) mô tả chi tiết về giao diện người dùng, các chức năng chính, các điều kiện tiên quyết, ràng buộc hệ thống và mô tả về cách thức mà phần mềm sẽ hoạt động trong mọi trường hợp khả thi. Đây là một tài liệu quan trọng để đảm bảo các bên liên quan đều nắm bắt trong quá trình phát triển phần mềm.
- Tài liệu FRS (Function Requirement Specification): Đây là tài liệu mô tả, xác định chức năng của hệ thống hoặc thành phần của hệ thống.
- Tài liệu Use case: Thể hiện sự tương tác của người dùng với từng chức năng của phần mềm.
- UI/UX: Tài liệu mô tả thiết kế người dùng và giao diện người dùng.

- Báo cáo (Reports): Các khách hàng cũng sẽ được cung cấp các bản sao của tài liệu, trong đó ghi lại các chi tiết ở mỗi giai đoạn trong quá trình phát triển phần mềm. Các báo cáo tiến độ này cũng sẽ cho phép họ nhận thức rõ ràng về các chi tiết của dự án từ quan điểm của họ.

Milestone 1 (08/10/2024 – 03/11/2024)

Module chuyển giao:

- Database cơ bản: Cơ sở dữ liệu với các bảng người dùng, dự án, và quyền truy cập.
- Mockups giao diện người dùng: Phiên bản thiết kế giao diện, cho phép khách hàng xem trước cách hệ thống sẽ hiển thị và tương tác.
- Chức năng đăng ký, đăng nhập: Phiên bản chức năng để người dùng có thể đăng ký tài khoản, đăng nhập, và thực hiện phân quyền.
- Chức năng quản lý dự án: Module cho phép người dùng xem/thêm/sửa/xóa dự án.

Mục tiêu:

- Nhận phản hồi từ khách hàng về giao diện và tính năng cơ bản của hệ thống.
- Sửa đổi theo ý kiến khách hàng trước khi tiếp tục phát triển các chức năng khác.

Milestone 2 (04/11/2024 – 01/12/2024)

Module chuyển giao:

- Quản lý và thao tác với nhiệm vụ: Phiên bản chức năng cho phép xem, thêm, sửa, xóa và các thao tác cơ bản với nhiệm vụ trong dự án.
- Cập nhật tiến độ công việc: Module cho phép người tham gia dự án cập nhật trạng thái và tiến độ công việc.
- Gửi email thông báo: Tích hợp hệ thống gửi email để thông báo khi nhiệm vụ sắp đến hạn, quá hạn, hoặc người dùng yêu cầu lấy lại mật khẩu.

Mục tiêu:

- Khách hàng có thể kiểm tra các chức năng quản lý nhiệm vụ và tiến độ, đồng thời nhận phản hồi về tính năng gửi email.
- Sửa đổi những yêu cầu từ khách hàng trước khi tiếp tục với Milestone 3.

Milestone 3 (02/12/2024 – 15/12/2024)

Module chuyển giao:

- Lưu trữ file dự án: Module cho phép người dùng tải lên và lưu trữ các file tài liệu cho từng dự án.
- Phiên bản hệ thống kiểm thử hoàn chỉnh: Hệ thống đã được kiểm tra toàn diện về chức năng và hiệu suất, sẵn sàng chuyển giao cho khách hàng.
- Tài liệu sử dụng và kỹ thuật: Hướng dẫn chi tiết cách sử dụng hệ thống và tài liệu dành cho đội ngũ kỹ thuật.

Mục tiêu:

- Đảm bảo khách hàng nhận được phiên bản hoàn thiện nhất của hệ thống và có đầy đủ tài liệu để sử dụng và quản lý sản phẩm.

V. TÍNH KHẢ THI VỀ MẶT KỸ THUẬT

1. Sắp xếp dữ liệu theo các trường khác nhau:

- Frontend: Sử dụng JavaScript để xử lý sắp xếp dữ liệu trên giao diện người dùng.
 - o Tiến độ dự án (hoàn thành bao nhiêu phần trăm).
 - o Ngày bắt đầu hoặc kết thúc dự án.
- Backend (Spring): Tạo các API RESTful trong Spring Boot để lấy dữ liệu từ cơ sở dữ liệu và sắp xếp theo yêu cầu từ phía frontend. Sử dụng JPA với các câu lệnh ORDER BY trong truy vấn để trả về dữ liệu đã được sắp xếp.
- PostgreSQL: Tối ưu hóa các truy vấn trong PostgreSQL để hỗ trợ việc sắp xếp, đặc biệt khi làm việc với lượng dữ liệu lớn. Có thể tạo các index phù hợp để tăng tốc độ truy xuất và sắp xếp dữ liệu.

2. Các cấp độ truy cập vào hệ thống

- Frontend: Sử dụng HTML, CSS, JS để hiển thị giao diện khác nhau cho từng loại người dùng (quản lý dự án, thành viên). JS sẽ kiểm tra vai trò của người dùng và điều chỉnh nội dung hiển thị tương ứng.
- Backend: Sử dụng Spring Security để quản lý quyền truy cập và phân quyền theo từng vai trò người dùng (quản lý dự án, thành viên, khách hàng). Điều này bao gồm việc giới hạn các API mà người dùng có thể truy cập dựa trên vai trò của họ.
- Database: Tạo các bảng hoặc trường dữ liệu trong PostgreSQL để lưu trữ thông tin phân quyền cho từng người dùng, ví dụ như bảng User_Roles để xác định vai trò.

3. Đa người dùng và truy cập đồng thời

- Frontend: Sử dụng các phương pháp AJAX hoặc Fetch API để cập nhật giao diện theo thời gian thực mà không cần làm mới trang. Điều này đảm bảo dữ liệu mới từ nhiều người dùng được cập nhật một cách nhanh chóng và mượt mà.
- Backend: Tối ưu hóa Spring Boot để hỗ trợ nhiều người dùng truy cập đồng thời. Có thể sử dụng Optimistic Locking trong JPA/Hibernate để tránh xung đột khi nhiều người cập nhật dữ liệu cùng lúc.
- Database: PostgreSQL hỗ trợ tốt cho việc quản lý nhiều người dùng đồng thời. Bạn có thể sử dụng các transaction để đảm bảo tính toàn vẹn dữ liệu khi nhiều người cùng cập nhật hệ thống.

4. Giao diện

- Frontend: Sử dụng HTML và CSS kết hợp với các framework như Bootstrap để xây dựng giao diện hiện đại, thân thiện với người dùng. JavaScript (có thể là React hoặc Vanilla JS) sẽ giúp tạo các tương tác động, như chuyển đổi giữa các dashboard, bảng dữ liệu, biểu đồ tiến độ.
- Backend: API RESTful từ Spring Boot sẽ cung cấp dữ liệu động cho giao diện người dùng. Sử dụng các công cụ như Thymeleaf nếu muốn tạo ra các trang web tĩnh với dữ liệu server-side.

5. Lưu trữ file cho từng dự án:

- Frontend: Thêm một form cho phép người dùng tải file lên khi họ truy cập vào trang chi tiết của từng dự án. Form này có thể sử dụng `<input type="file">` để cho phép người dùng chọn file từ máy tính của họ. Và tạo một nhánh để hiển thị những file đã được tải lên.
- Backend: Sử dụng Spring Boot để lưu trữ file theo dự án trên hệ thống tệp của server (local storage).
- Database: Xây dựng một bảng mới để lưu trữ thông tin về các file được tải lên cho từng dự án. Bảng này sẽ chứa thông tin về file (tên file, đường dẫn lưu trữ) và liên kết tới dự án mà file thuộc về.

6. Thông báo tới người dùng:

- Backend: Kết nối với API MailService để gửi thông báo tới người dùng qua mail, đồng thời hỗ trợ người dùng lấy lại mật khẩu bằng mail.

7. Bảo mật

- Frontend: Sử dụng HTTPS để đảm bảo rằng tất cả dữ liệu giữa trình duyệt và server được mã hóa. JavaScript cũng có thể xử lý token từ phía backend để đảm bảo tính bảo mật khi gọi các API.
- Backend: Sử dụng Spring Security kết hợp với JWT (JSON Web Token) để quản lý phiên đăng nhập của người dùng và bảo mật các endpoint API. Tích hợp thêm các phương pháp bảo mật như OAuth2 nếu cần thiết.
- Database: Bảo vệ dữ liệu trong PostgreSQL bằng cách giới hạn quyền truy cập vào cơ sở dữ liệu cho các người dùng khác nhau, và mã hóa các trường dữ liệu nhạy cảm như mật khẩu và thông tin cá nhân.

Kết luận

- Hệ thống sử dụng Postgres làm cơ sở dữ liệu.
- Sử dụng Spring Framework để xây dựng backend web.
- Sử dụng HTML, CSS, JS để xây dựng giao diện người dùng và tương tác với backend.
- Hệ thống trên đảm bảo được các yêu cầu kỹ thuật đề ra và hoàn toàn khả thi .

VI. PHÂN TÍCH RỦI RO

1. Rủi ro về thời gian:

- Do thời gian yêu cầu của môn học phải hoàn thành trong thời gian học phần nên những chức năng mới, mang tính đột phá hoặc ứng dụng cao chưa được đưa vào, dẫn đến hệ thống chức năng vẫn còn đơn sơ và có thể chưa làm hài lòng khách hàng hàng.
- Quá trình vừa học vừa làm trong thời gian ngắn kết hợp với việc thành viên của nhóm phải chia thời gian để thực hiện các môn học và các project khác có thể khiến thời gian đầu của dự án bị chậm tiến độ, chậm báo cáo.
- Có thể có những lỗi nặng phát sinh trong thời gian thực hiện dự án khiến cho dự án bị đình trệ trong thời gian dài do chưa khắc phục được vì người thực hiện dự án chưa tích lũy đủ kiến thức chuyên môn.
- Khi sử dụng các công nghệ mới mà các thành viên trong nhóm chưa nắm rõ, quá trình học hỏi và làm quen có thể mất nhiều thời gian. Điều này làm giảm năng suất công việc, đặc biệt khi cần triển khai các tính năng nâng cao hoặc sửa lỗi phức tạp.

Giải pháp khắc phục, các tùy chọn dự phòng:

- Tập trung vào các chức năng cơ bản và đảm bảo hoàn thiện những tính năng cần thiết đầu tiên. Sau đó, nếu có đủ thời gian, có thể phát triển các chức năng mới hoặc cải tiến.
- Áp dụng phương pháp Minimum Viable Product (MVP): phát triển hệ thống với các tính năng tối thiểu có thể vận hành, sau đó mới tiến hành tối ưu hóa.
- Thiết lập ưu tiên cho từng tính năng và chỉ làm những tính năng quan trọng nếu gặp áp lực về thời gian.
- Trong trường hợp chậm tiến độ, có thể xem xét việc tạm dừng các hoạt động ngoài dự án (nếu có thể) hoặc giảm bớt các môn học khác (nếu linh động) để tập trung hoàn toàn vào dự án.
- Tăng cường làm thêm ngoài giờ (nếu cần thiết) vào các ngày cuối tuần.
- Nếu gặp lỗi nặng mà nhóm không xử lý được, có thể tìm sự giúp đỡ từ các chuyên gia hoặc mentor bên ngoài để có giải pháp nhanh chóng.

2. Rủi ro về mặt nguồn lực:

- Về nhân lực, vài đa số các bạn trong nhóm đang thiếu kinh nghiệm trong việc phát triển dự án, chưa nắm rõ được hết công nghệ phát triển, còn đang trong quá trình học hỏi các công nghệ và framework nên khi làm sản phẩm thời gian đầu có thể rất khó khăn. Đồng thời, số lượng thành viên nhóm là 5 người cũng gây ra hạn chế về mặt tiến độ trong việc phát triển các chức năng nâng cao
- Khi các sự cố kỹ thuật phát sinh (ví dụ lỗi phần mềm, hỏng hóc phần cứng), các thành viên thiếu kinh nghiệm có thể gặp khó khăn trong việc tìm ra giải pháp khắc phục. Việc không xử lý kịp thời các vấn đề này có thể dẫn đến việc dự án bị đình trệ trong thời gian dài.

- Nếu nhóm không có đủ các phần mềm hoặc công cụ cần thiết cho việc phát triển và kiểm thử (như IDE, công cụ kiểm thử tự động, hay phần mềm quản lý dự án), tiến độ dự án có thể bị chậm lại do các thành viên phải tìm cách khắc phục hoặc tìm giải pháp thay thế.
- Nếu không có quy trình rõ ràng để xử lý các lỗi kỹ thuật và vấn đề phát sinh, nhóm có thể gặp khó khăn trong việc theo dõi, phát hiện, và giải quyết vấn đề kịp thời. Điều này có thể làm tăng thời gian khắc phục sự cố và ảnh hưởng đến chất lượng của sản phẩm cuối cùng.

Giải pháp khắc phục, các tùy chọn dự phòng:

- Chia các nhiệm vụ theo mức độ phức tạp. Những thành viên ít kinh nghiệm nên đảm nhận các phần đơn giản trước và tăng dần độ phức tạp khi đã quen với công nghệ và framework.
- Tổ chức các buổi học chung trong nhóm, nơi các thành viên có kinh nghiệm hơn hướng dẫn những người khác về công nghệ hoặc framework cần thiết. Tận dụng các khóa học online, ví dụ như Udemy, Coursera, hay các tài liệu chính thức của framework (Spring Boot, REST API) để học nhanh.
- Duy trì bộ tài liệu chi tiết về các quy trình và công nghệ trong dự án, để giảm thời gian học hỏi của các thành viên mới và tiết kiệm thời gian.

3. Rủi ro về mặt chức năng:

- Rủi ro về chức năng liên quan đến khả năng hoạt động của hệ thống, thiết kế không phản ánh yêu cầu sản phẩm làm cho khách hàng cảm thấy không ưa thích, hay là giao diện quá sơ sài hay rắc rối làm cho người dùng cảm thấy không thân thiện.
- Hệ thống có thể gặp tình trạng tải chậm, phản hồi chậm hoặc không ổn định khi xử lý số lượng lớn yêu cầu từ người dùng. Điều này dẫn đến trải nghiệm kém, đặc biệt nếu hệ thống phải phục vụ nhiều người dùng đồng thời.
- Các module hoặc chức năng riêng lẻ có thể hoạt động tốt, nhưng khi tích hợp với nhau lại phát sinh lỗi hoặc không hoạt động như mong đợi. Điều này có thể làm hệ thống bị crash hoặc gây ra các hành vi không mong muốn.
- Giao diện không được tối ưu hóa để hoạt động trên các thiết bị khác nhau (như máy tính, điện thoại, tablet) có thể gây ra vấn đề về hiển thị và trải nghiệm. Nếu hệ thống không tương thích tốt trên các nền tảng phổ biến, người dùng sẽ cảm thấy bất tiện khi sử dụng.

Giải pháp khắc phục ,các tùy chọn dự phòng:

- Khi có bản demo hoặc chức năng sơ khai, tiến hành testing với người dùng (user testing) để nhận phản hồi về trải nghiệm giao diện và chức năng
- Nếu gặp rủi ro về giao diện không thân thiện, có thể đơn giản hóa giao diện bằng cách tập trung vào tính năng cốt lõi, tránh thêm các yếu tố gây rối mắt cho người dùng.
- Cân nhắc việc sử dụng lại các mẫu giao diện UI đã được công nhận là tốt từ các thư viện thiết kế hoặc framework phổ biến như Bootstrap hoặc Material UI để tiết kiệm thời gian phát triển và đảm bảo tính thẩm mỹ.
- Nếu không thể triển khai một chức năng do giới hạn thời gian hoặc nguồn lực, có thể bỏ qua hoặc tạm hoãn chức năng đó, và thay vào đó tập trung hoàn thiện các chức năng quan trọng hơn trước.

- Thay vì tự xây dựng các chức năng phức tạp, có thể tìm kiếm các giải pháp hoặc dịch vụ API đã được phát triển sẵn để tích hợp vào hệ thống.
- Kiểm thử tích hợp sớm: Ngay khi một module mới được phát triển, cần thực hiện integration testing để kiểm tra việc module này hoạt động đúng khi kết hợp với các module khác. Việc phát hiện lỗi sớm giúp giảm thiểu sự cố hệ thống hoặc lỗi phức tạp khi tích hợp toàn bộ.

VIII. CONCLUSION/ KẾT LUẬN

Từ kết quả của nghiên cứu khả thi, nhóm nhận thấy rằng ứng dụng quản lý dự án nhóm khả thi về tính kỹ thuật, kỹ năng của các thành viên trong nhóm và thời gian. Với giới hạn thời gian là một học kỳ, nhóm tin rằng phạm vi của dự án có thể quản lý được và các yêu cầu của khách hàng có thể được đáp ứng một cách thỏa đáng khi hoàn thành hệ thống. Các thành viên trong nhóm cũng có đủ kỹ năng để triển khai hệ thống và quen thuộc với phần cứng và phần mềm có thể được sử dụng trong dự án này. Kết luận của báo cáo khả thi là để tiếp tục với dự án phát triển phần mềm này.