
TRƯỜNG ĐẠI HỌC PHENIKAA
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN
THIẾT KẾ WEB NÂNG CAO

Xây dựng ứng dụng web quản lý lỗi BugTrackr

Thành viên

Nguyễn Vĩnh An, 21011405, 21011405@st.phenikaa-uni.edu.vn
Dương Đức Bình, 21011878, 21011878@st.phenikaa-uni.edu.vn
Đoàn Hữu Bách, 21010998, 21010998@st.phenikaa-uni.edu.vn

Giảng viên hướng dẫn:
Nguyễn Lệ Thu

07/06/2024

MỤC LỤC

- 1. Giới thiệu.....2
 - 1.1 Đặt vấn đề..... 2
 - 1.2 Các giải pháp đã có..... 2
 - 1.3 Giải pháp đề xuất.....2
- 2. Thiết kế và triển khai..... 2
 - 2.1 Các yêu cầu chức năng.....2
 - 2.2 Các yêu cầu phi chức năng..... 3
 - 2.3 Sơ đồ (Use-cases).....3
 - 2.4 Đặc tả các Use-case.....3
 - 2.5 Các biểu đồ tuần tự..... 7
 - 2.6 Cơ sở dữ liệu..... 11
 - 2.7 Mô hình Model-View-Controller..... 12
 - 2.8 OAuth của GitHub..... 13
- 3. Một số thành phần khác của đồ án..... 14
 - 3.1 Tiến độ thực hiện dự án.....14
 - 3.2 Phân chia công việc.....15
 - 3.3 Mã nguồn.....15
 - 3.4 Triển khai ứng dụng web..... 15
- 4. Tài liệu tham khảo..... 15

1. Giới thiệu

1.1 Đặt vấn đề

Trong quá trình phát triển phần mềm, việc quản lý và giải quyết các lỗi là một vấn đề quan trọng và phức tạp. Các lỗi có thể xuất hiện từ nhiều nguồn khác nhau và ảnh hưởng đến chất lượng và hiệu suất của phần mềm. Để giải quyết vấn đề này, việc xây dựng một hệ thống quản lý lỗi hiệu quả là cần thiết.

1.2 Các giải pháp đã có

Hiện tại có nhiều ứng dụng và nền tảng quản lý lỗi đã được phát triển và sử dụng phổ biến trong quá trình phát triển phần mềm. Các ứng dụng này bao gồm:

- **GitHub:**
 - + **Ưu điểm:** Chức năng quản lý lỗi của GitHub có nhiều tính năng. Người tham gia dự án có thể phân loại lỗi dễ dàng bằng việc gắn thẻ cho lỗi.
 - + **Nhược điểm:** Mặc dù có nhiều tính năng hỗ trợ quản lý dự án, GitHub còn ít tính năng thuận lợi cho việc quản lý thông tin của lỗi.
- **Jira:**
 - + **Ưu điểm:** Chức năng quản lý lỗi của Jira có nhiều tính năng như mức độ ưu tiên, thời gian hay liên kết với các lỗi khác.
 - + **Nhược điểm:** Chức năng quản lý lỗi tốt tuy nhiên lỗi chỉ có thể thuộc một loại duy nhất.
- **GitLab:**
 - + **Ưu điểm:** Có nhiều chức năng hỗ trợ việc quản lý lỗi cho dự án phát triển phần mềm.
 - + **Nhược điểm:** Các trạng thái có thể có của lỗi còn ít.

1.3 Giải pháp đề xuất

Giải pháp nhóm chúng tôi đề xuất là ứng dụng BugTrackr. Ứng dụng này cho phép người dùng dễ dàng tạo, quản lý lỗi cho một dự án phát triển phần mềm. Nhờ việc sử dụng giao diện người dùng thân thiện, ứng dụng này giúp người dùng dễ dàng quản lý tiến độ phát triển phần mềm. Các ưu điểm của ứng dụng bao gồm:

- **Dễ dàng quản lý lỗi:** Ứng dụng cho phép quản lý lỗi theo nhiều cách như loại lỗi, người đảm nhận, trạng thái. Điều này giúp thành viên của dự án có thể dễ dàng lọc lỗi theo mong muốn.
- **Thân thiện với người dùng:** Giao diện của ứng dụng đơn giản, dễ hiểu và dễ sử dụng giúp người dùng quản lý lỗi của dự án một cách thuận lợi và không gặp phải trở ngại trong quá trình sử dụng.

2. Thiết kế và triển khai

2.1 Các yêu cầu chức năng

Các yêu cầu chức năng cơ bản của ứng dụng bao gồm:

- **Đăng ký/Đăng nhập:** Người dùng có thể đăng ký và đăng nhập tài khoản.
- **Tạo dự án:** Người dùng có thể tạo dự án mới và làm người quản lý dự án.

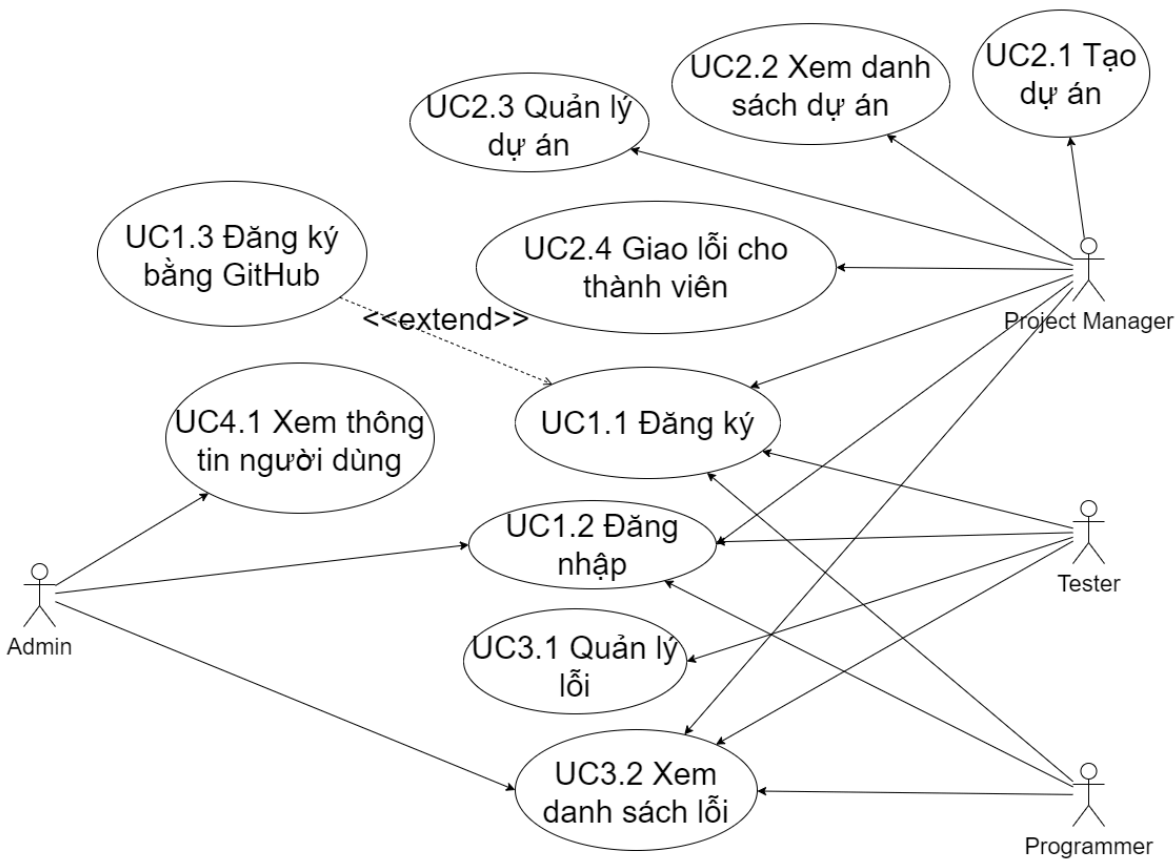
- **Quản lý dự án:** Người dùng có thể xem, sửa hoặc xóa các dự án của bản thân.
- **Quản lý thành viên:** Người quản lý dự án có thể xem, thêm hoặc xóa thành viên của dự án.
- **Giao lỗi cho thành viên:** Người quản lý dự án có thể chỉ định tài khoản thành viên phụ trách lỗi.
- **Tạo lỗi:** Thành viên của dự án có thể tạo một lỗi mới và các thông tin liên quan.
- **Quản lý lỗi:** Các thành viên của dự án có thể xem hoặc thay đổi các thông tin của lỗi.
- **Xem danh sách tài khoản:** Người quản lý phần mềm có thể xem các tài khoản đã đăng ký.
- **Xem danh sách dự án:** Người quản lý phần mềm có thể xem các dự án có trong phần mềm.
- **Xem danh sách lỗi:** Người quản lý phần mềm có thể xem danh sách các lỗi.

2.2 Các yêu cầu phi chức năng

Các yêu cầu phi chức năng cơ bản của ứng dụng bao gồm:

- Giao diện người dùng thân thiện và dễ sử dụng.
- Bảo mật thông tin người dùng và dữ liệu dự án.

2.3 Sơ đồ (Use-cases)



Hình 1.1. Sơ đồ Use-Case

2.4 Đặc tả các Use-case

❖ UC-1.1: Đăng ký

Mô tả: Người dùng tạo tài khoản.

Luồng sự kiện:

1. Người dùng bấm đăng ký.
2. Người dùng nhập các thông tin cần thiết.
3. Người dùng bấm nút đăng ký.
4. Hệ thống kiểm tra thông tin đăng ký.

Đăng ký thành công

6. Hệ thống lưu thông tin tài khoản.
7. Hệ thống chuyển về trang chủ.

Có lỗi

5. Hệ thống thông báo lỗi.

Tiền điều kiện: Người dùng chưa có tài khoản.

Kết quả: Người dùng đăng ký tài khoản thành công.

❖ **UC-1.2: Đăng nhập**

Mô tả: Người dùng đăng nhập vào tài khoản đã tạo.

Luồng sự kiện:

1. Người dùng bấm đăng nhập.
2. Người dùng nhập các thông tin cần thiết.
3. Người dùng bấm nút đăng nhập.
4. Hệ thống kiểm tra thông tin đăng nhập.

Đăng nhập thành công

6. Hệ thống lưu thông tin phiên đăng nhập.
7. Hệ thống chuyển về trang chủ.

Có lỗi

6. Hệ thống thông báo lỗi.

Tiền điều kiện: Người dùng chưa đăng nhập.

Kết quả: Người dùng đăng nhập thành công.

❖ **UC-1.3: Đăng ký bằng GitHub**

Mô tả: Người dùng có thể sử dụng tài khoản GitHub để đăng ký.

Luồng sự kiện:

1. Người dùng bấm đăng ký.
2. Người dùng bấm nút đăng ký bằng GitHub.
3. Hệ thống chuyển hướng đến GitHub để ủy quyền.
4. Hệ thống điều hướng về trang đăng ký với thông tin tài khoản GitHub.
5. Người dùng nhập thông tin cần thiết.
6. Hệ thống kiểm tra thông tin.

Đăng ký thành công

8. Hệ thống lưu thông tin tài khoản.
9. Hệ thống chuyển về trang chủ.

Có lỗi

8. Hệ thống thông báo lỗi.

Tiền điều kiện: Người dùng chưa đăng ký và có tài khoản GitHub.

Kết quả: Người dùng đăng ký thành công bằng thông tin tài khoản GitHub.

❖ **UC-2.1: Tạo dự án**

Mô tả: Người dùng tạo dự án mới.

Luồng sự kiện:

1. Người dùng mở trang danh sách dự án.
2. Người dùng bấm nút tạo dự án.
3. Hệ thống điều hướng đến trang tạo dự án.
4. Người dùng nhập thông tin cần thiết.
5. Hệ thống kiểm tra thông tin.

Đăng ký thành công

7. Hệ thống lưu thông tin dự án.
8. Hệ thống chuyển về trang danh sách dự án.

Có lỗi

7. Hệ thống thông báo lỗi.

Tiền điều kiện: Người dùng đã đăng nhập.

Kết quả: Người dùng tạo dự án mới thành công.

❖ **UC-2.2: Xem danh sách dự án**

Mô tả: Người dùng xem các dự án đang tham gia.

Luồng sự kiện:

1. Người dùng mở trang danh sách dự án.
2. Hệ thống hiển thị các dự án.

Tiền điều kiện: Người dùng đã đăng nhập.

Kết quả: Người dùng xem các dự án.

❖ **UC-2.3: Quản lý dự án**

Mô tả: Người dùng quản lý các thành phần của dự án.

Luồng sự kiện:

1. Người dùng mở trang danh sách dự án.
2. Người dùng chọn một dự án.
3. Hệ thống điều hướng sang trang dự án
4. Người dùng quản lý dự án theo nhu cầu.

Tiền điều kiện: Người dùng đã đăng nhập và đã tạo ít nhất một dự án.

Kết quả: Người dùng có thể quản lý dự án đã tạo.

❖ **UC-2.4: Giao lỗi cho thành viên**

Mô tả: Người quản lý dự án có thể giao lỗi cho thành viên.

Luồng sự kiện:

1. Người dùng mở trang danh sách dự án.
2. Người dùng chọn một dự án.
3. Hệ thống điều hướng sang trang dự án.
4. Người quản lý chọn lỗi.
5. Hệ thống hiển thị thông tin lỗi.
6. Người quản lý chọn thành viên phụ trách lỗi.

Tiền điều kiện: Người dùng đã đăng nhập, đã tạo ít nhất một dự án và ít nhất một lỗi.

Kết quả: Người quản lý dự án có thể giao lỗi cho thành viên của dự án.

❖ UC-3.1: Quản lý lỗi

Mô tả: Thành viên của dự án có thể tạo và sửa thông tin của lỗi.

Luồng sự kiện:

1. Người dùng mở trang danh sách lỗi.
2. Hệ thống điều hướng sang trang lỗi.
3. Người dùng bấm nút tạo lỗi.
4. Người dùng nhập thông tin cần thiết.
5. Hệ thống lưu lỗi mới.
6. Người dùng chọn lỗi.
7. Người dùng sửa thông tin cần thiết.

Tiền điều kiện: Người dùng đã đăng nhập.

Kết quả: Người dùng có thể tạo lỗi mới.

❖ UC-3.2: Xem danh sách lỗi

Mô tả: Người dùng có thể xem danh sách lỗi liên quan.

Luồng sự kiện:

1. Người dùng mở trang danh sách lỗi.
2. Hệ thống điều hướng sang trang lỗi.
3. Hệ thống hiển thị lỗi.

Tiền điều kiện: Người dùng đã đăng nhập.

Kết quả: Người dùng có thể xem danh sách lỗi.

❖ UC-4.1: Xem thông tin người dùng

Mô tả: Người quản trị có thể xem thông tin tài khoản.

Luồng sự kiện:

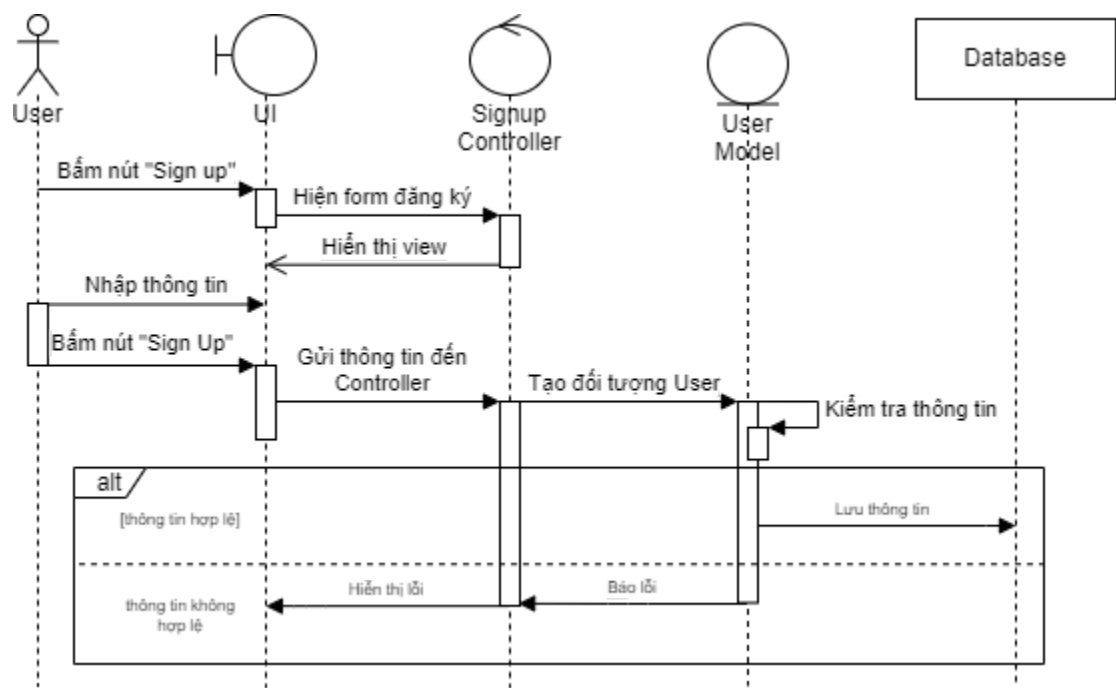
1. Người dùng mở ứng dụng.
2. Người dùng sang trang danh sách tài khoản.
3. Hệ thống hiển thị các tài khoản có trong hệ thống.

Tiền điều kiện: Người quản trị đã đăng nhập.

Kết quả: Người quản trị có thể xem thông tin tài khoản.

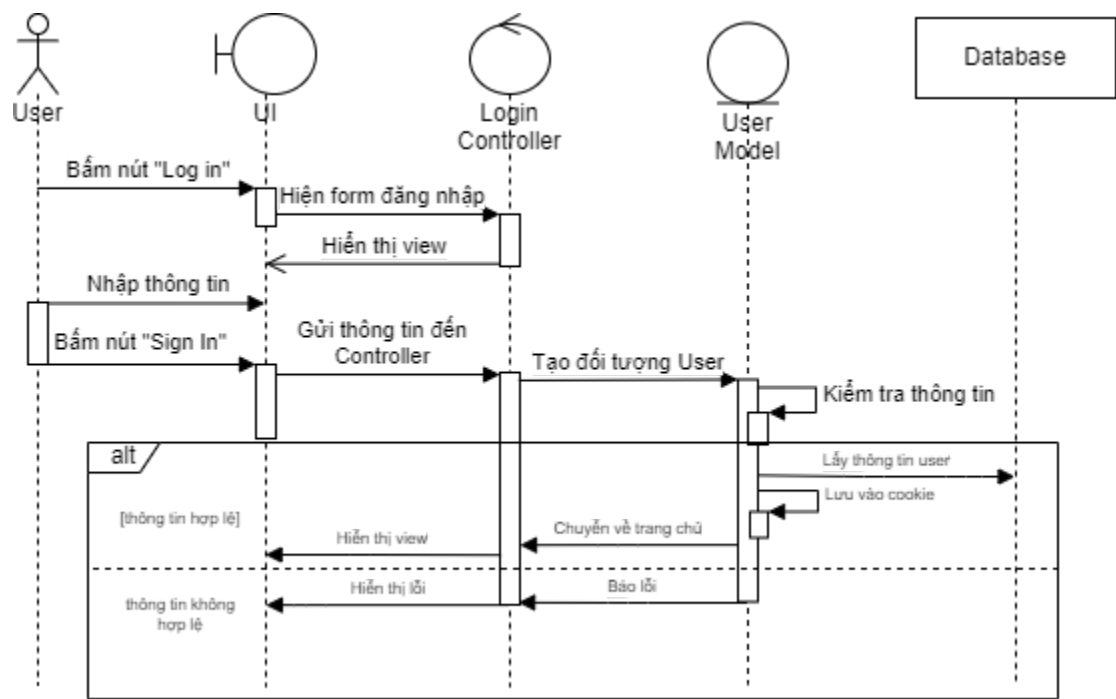
2.5 Các biểu đồ tuần tự

2.5.1 Chức năng đăng ký



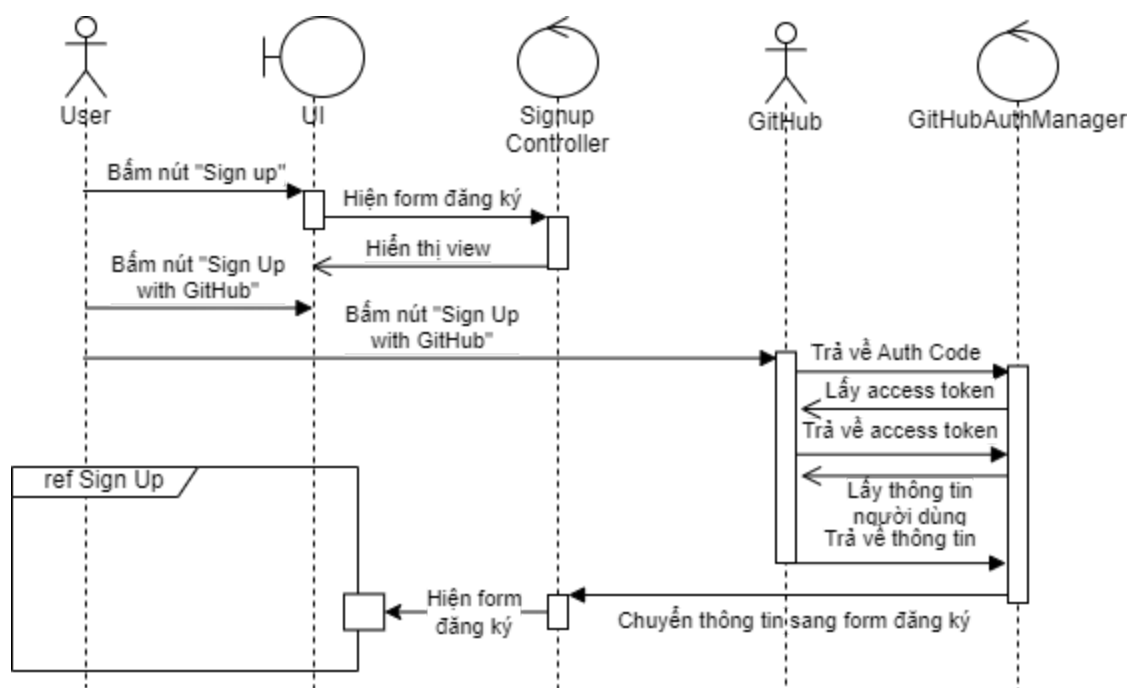
Hình 1.2. Biểu đồ tuần tự chức năng đăng ký

2.5.2 Chức năng đăng nhập



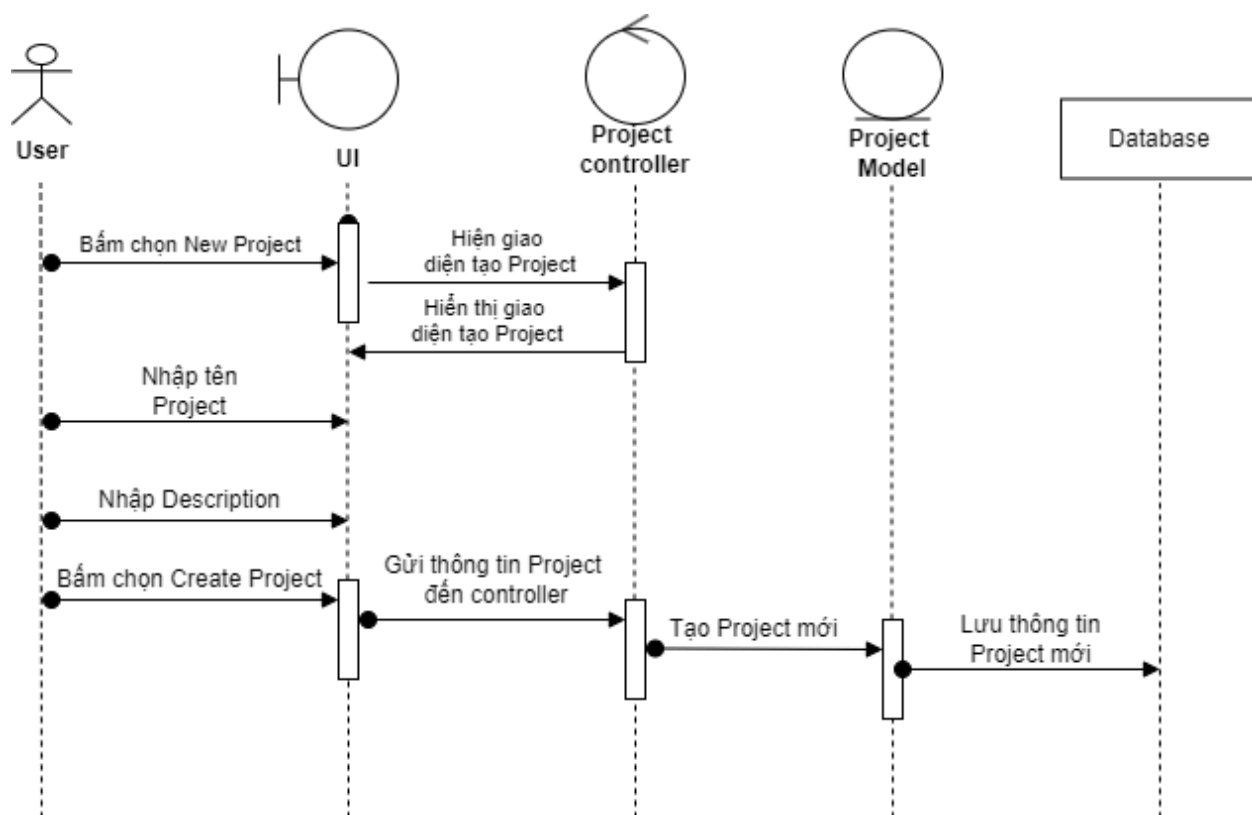
Hình 1.3. Biểu đồ tuần tự chức năng đăng nhập

2.5.3 Chức năng đăng ký bằng GitHub



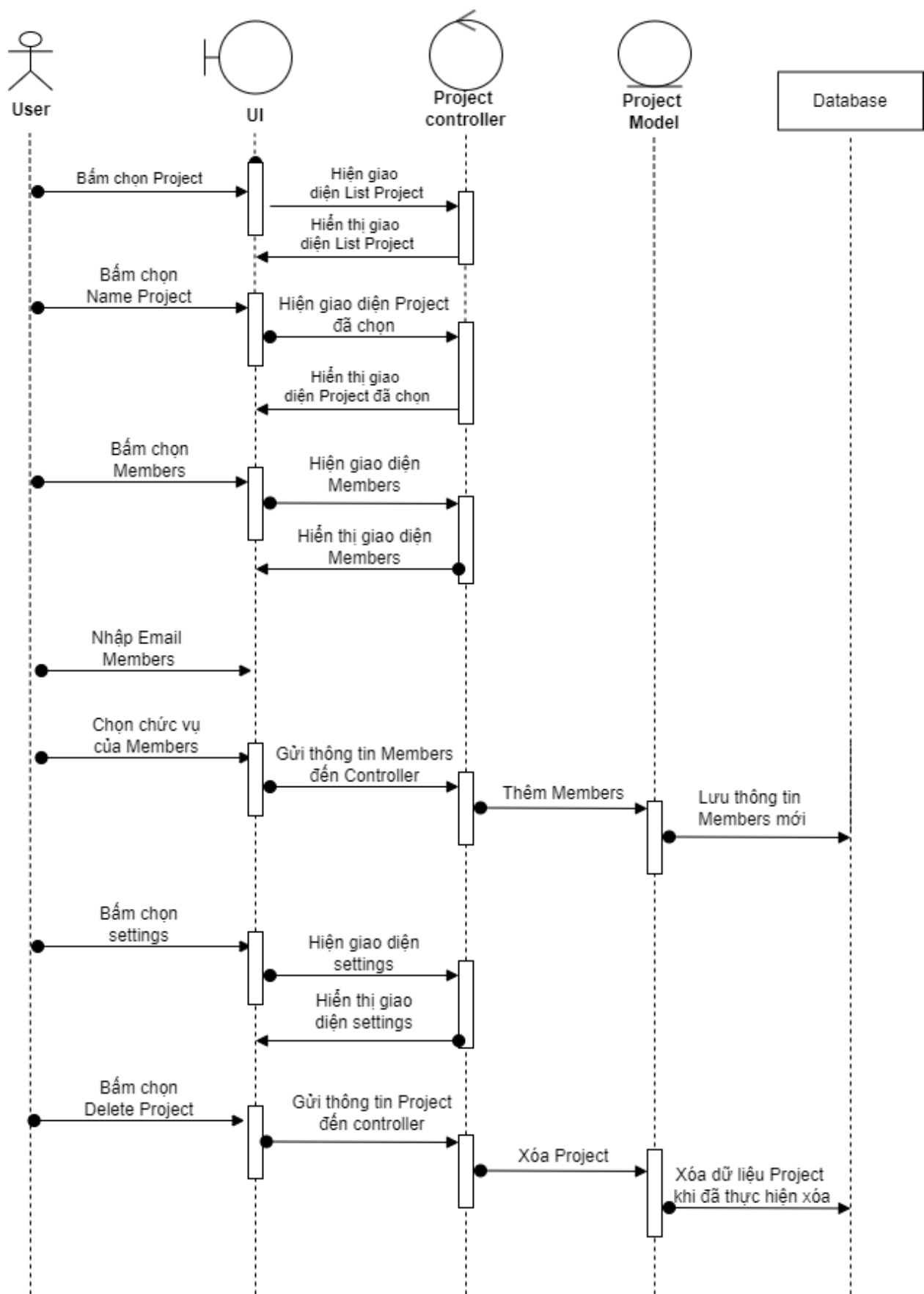
Hình 1.4. Biểu đồ tuần tự chức năng đăng ký bằng GitHub

2.5.4 Chức năng tạo dự án



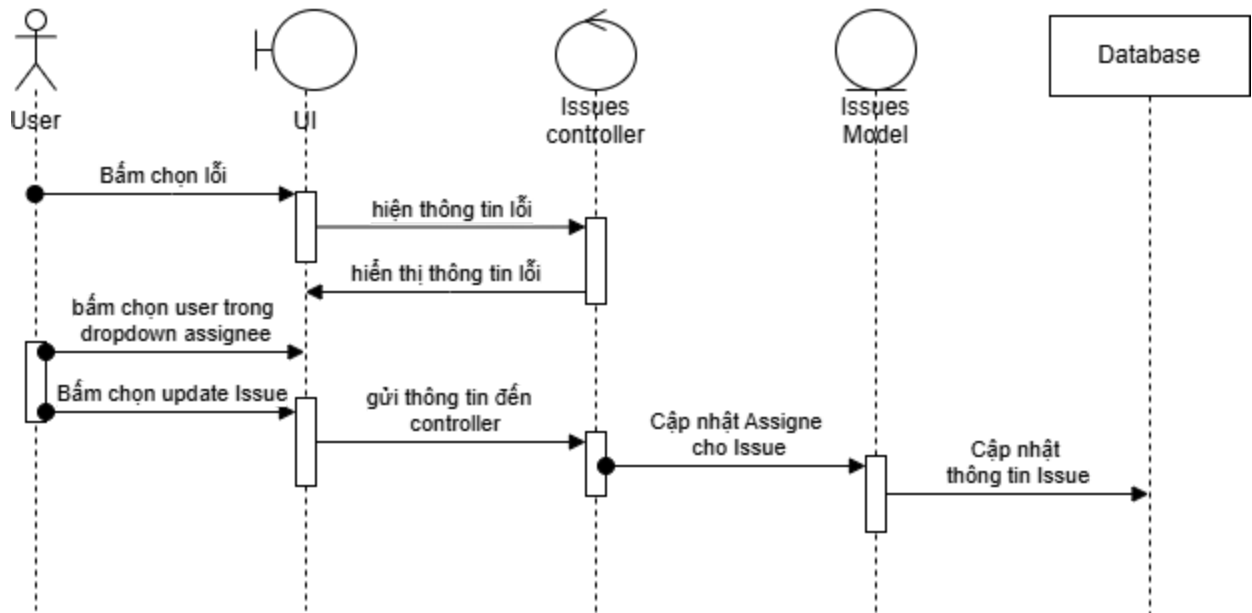
Hình 1.5. Biểu đồ tuần tự chức năng tạo dự án

2.5.5 Chức năng quản lý dự án



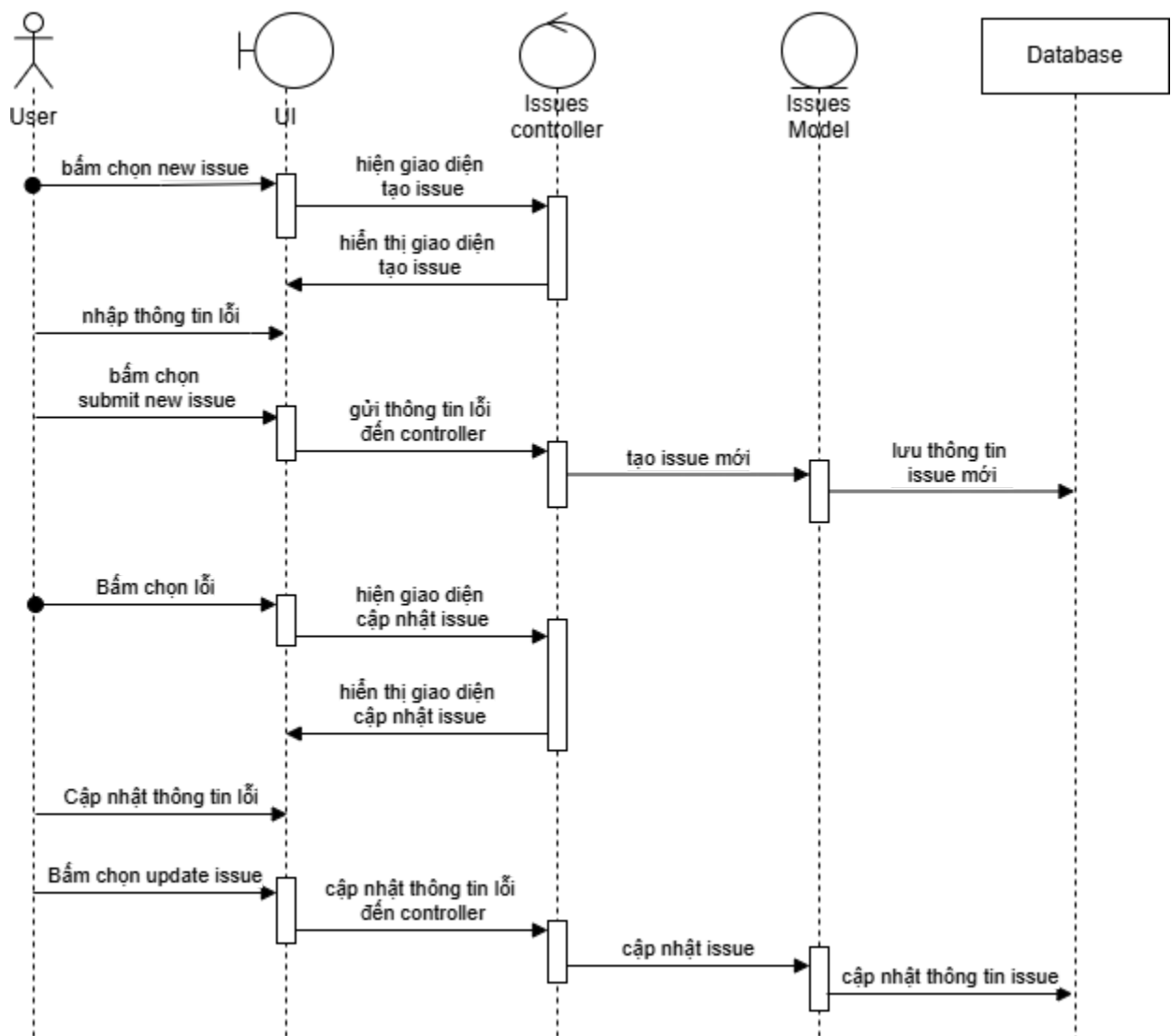
Hình 1.6. Biểu đồ tuần tự chức năng quản lý dự án

2.5.6 Chức năng giao lỗi cho thành viên



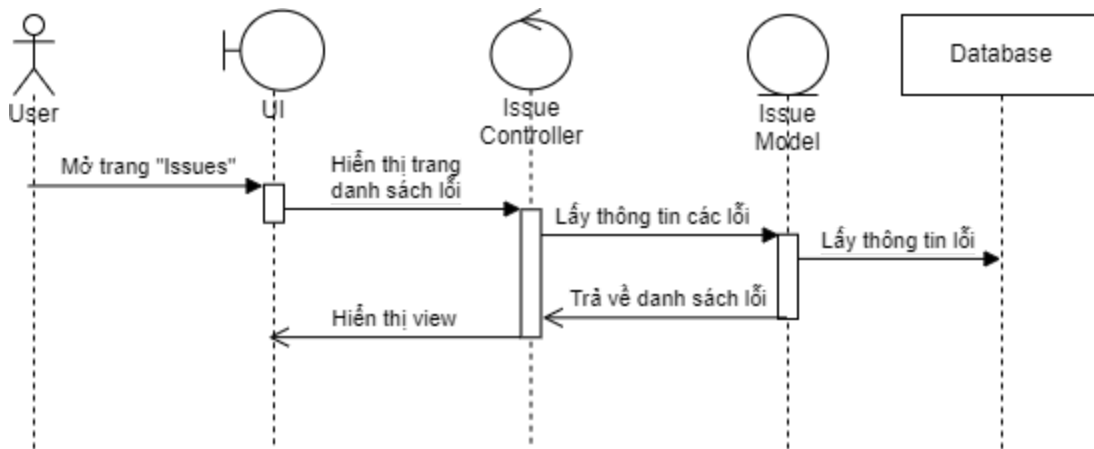
Hình 1.7. Biểu đồ tuần tự chức năng giao lỗi cho thành viên

2.5.7 Chức năng quản lý lỗi



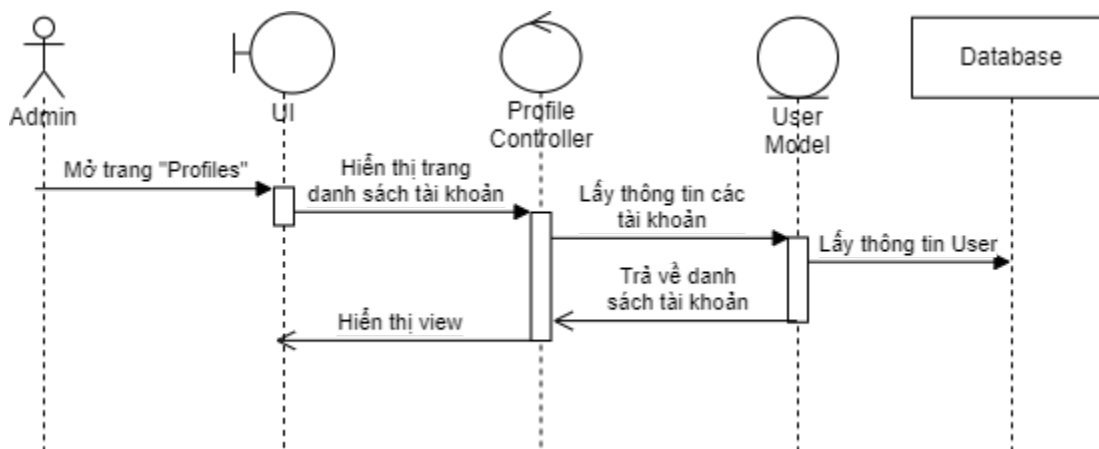
Hình 1.8. Biểu đồ tuần tự chức năng quản lý lỗi

2.5.8 Chức năng xem danh sách lỗi



Hình 1.9. Biểu đồ tuần tự chức năng xem danh sách lỗi

2.5.9 Chức năng xem thông tin người dùng



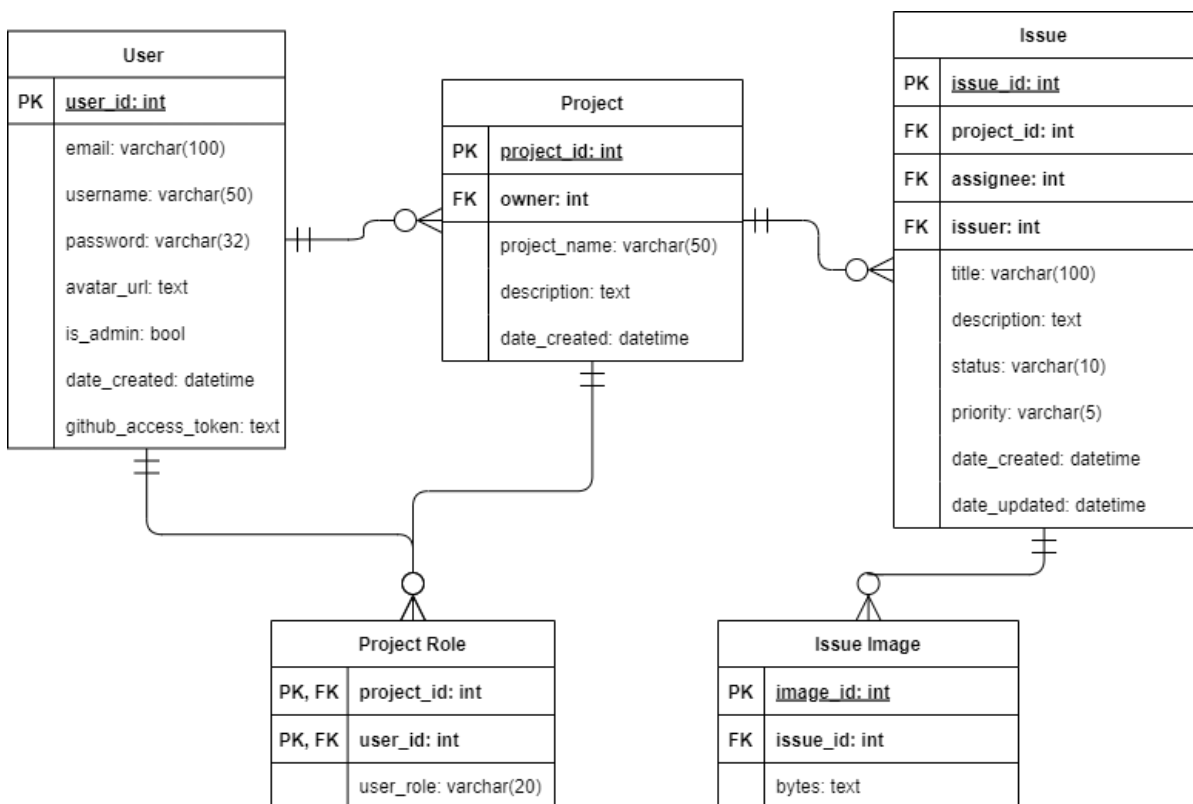
Hình 1.10. Biểu đồ tuần tự chức năng xem thông tin người dùng

2.6 Cơ sở dữ liệu

Cơ sở dữ liệu của ứng dụng được triển khai trên nền tảng điện toán đám mây thông qua Aiven (<https://aiven.io/>). Aiven cho phép người dùng tạo dịch vụ cơ sở dữ liệu và có thể kết nối với ứng dụng sử dụng cơ sở dữ liệu.



Hình 1.11. Kết nối với cơ sở dữ liệu trên Aiven bằng PHP

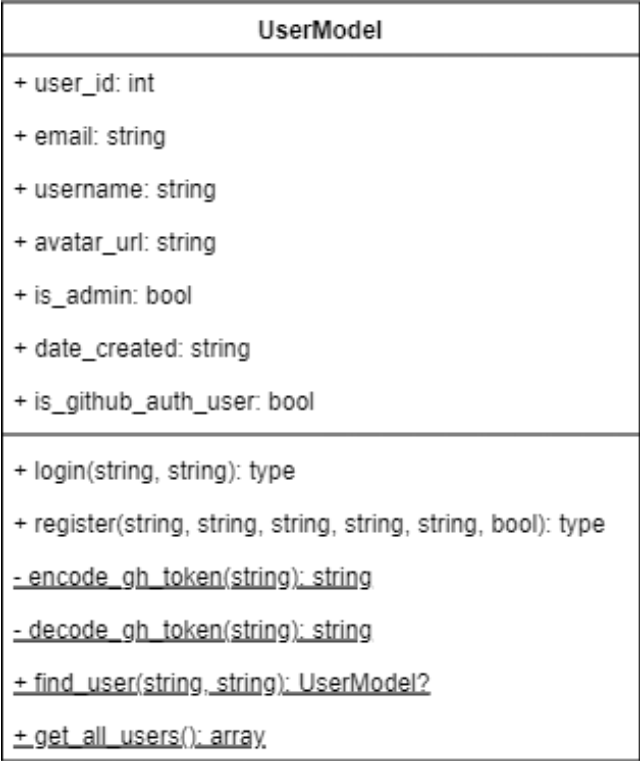


Hình 1.12. Mô hình cơ sở dữ liệu quan hệ

2.7 Mô hình Model-View-Controller

2.7.1 Model

Hệ thống bao gồm 3 model chính: UserModel, ProjectModel và IssueModel. UserModel là lớp chứa thông tin tài khoản của người dùng và thao tác với cơ sở dữ liệu để lấy thông tin người dùng. ProjectModel là lớp lưu thông tin của dự án và thao tác với cơ sở dữ liệu để lấy thông tin dự án, danh sách thành viên và lỗi. IssueModel là lớp lưu thông tin của lỗi như tiêu đề, nội dung, người phụ trách, trạng thái và lấy các thông tin đó từ cơ sở dữ liệu.



Hình 1.13. Biểu đồ lớp UserModel

ProjectModel
+ project_id: int + project_name: string + description: string + date_created: string + owner_id: int + owner: string + date_created: string + issue-count: int
+ <u>create_project(string, string): void</u> + <u>get_all_projects(int?): array</u> . + <u>get_owned_projects(int?): array</u> . + <u>get_joined_projects(int?): array</u> . + <u>get_project(int): ProjectModel</u> + <u>add_member(int, int): void</u> + <u>remove_member(int, int): void</u> + <u>get_members(int): array</u> . + <u>is_member_owner(int, int): bool</u> + <u>is_user_member(int, int): bool</u> + <u>delete_project(int): void</u>

Hình 1.14. Biểu đồ lớp ProjectModel

IssueModel
+ issue_id: int + project_id: int + title: string + description: string + status: IssueStatus + priority: IssuePriority + issuer: int? + issuer_name: string? + assignee: int? + assignee_name: string? + date_created: string + date_updated: string
+ <u>create_issue(int, string, string, IssueStatus, IssuePriorit</u> + <u>update_issue(int \$issue_id, ?string \$title = null, ?string :</u> + <u>get_issue(int, int): IssueModel</u> + <u>get_all_issues(bool?): array</u> . + <u>get_issues(int): array</u> . + <u>issue_type_count(): array</u> . + <u>filter_issues(IssueStatus): array</u> . + <u>get_created_issues(int, bool?, string?): array</u> . + <u>get_assigned_issues(int, bool?, string?): array</u> .

Hình 1.15. Biểu đồ lớp IssueModel

2.7.2 View

Hệ thống có 11 view thuộc về 5 loại

- Account: Chứa các view đăng ký, đăng nhập.
- Home: Chứa view trang chủ.
- Issue: Chứa các view xem, quản lý lỗi.
- Profile: Chứa view trang danh sách tài khoản.
- Project: Chứa các view xem, quản lý dự án.

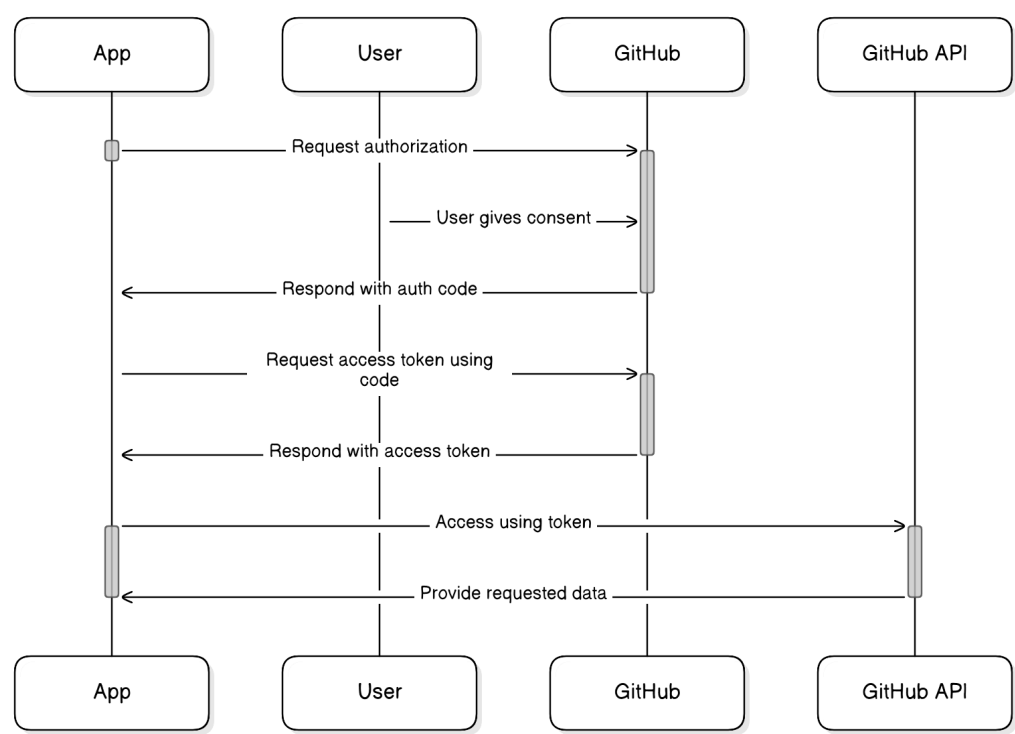
2.7.3 Controller

Hệ thống có 8 Controllers: GitHubAuth, Home, IssueController, Login, Logout, Profile, ProjectController, SignUp.

- GitHubAuth: Controller cho trang xử lý OAuth của GitHub.
- Home: Controller cho trang chủ.
- IssueController: Controller cho các trang và tác vụ liên quan đến lỗi.
- Login: Controller cho chức năng đăng nhập.
- Logout: Controller thực hiện việc đăng xuất.
- Profile: Controller cho trang hiển thị danh sách tài khoản.
- ProjectController: Controller cho các trang liên quan đến dự án.
- SignUp: Controller cho trang đăng ký.

2.8 OAuth của GitHub

Để có thể cho phép người dùng đăng ký thông qua GitHub cần phải sử dụng API OAuth của GitHub.



Hình 1.16. Các bước sử dụng OAuth

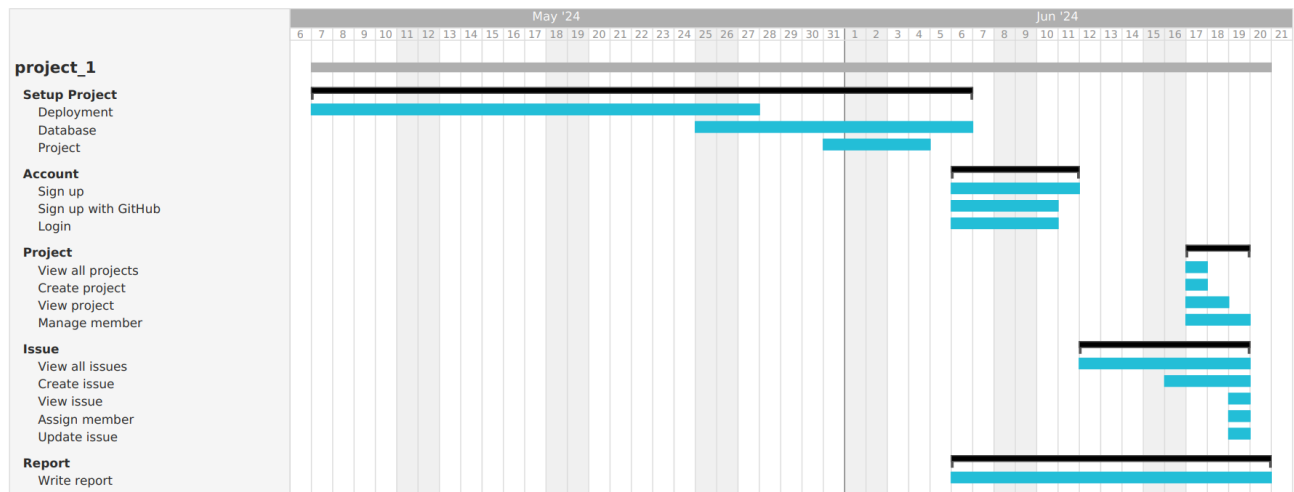
Để có thể yêu cầu Authorization từ GitHub trước tiên cần đăng ký ứng dụng OAuth bằng tài khoản GitHub của người phát triển. Thông tin của ứng dụng OAuth cần có tên ứng dụng, đường dẫn của ứng dụng và đường dẫn điều hướng khi GitHub trả về code.

Để gửi code cho GitHub và lấy access token, ứng dụng cần gửi yêu cầu HTTP POST đến https://github.com/login/oauth/access_token với Client ID và Client Secret của

ứng dụng đã đăng ký. Sau đó, để lấy thông tin của người dùng từ GitHub, ứng dụng cần gửi yêu cầu HTTP GET đến <https://api.github.com/user> cùng với access token.

3. Một số thành phần khác của đồ án

3.1 Tiến độ thực hiện dự án



Hình 2.1. Biểu đồ Gantt tiến độ công việc

3.2 Phân chia công việc

Thành viên	Công việc đã thực hiện
Nguyễn Vĩnh An	Thiết kế cơ sở dữ liệu
	Chức năng đăng ký
	Chức năng đăng ký bằng GitHub OAuth
	Chức năng đăng nhập
	Chức năng xem danh sách dự án
	Chức năng tạo dự án
	Chức năng xem thông tin người dùng
	Chức năng giao lỗi cho thành viên
	Deploy app trên nền tảng Cloud
Dương Đức Bình	Chức năng xem lỗi
Đoàn Hữu Bách	Chức năng quản lý dự án
	Chức năng quản lý lỗi

Bảng 1.1. Công việc đã thực hiện của các thành viên

3.3 Mã nguồn

Mã nguồn của ứng dụng được lưu trên GitHub (<https://github.com/Bill-GD/web-php>) thông qua Git để các thành viên trong nhóm có thể dễ dàng chia sẻ và đóng góp vào dự án.

3.4 Triển khai ứng dụng web

Ứng dụng sau khi đã được phát triển cũng đã được triển khai trên nền tảng đám mây thông qua dịch vụ Web Hosting được cung cấp bởi Koyeb (<https://www.koyeb.com>) tại đường dẫn <https://web-php-bill-gd-e914d913.koyeb.app>.

4. Tài liệu tham khảo

[1] “CodeIgniter 4.5.2 User Guide.” *CodeIgniter*,

<https://codeigniter4.github.io/userguide>.

[2] “Get started with Bootstrap · Bootstrap v5.3.” *Bootstrap*,

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>.

[3] “PHP Manual - Manual.” *PHP*, <https://www.php.net/manual/en/index.php>.