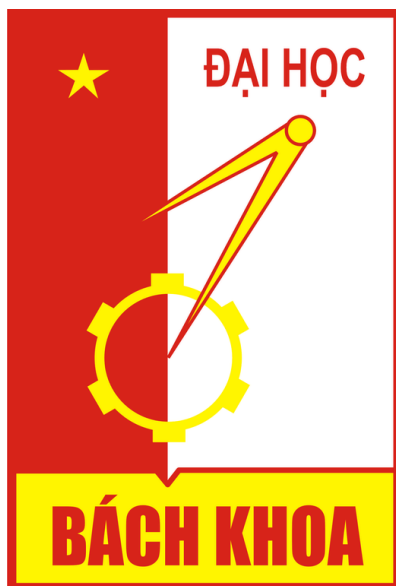


ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



Khảo sát ngôn ngữ

Project: Phần mềm học từ vựng

Môn học: Nghiên cứu tốt nghiệp 1 - IT5021

Giảng viên hướng dẫn : Đỗ Quốc Huy

Sinh viên thực hiện : Đinh Thị Hồng Phúc - 20215118

Lớp : Việt Nhật 03 - K66

Mục lục

1. Ngôn ngữ lập trình thường được sử dụng trong lập trình web	3
2. Cơ sở dữ liệu thường được sử dụng	9
3. Tổng quan project	11
3.1. Sơ đồ thực thể liên kết ERD	11
3.2. Kỹ thuật sử dụng	11
4. Thiết kế mockup	12

1. Ngôn ngữ lập trình thường được sử dụng trong lập trình web¹

Ngôn ngữ lập trình (Framework)	Mô tả	Ưu điểm	Nhược điểm
Python (Django)	<ul style="list-style-type: none"> - Sử dụng mô hình MVT (Model View Template) thay vì MVC (Model View Controller). Mô hình này kết hợp HTML và Django Template Language. Điều này cho phép Django tập trung vào quản lý và xử lý dữ liệu một cách hiệu quả, đồng thời phân chia rõ ràng giữa logic của ứng dụng và giao diện người dùng. - Các template của Django bao gồm HTML tích hợp sẵn, định tuyến URL, ORM, và quản lý phiên làm việc. Với bảng quản trị độc đáo, Django cung cấp một lợi thế so với các framework khác như Laravel, đặc biệt là trong việc quản lý. 	<ul style="list-style-type: none"> - Chu kỳ phát triển nhanh hơn do mã ngắn gọn và dễ đọc, nguyên tắc DRY: có thể tái sử dụng code hiện có. - Đảm bảo an ninh bằng cách ngăn chặn các lỗ hổng bảo mật thường gặp như: Cross-site request forgery, SQL injection. - Cài đặt dễ dàng thông qua dòng lệnh và hỗ trợ bởi tài liệu chi tiết, cập nhật thường xuyên. - Các thư viện như numpy, pandas... thúc đẩy sự phát triển trong khoa học dữ liệu và học máy. - Django cung cấp các tính năng như quản lý đăng nhập, quyền truy cập, bảo mật, và xử lý form, giúp tăng tốc độ và đơn giản hóa quá trình phát triển 	<ul style="list-style-type: none"> - Django không thích hợp với các dự án nhỏ do yêu cầu lượng code nhiều. - Đặc tính monolithic của nó có thể làm phức tạp việc mở rộng và quản lý dự án. - Django gặp hạn chế trong việc xử lý đồng thời nhiều request. - Tiêu thụ bộ nhớ cao, không áp dụng cho các tác vụ đòi hỏi nhiều bộ nhớ.

¹ [13 best languages for web development in 2024](#)

		web.	
PHP (Laravel)	<ul style="list-style-type: none"> - Mô hình MVC được áp dụng để tương tác với cơ sở dữ liệu thông qua Eloquent ORM, giúp thao tác dữ liệu bằng cách sử dụng cú pháp giống như ngôn ngữ lập trình, không cần viết truy vấn SQL. - Tích hợp sẵn các chức năng bảo mật như xác thực người dùng, phân quyền, và chống lại các cuộc tấn công CSRF. - Công cụ migration và seed được cung cấp để quản lý cấu trúc và dữ liệu mẫu của cơ sở dữ liệu, với khả năng chỉnh sửa và tạo dữ liệu mẫu một cách linh hoạt, hỗ trợ quá trình phát triển và kiểm thử. Kết nối với dịch vụ thư điện tử thông qua nền tảng đám mây hoặc hệ thống cục bộ sử dụng API đơn giản từ thư viện SwiftMailer. 	<ul style="list-style-type: none"> - Đảm bảo an ninh thông tin mức độ cao, và đã tích hợp sẵn các hệ thống lưu trữ đệm như Memcached và Redis. - Laravel có cộng đồng lớn và nhiều tài nguyên hỗ trợ. - Blade template engine của Laravel mạnh mẽ và dễ sử dụng, giúp tạo giao diện người dùng nhanh chóng. - Laravel hỗ trợ Artisan - công cụ dòng lệnh giúp quản lý các tác vụ lặp lại dễ dàng. 	<ul style="list-style-type: none"> - Việc quản lý gặp trở ngại vì sự đa dạng của các phiên bản và sự không tương thích giữa chúng. - Tốc độ tải trang cũng bị ảnh hưởng do mã nguồn Laravel có phần nặng. - Laravel có thể phức tạp đối với người mới bắt đầu do số lượng tính năng và công cụ đi kèm. - Cấu hình và quản lý mối quan hệ giữa các module trong ứng dụng lớn có thể gặp nhiều khó khăn.
Java (Spring)	<ul style="list-style-type: none"> - Spring Framework được thiết kế theo các nguyên lý Dependency Injection và Aspect Oriented Programming, bao gồm nhiều module hỗ trợ mục tiêu đa dạng như kiểm thử, quản lý core, data access, integration và web. - Tương thích với nhiều công nghệ như ORM, logging, JEE và các thư viện lịch trình như Quartz và JDK Timer. - Spring MVC, một phần của Spring Framework, chủ yếu được sử dụng để tạo ứng 	<ul style="list-style-type: none"> - Tích hợp trọn vẹn trong hệ sinh thái Spring: Spring Boot, Spring MVC,... - Cấu hình tự động toàn bộ các thành phần cần thiết cho một ứng dụng Spring sẵn sàng cho sản xuất. - Spring Boot cung cấp một hệ sinh thái hoàn chỉnh với khả năng cấu hình tự động và môi 	<ul style="list-style-type: none"> - Quản lý nhiều phiên bản và module đòi hỏi nhiều thời gian. - Việc cấu hình có thể trở nên phức tạp do việc định nghĩa bean và quản lý các mối quan hệ phụ thuộc. - Spring có thể tốn nhiều tài nguyên và đòi hỏi một lượng lớn bộ nhớ, đặc biệt trong các

	<p>dụng web theo mô hình MVC.</p> <ul style="list-style-type: none"> - Spring Boot, cũng là một phần của Spring Framework, giúp phát triển ứng dụng độc lập một cách nhanh chóng và hiệu quả với ít cấu hình hơn. Truy cập nhanh chóng vào các cơ sở dữ liệu và dịch vụ hàng đợi như MySQL, Oracle... 	<p>trường phát triển nhanh chóng, giúp các nhà phát triển tiết kiệm thời gian.</p> <ul style="list-style-type: none"> - Spring Security giúp bảo vệ ứng dụng web với các tính năng bảo mật tích hợp sẵn. 	<p>ứng dụng lớn.</p> <ul style="list-style-type: none"> - Việc học và làm quen với Spring có thể mất nhiều thời gian do độ phức tạp của nó.
C# (.NET)	<ul style="list-style-type: none"> - .NET là một framework của Microsoft hỗ trợ nhiều ngôn ngữ lập trình, trong đó C# là ngôn ngữ chính. - Sử dụng mô hình MVC, .NET giúp phát triển ứng dụng mạnh mẽ và có khả năng mở rộng. - Hỗ trợ tích hợp với nhiều dịch vụ và công cụ của Microsoft như Azure, SQL Server và Active Directory. - Được sử dụng rộng rãi trong các ứng dụng doanh nghiệp và web, đặc biệt trong môi trường Windows. 	<ul style="list-style-type: none"> - Hiệu suất cao, bảo mật tốt và hỗ trợ mạnh mẽ từ Microsoft. - Công cụ phát triển và IDE như Visual Studio cung cấp nhiều tiện ích và tính năng mạnh mẽ cho lập trình viên. - .NET Core (phiên bản mã nguồn mở của .NET) hỗ trợ đa nền tảng, cho phép phát triển ứng dụng trên Windows, macOS và Linux. - .NET có thư viện phong phú và hỗ trợ tốt cho phát triển ứng dụng web, desktop và mobile. 	<ul style="list-style-type: none"> - Chi phí cao cho giấy phép và dịch vụ của Microsoft. - Khả năng tùy biến thấp so với một số framework khác. - Đôi khi các bản cập nhật và thay đổi trong .NET có thể gây ra sự không tương thích và yêu cầu điều chỉnh mã nguồn.
Ruby (Ruby on Rails - RoR)	<ul style="list-style-type: none"> - Sử dụng mô hình MVC, RoR là một framework mạnh mẽ và dễ sử dụng để phát triển ứng dụng web. - RoR nổi tiếng với nguyên tắc “Convention over Configuration” (Ưu tiên quy ước hơn cấu hình) và “Don’t Repeat Yourself” (Không lặp 	<ul style="list-style-type: none"> - Tốc độ phát triển nhanh, cộng đồng hỗ trợ mạnh mẽ và nhiều thư viện phong phú. - Cung cấp các công cụ và tính năng giúp kiểm thử và triển khai dễ dàng. 	<ul style="list-style-type: none"> - Hiệu suất không cao bằng các framework khác như Django hay Spring, đặc biệt là với các ứng dụng lớn. - RoR có thể trở nên chậm và khó

	<p>lại chính mình), giúp tăng tốc độ phát triển và giảm thiểu lỗi.</p>	<ul style="list-style-type: none"> - RoR có các tính năng tích hợp sẵn cho quản lý cơ sở dữ liệu, kiểm thử và bảo mật, giúp giảm bớt khối lượng công việc cho các nhà phát triển. - Rails cũng có hệ sinh thái phong phú với nhiều gem (thư viện mở rộng) giúp giải quyết hầu hết các yêu cầu phát triển. 	<p>bảo trì khi ứng dụng phát triển lớn và phức tạp.</p> <ul style="list-style-type: none"> - Rails cần một lượng lớn bộ nhớ và tài nguyên CPU, điều này có thể là vấn đề đối với các ứng dụng có lượng truy cập lớn.
Swift (iOS Development)	<ul style="list-style-type: none"> - Swift là ngôn ngữ lập trình được Apple phát triển dành cho phát triển ứng dụng iOS, macOS, watchOS và tvOS. - Swift kết hợp cú pháp hiện đại và an toàn với hiệu suất nhanh và tính linh hoạt cao, giúp phát triển ứng dụng trên các nền tảng của Apple hiệu quả và dễ dàng. 	<ul style="list-style-type: none"> - Cú pháp hiện đại, dễ học và sử dụng. - Tích hợp chặt chẽ với các công cụ và nền tảng của Apple, cung cấp hiệu suất cao và khả năng tối ưu hóa tốt. - Swift có tính an toàn cao với việc quản lý bộ nhớ tự động và kiểm soát lỗi chặt chẽ. 	<ul style="list-style-type: none"> - Chỉ sử dụng được cho các nền tảng của Apple, hạn chế khả năng phát triển ứng dụng đa nền tảng. - Cộng đồng và tài liệu chưa phong phú như một số ngôn ngữ khác.
Kotlin (Android Development)	<ul style="list-style-type: none"> - Kotlin là ngôn ngữ lập trình được Google chọn làm ngôn ngữ chính cho phát triển ứng dụng Android. - Kotlin tương thích hoàn toàn với Java và cải tiến nhiều vấn đề của Java, có thể sử dụng cùng với mã Java trong cùng một dự án. 	<ul style="list-style-type: none"> - Cú pháp ngắn gọn, an toàn và dễ bảo trì. - Tương thích hoàn toàn với Java, có thể sử dụng cùng với các thư viện và công cụ hiện có của Java. - Hỗ trợ mạnh mẽ từ Google và cộng đồng phát triển Android. - Tính an toàn cao với kiểm soát null chặt chẽ, giảm 	<ul style="list-style-type: none"> - Dù tương thích với Java, nhưng việc chuyển đổi từ Java sang Kotlin có thể gặp một số khó khăn nhất định. - Cộng đồng và tài liệu chưa phong phú như Java.

		thiếu lỗi runtime.	
HTML/CSS	HTML và CSS kết hợp với nhau tạo nên giao diện và trình bày của các trang web trên Internet.	<ul style="list-style-type: none"> - Cơ bản và dễ học, là nền tảng của mọi trang web. - Được hỗ trợ rộng rãi bởi tất cả các trình duyệt web. - Cho phép tạo ra các trang web đẹp mắt và tương tác cao khi kết hợp với JavaScript. 	<ul style="list-style-type: none"> - Không có khả năng tạo logic phức tạp, cần kết hợp với các ngôn ngữ lập trình khác để tạo ra các ứng dụng web hoàn chỉnh. - CSS có thể phức tạp và khó quản lý trong các dự án lớn nếu không có cấu trúc tốt.
Javascript (ReactJS, VueJS, ExpressJS...)	<ul style="list-style-type: none"> - npm là hệ thống quản lý gói phần mềm cho JavaScript, đồng thời là kho phần mềm lớn nhất thế giới với hơn 350,000 packages. - ReactJS sử dụng model component-based, các component là những thành phần UI được tái sử dụng và mỗi component có trạng thái độc lập và không cần tương tác trực tiếp với DOM nhờ vào virtual DOM, cung cấp luồng dữ liệu đơn hướng, tùy chọn cú pháp JSX, cung cấp công cụ dòng lệnh create-react-app để khởi tạo dự án React Js mà không yêu cầu cấu hình. - VueJS là sự kết hợp giữa Angular và React, hỗ trợ animated transitions, cho phép người dùng di chuyển giữa các trang web mà không cần tải lại trang. - ExpressJS là một framework dành cho NodeJS, một môi trường chạy JavaScript phía máy chủ, hỗ trợ giao thức HTTP và middleware, giúp xây dựng API mạnh mẽ và dễ 	<ul style="list-style-type: none"> - Thực thi trên trình duyệt, giao tiếp với DOM, và xây dựng các ứng dụng web đa chức năng, tương tác cao. - Có khả năng đáp ứng tức thì thông qua việc định tuyến và sử dụng virtual DOM. - Cung cấp một môi trường phát triển đầy đủ tính năng. 	<ul style="list-style-type: none"> - Thiếu khả năng xử lý đồng thời và đa nhiệm. - Dễ bị tấn công bằng cách chèn mã độc. - Có những giới hạn với một số loại trình duyệt và cần được điều chỉnh cụ thể cho từng loại thiết bị và trình duyệt để đảm bảo tính nhất quán trong hiển thị. - Có thể khó quản lý khi dự án lớn và phức tạp.

	dòng định tuyến dựa trên phương thức HTTP và URL.		
Perl	<ul style="list-style-type: none"> - Perl là ngôn ngữ lập trình bậc cao, linh hoạt và mạnh mẽ, thường được sử dụng trong quản trị hệ thống, xử lý văn bản và phát triển web. - Perl hỗ trợ cả lập trình hướng đối tượng và lập trình thủ tục, với cú pháp linh hoạt và phong phú. 	<ul style="list-style-type: none"> - Rất mạnh trong xử lý văn bản và chuỗi, có thể tích hợp dễ dàng với nhiều hệ thống và cơ sở dữ liệu. - Cộng đồng lớn và nhiều tài liệu hỗ trợ. - Có khả năng xử lý biểu thức chính quy mạnh mẽ, giúp thao tác và xử lý văn bản hiệu quả. - Có thể sử dụng cho nhiều mục đích khác nhau từ scripting, automation, đến web development và data processing. 	<ul style="list-style-type: none"> - Cú pháp phức tạp và khó đọc. - Hiệu suất không cao bằng các ngôn ngữ hiện đại hơn như Python hoặc Ruby. - Có thể trở nên khó duy trì và mở rộng khi dự án lớn và phức tạp.
Go (GoLang)	<ul style="list-style-type: none"> - Go là ngôn ngữ lập trình mã nguồn mở do Google phát triển, được thiết kế hiệu quả, dễ đọc và dễ bảo trì. - Go thường được sử dụng trong các hệ thống phân tán, dịch vụ web và ứng dụng cần hiệu suất cao và đồng thời. - Go hỗ trợ lập trình hướng đối tượng, có cú pháp đơn giản và hệ thống quản lý gói mạnh mẽ. 	<ul style="list-style-type: none"> - Hiệu suất cao, hỗ trợ đồng thời tốt với goroutines. - Cú pháp đơn giản, dễ đọc và dễ bảo trì. - Quản lý bộ nhớ tự động và có nhiều công cụ hỗ trợ phát triển. - Go có khả năng biên dịch nhanh giúp triển khai ứng dụng dễ dàng. 	<ul style="list-style-type: none"> - Thiếu nhiều thư viện và framework so với các ngôn ngữ lập trình khác. - Không hỗ trợ tính năng generics, điều này có thể làm hạn chế trong một số trường hợp. - Có thể không phù hợp cho các ứng dụng cần GUI hoặc các ứng dụng đòi hỏi tính năng phức tạp và đa nhiệm cao

2. Cơ sở dữ liệu thường được sử dụng²

Cơ sở dữ liệu	Mô tả	Ưu điểm	Nhược điểm
Oracle	<ul style="list-style-type: none"> - Là hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ, được sử dụng rộng rãi trong các doanh nghiệp lớn. - Hỗ trợ nhiều tính năng nâng cao như phân vùng, nhân bản dữ liệu và bảo mật cao. 	<ul style="list-style-type: none"> - Hiệu suất cao và ổn định, đặc biệt với các hệ thống lớn. - Hỗ trợ nhiều tính năng bảo mật và quản lý người dùng. - Khả năng mở rộng tốt cho các ứng dụng lớn và phức tạp. 	<ul style="list-style-type: none"> - Chi phí bản quyền cao. - Quá trình cài đặt và cấu hình phức tạp. - Yêu cầu phần cứng mạnh và có thể khó khăn trong việc quản lý. - Đòi hỏi kiến thức chuyên sâu về cơ sở dữ liệu để sử dụng hiệu quả.
MySQL	<ul style="list-style-type: none"> - Là một hệ quản trị cơ sở dữ liệu mã nguồn mở, phổ biến trong các ứng dụng web và dịch vụ trực tuyến. - Hỗ trợ cả lưu trữ theo dạng bảng (MyISAM) và giao dịch (InnoDB). 	<ul style="list-style-type: none"> - Dễ dàng cài đặt và sử dụng. Miễn phí và mã nguồn mở. - Hiệu suất tốt cho các ứng dụng web. - Hỗ trợ nhiều công cụ quản lý và giao diện người dùng. - Cộng đồng lớn và nhiều tài liệu hỗ trợ. 	<ul style="list-style-type: none"> - Hạn chế trong việc hỗ trợ các tính năng cao cấp so với các RDBMS khác như Oracle. - Hiệu suất có thể giảm khi quy mô dữ liệu lớn. - Các tính năng bảo mật và quản lý người dùng không mạnh mẽ bằng Oracle.
Microsoft SQL Server	<ul style="list-style-type: none"> - Là hệ quản trị cơ sở dữ liệu quan hệ của Microsoft, được sử dụng rộng rãi trong các ứng dụng doanh nghiệp và dịch vụ trực tuyến. - Hỗ trợ tích hợp tốt với các sản phẩm khác của Microsoft. 	<ul style="list-style-type: none"> - Tích hợp tốt với hệ sinh thái Microsoft, đặc biệt là các sản phẩm như .NET và Azure. - Hỗ trợ các công cụ quản lý và phát triển mạnh mẽ. - Bảo mật cao và khả năng phục hồi dữ liệu tốt. - Hiệu suất tốt cho các ứng dụng 	<ul style="list-style-type: none"> - Chi phí bản quyền cao. - Chỉ chạy trên hệ điều hành Windows, giới hạn sự linh hoạt. - Yêu cầu phần cứng mạnh và tài nguyên hệ thống lớn. - Quá trình cài đặt và cấu hình có thể phức tạp.

² [Most Popular Databases in 2024](#)

		doanh nghiệp.	
PostgreSQL	Là một hệ quản trị cơ sở dữ liệu mã nguồn mở, mạnh mẽ và linh hoạt, hỗ trợ nhiều kiểu dữ liệu và tính năng cao cấp như JSON, XML và các hàm người dùng tùy chỉnh.	<ul style="list-style-type: none"> - Miễn phí và mã nguồn mở. - Hỗ trợ nhiều tính năng nâng cao và kiểu dữ liệu đa dạng. - Tính toàn vẹn dữ liệu và bảo mật cao. - Khả năng mở rộng tốt và hiệu suất cao. - Cộng đồng lớn và nhiều tài liệu hỗ trợ. 	<ul style="list-style-type: none"> - Quá trình cài đặt và cấu hình có thể phức tạp. - Đòi hỏi kiến thức chuyên sâu để tối ưu hóa và quản lý hiệu quả. - Hiệu suất có thể không cao bằng một số hệ quản trị cơ sở dữ liệu thương mại trong các ứng dụng yêu cầu cao về tốc độ và dung lượng.
MongoDB	<ul style="list-style-type: none"> - Là cơ sở dữ liệu NoSQL, sử dụng mô hình tài liệu để lưu trữ dữ liệu. - Phổ biến cho các ứng dụng yêu cầu tính linh hoạt cao trong việc lưu trữ và truy xuất dữ liệu không cấu trúc. 	<ul style="list-style-type: none"> - Dễ dàng mở rộng và phù hợp với các hệ thống phân tán. Miễn phí và mã nguồn mở. - Hỗ trợ lưu trữ dữ liệu không cấu trúc và linh hoạt trong việc thay đổi cấu trúc dữ liệu. - Hiệu suất cao cho các truy vấn không chuẩn hóa. 	<ul style="list-style-type: none"> - Thiếu tính toàn vẹn dữ liệu so với các RDBMS. - Không hỗ trợ tốt các giao dịch phức tạp. - Yêu cầu thiết kế cẩn thận để tránh vấn đề về hiệu suất và bảo mật. - Không phù hợp cho các ứng dụng yêu cầu tính nhất quán mạnh và quản lý giao dịch phức tạp.

3. Tổng quan project

3.1. Sơ đồ thực thể liên kết ERD

Mục đích của phần mềm nhằm tạo ra hệ thống học từ vựng với các chức năng (function) mà người dùng/vai trò người dùng có thể sử dụng tại thời điểm chạy.

Trước khi đăng nhập, người dùng có thể xem các public collection và standard collection nhưng không thể tương tác hoặc tạo collection riêng. Bất kỳ ai cũng có thể đăng ký, đăng nhập để trải nghiệm đầy đủ tính năng. Sau khi đăng nhập vào hệ thống, người dùng được cấp một không gian lưu trữ cho collection cá nhân (study set), có thể tạo và quản lý collection chứa flashcards theo nhu cầu bản thân. Người dùng có thể học, thêm mới, xem, chỉnh sửa, hoặc xóa flashcard. Có thể thiết lập quyền riêng tư cho collection của mình là public hoặc private. Người dùng cũng có thể xem, tìm kiếm public collection của người khác và sử dụng standard collection của hệ thống, và clone collection đó vào không gian lưu trữ cá nhân.

Quản trị hệ thống đóng vai trò quan trọng trong việc quản lý thông báo và người dùng, đảm bảo an ninh, quyền riêng tư, và bảo trì hệ thống. Họ chịu trách nhiệm cập nhật và duy trì cơ sở hạ tầng kỹ thuật số, cũng như quản lý dữ liệu để bảo đảm nội dung luôn chất lượng và hiện đại.

3.2. Kỹ thuật sử dụng

- Frontend: JavaScript (ReactJS)

Lý do:

- + ReactJS có cộng đồng lớn, nhiều tài liệu, ví dụ minh họa và thư viện hỗ trợ.
- + ReactJS sử dụng kiến trúc component, giúp tách biệt các phần giao diện thành các thành phần nhỏ. Điều này giúp quản lý mã nguồn dễ dàng hơn và tái sử dụng các component.

- Backend: Ruby (Ruby on Rails)

Lý do:

- + Ruby on Rails (RoR) được ưa chuộng tại Nhật Bản và hỗ trợ việc phát triển các ứng dụng trên nhiều nền tảng khác nhau, nhờ vào một cộng đồng hỗ trợ đông đảo.
- + RoR cung cấp các công cụ như scaffolding, giúp xây dựng nhanh chóng cấu trúc cơ bản của ứng dụng, từ đó rút ngắn đáng kể thời gian phát triển.

- Database: PostgreSQL

Lý do:

- + PostgreSQL hỗ trợ nhiều tính năng phức tạp như truy vấn phân tán, giao dịch, và quản lý dữ liệu lớn.
- + PostgreSQL có cộng đồng lớn và tài liệu phong phú (miễn phí, mã nguồn mở).

- Quản lý code: Github

4. Thiết kế mockup

[Figma](#)

[Mẫu thử tương tác độ tin cậy thấp](#)