**ĐẠI HỌC BÁCH KHOA HÀ NỘI**



**Võ Anh Tuấn**

**XÂY DỰNG ỨNG DỤNG CHIA SẺ KIẾN THỨC**

**BẰNG LARAVEL VÀ VUEJS**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Công nghệ thông tin**

**Cán bộ hướng dẫn: Nguyễn Hồng Phương**

**Sinh viên thực hiện: Võ Anh Tuấn**

**HÀ NỘI – 2019**

**LỜI CẢM ƠN**

Trước hết, tôi xin phép gửi lời cảm ơn chân thành nhất đến quý thầy giáo, cô giáo, đội ngũ giảng viên trường đại học Bách khoa Hà Nội đã hướng dẫn, giảng dạy và truyền đạt những bài học quý báu cho tôi trong công việc và cuộc sống. Đó không chỉ là kiến thức về chuyên ngành, mà hơn thế, đó còn là những bài học sâu sắc trong cuộc sống, là những kinh nghiệm quý giá, vô cùng cần thiết cho cả những năm tháng ngồi trên ghế giảng đường đại học và cả cho công việc trong tương lai. Là một sinh viên của trường đại học Bách khoa Hà Nội, bản thân tôi luôn tự hào vì điều đó, vì tôi là sinh viên của trường đại học Bách khoa Hà Nội, ngôi trường đại học có bề dày lịch sử, với đội ngũ giảng viên giỏi, đầy lòng nhiệt huyết. Cùng với đó là hệ thống cơ sở vật chất trường học đầy đủ, được đảm bảo phục vụ tốt nhất cho quá trình học tập và nghiên cứu của sinh viên.

Đồng thời, tôi xin gửi lời cảm ơn sâu sắc đến thầy giáo Nguyễn Hồng Phương đã tận tình giúp đỡ và hướng dẫn tôi trong suốt quá trình nghiên cứu và thực hiện khóa luận tốt nghiệp. Thầy cô đã nhiệt tình giảng giải, hỗ trợ và định hướng đưa ra những đóng góp quý báu cho tôi.

Để có được ngày hôm nay, tôi cũng xin gửi lời cảm ơn tới gia đình của tôi, lời cảm ơn bố mẹ, chị gái - những người đã lao động vất vả để nuôi tôi ăn học và là nguồn động lực để tôi phấn đấu trưởng thành hơn. Cuối cùng, tôi xin cảm ơn tới các bạn sinh viên của trường nói chung và tập thể lớp CNTT2.1 nói riêng. Cảm ơn các bạn đã đồng hành với tôi trong suốt 5 năm học qua. Các bạn là những người bạn chân thành tốt bụng năng động, và giúp đỡ tôi trong học tập cũng nhưng trong cuộc sống.

**TÓM TẮT**

**Tóm tắt:** Ngày nay, các kiến thức rất đa dạng phong phú trong học tập và đời sống, vì vậy việc chia sẻ các khái niệm, các kiến thức mới luôn luôn cần thiết. Dựa trên ý tưởng của trang web Viblo tôi đã phát triển một ứng dụng web dựa trên công nghệ Laravel và Vuejs. Tôi đã sử dụng Laravel và một chút Nodejs để phát triển ứng dụng phần máy chủ và sử dụng Vuejs để xây dựng giao diện người dùng.

Khóa luận gồm 6 chương có cấu trúc như sau:

Chương 1: Lời mở đầu - Trình bày tính cần thiết, động lực phát triển của đề tài.

Chương 2: Cơ sở lý thuyết và công nghệ - Trình cày cơ sở lý thuyết, giới thiệu các công nghệ sử dụng để xây dựng ứng dụng.

Chương 3: Phân tích yêu cầu bài toán - Trình bày các yêu cầu của hệ thống, mô tả và phân tích yêu cầu các chức năng của bài toán.

Chương 4: Giải quyết bài tóan với Ruby On Rails - Trình bày cách áp dụng công nghệ vào bài toán để triển khai và xây dựng ứng dụng.

Chương 5: Thực nghiệm - Trình bày thực nghiệm của ngừoi dùng thử và mô tả cách tiếp cận ứng dụng.

Chương 6: Kết luận – Trình bày kết quả mà khóa luận đạt được, nhưng điều bản thân đã học được khi làm khóa luận và đưa ra định hướng tiếp theo của ứng dụng.

**LỜI CAM ĐOAN**

Tôi xin cam đoan mọi kết quả trong khóa luận này đều do tôi thực hiện dưới sự hướng dẫn của thầy Nguyễn Hồng Phương. Tất cả các tham khảo nghiên cứu liên quan đều được nêu rõ nguồn gốc một cách rõ ràng từ danh mục tài liệu tham khảo trong khóa luận.

Khóa luận không sao chép tài liệu, công trình nghiên cứu từ người khác mà không chỉ rõ về mặt tài liệu tham khảo. Các thống kê, các kết quả trình bày trong khóa luận đều được lấy từ thực nghiệm khi chạy chương trình. Nếu tôi sai tôi hoàn toàn chịu trách nhiệm theo quy định của trường Đại học Bách khoa Hà Nội.

Hà Nội, ngày 02 tháng 05 năm 2019

Sinh viên

**MỤC LỤC**

[Chương 1. Lời mở đầu 1](#_Toc512896068)

[1.1. Lý do chọn đề tài 1](#_Toc512896069)

[1.2. Phạm vi và đối tượng nghiên cứu 2](#_Toc512896070)

[1.3. Mục tiêu khóa luận 2](#_Toc512896071)

[1.4. Khó khăn và giải pháp 2](#_Toc512896072)

[1.5. Nội dung khóa luận 3](#_Toc512896073)

[Chương 2. Cơ sở lý thuyết và công nghệ 4](#_Toc512896074)

[2.1. Tổng quan về lập trình web và phát triển ứng dụng web 4](#_Toc512896075)

[2.2. Angular 2 5](#_Toc512896076)

[2.2.1. Single-page Application (SPA) 5](#_Toc512896077)

[2.2.2. Giới thiệu về Angular 2 5](#_Toc512896078)

[2.2.3. Những thành phần chính của Angular 2 6](#_Toc512896079)

[2.3. Ruby và Ruby on Rails Framework 7](#_Toc512896080)

[2.3.1. Ruby là gì? 7](#_Toc512896081)

[2.3.2. Ruby on Rails framework 8](#_Toc512896082)

[2.4. MySQL 10](#_Toc512896083)

[2.4.1. Hệ quản trị cơ sở dữ liệu 10](#_Toc512896084)

[2.4.2. MySQL là gì 10](#_Toc512896085)

[Chương 3. Phân tích yêu cầu bài toán 12](#_Toc512896086)

[3.1. Mô tả bài toán 12](#_Toc512896087)

[3.2. Thu thập yêu cầu 12](#_Toc512896088)

[3.3. Phân tích tác nhân chức năng của hệ thống 12](#_Toc512896089)

[3.4. Phân tích yêu cầu 15](#_Toc512896090)

[3.4.1. Thiết kế ca sử dụng 15](#_Toc512896091)

[3.4.2. Phân tích biểu đồ ca sử dụng 17](#_Toc512896092)

[Chương 4. Giải quyết bài toán với Ruby On Rails, Angular 2 29](#_Toc512896093)

[4.1. Thiết kế phần mềm 29](#_Toc512896094)

[4.1.1. Thiết kế giao diện 29](#_Toc512896095)

[4.1.2. Thiết kế cơ sở dữ liệu 29](#_Toc512896096)

[4.2. Triển khai phần mềm: 32](#_Toc512896097)

[4.2.1. Ứng dụng Angular 2 để xây dựng phía máy khách 32](#_Toc512896098)

[4.2.2. Xây dựng backend với ruby on rails 34](#_Toc512896099)

[4.3. Kiểm thử tự động 35](#_Toc512896100)

[Chương 5. Thực nghiệm 37](#_Toc512896101)

[5.1. Môi trường thực nghiệm 37](#_Toc512896102)

[5.2. Dữ liệu thực nghiệm 37](#_Toc512896103)

[5.3. Kết quả thực nghiệm 38](#_Toc512896104)

[5.3.1. Xác thực và ủy quyền ngừoi dùng 38](#_Toc512896105)

[5.3.2. Dành cho người dùng 40](#_Toc512896106)

[5.3.3. Khu vực quản lý của quản trị viên 47](#_Toc512896107)

[5.4. Nhận xét, đánh giá ứng dụng 49](#_Toc512896108)

[Chương 6. Kết luận 51](#_Toc512896109)

[6.1. Kết quả đạt được 51](#_Toc512896110)

[6.2. Định hướng phát triển 51](#_Toc512896111)

[TÀI LIỆU THAM KHẢO 52](#_Toc512896112)

**DANH MỤC HÌNH ẢNH**

[Hình 2.1: Mô hình MVC 9](#_Toc512417602)

[Hình 3.1: Mô hình ca sử dụng của người dùng 15](#_Toc512417603)

[Hình 3.2: Mô hình ca sử dụng của quản trị viên 15](#_Toc512417604)

[Hình 3.3 Ràng buộc giữa các use-case 16](#_Toc512417605)

[Hình 4.1: Mô hình dữ liệu trong cơ sở dữ liệu MySQL 29](#_Toc512417606)

[Hình 4.2: Kết quả chạy kiểm thử tự động 36](#_Toc512417607)

[Hình 5.1: Dữ liệu dưới dạng excel của google sheets 37](#_Toc512417608)

[Hình 5.2: Giao diện khi chưa đăng nhập 38](#_Toc512417609)

[Hình 5.3: Giao diện đăng ký tài khoản của ngừoi dùng 39](#_Toc512417610)

[Hình 5.4: Giao diện trang chủ dành cho quản trị viên 39](#_Toc512417611)

[Hình 5.5: Giao diện trang chủ dành cho người dùng 40](#_Toc512417612)

[Hình 5.6: Giao diện trang cá nhân của người dùng 40](#_Toc512417613)

[Hình 5.7: Giao diện trang cá nhân của người dùng 41](#_Toc512417614)

[Hình 5.8: Giao diện xem và chỉnh sửa thông tin người dùng 41](#_Toc512417615)

[Hình 5.9: Giao diện trang chủ của người dùng 42](#_Toc512417616)

[Hình 5.10: Giao diện của một bộ thẻ với các bài học 43](#_Toc512417617)

[Hình 5.11: Giao diện của một bộ thẻ với các thẻ 43](#_Toc512417618)

[Hình 5.12: Giao diện bài học với Dạng câu hỏi học từ 44](#_Toc512417619)

[Hình 5.13: Giao diện bài học với Dạng câu hỏi viết 44](#_Toc512417620)

[Hình 5.14: Giao diện bài học với Dạng câu hỏi nhớ các mặt 45](#_Toc512417621)

[Hình 5.15: Tạo bộ thẻ 45](#_Toc512417622)

[Hình 5.16: Nhập dữ liệu màn tạo bộ thẻ 46](#_Toc512417623)

[Hình 5.17: Giao diện sau khi đăng nhập của quản trị viên 47](#_Toc512417624)

[Hình 5.18: Giao diện tất cả người dùng 47](#_Toc512417625)

[Hình 5.19: Giao diện tất cả bộ thẻ 48](#_Toc512417626)

[Hình 5.20: Giao diện tất cả phản hồi 48](#_Toc512417627)

**DANH MỤC BẢNG BIỂU**

[Bảng 3.1: Mô tả ca sử dụng đăng nhập 17](#_Toc512216527)

[Bảng 3.2: Mô tả ca sử dụng đăng ký 18](#_Toc512216528)

[Bảng 3.3: Mô tả ca sử dụng đăng xuất 19](#_Toc512216529)

[Bảng 3.4: Mô tả ca sử dụng tạo bộ thẻ 20](#_Toc512216530)

[Bảng 3.5: Mô tả ca sử dụng học theo bài học 21](#_Toc512216531)

[Bảng 3.6: Mô tả ca sử dụng chỉnh sửa bộ thẻ 22](#_Toc512216532)

[Bảng 3.7: Mô tả ca sử dụng tìm kiếm bộ thẻ 23](#_Toc512216533)

[Bảng 3.8: Mô tả ca sử dụng chỉnh sửa trang cá nhân 24](#_Toc512216534)

[Bảng 3.9: Mô tả ca sử dụng xem tất cả bộ thẻ 26](#_Toc512216535)

[Bảng 3.10: Mô tả ca sử dụng xem tất cả người dùng 27](#_Toc512216536)

[Bảng 3.11: Mô tả ca sử dụng phản hồi của người dùng 25](#_Toc512216537)

[Bảng 3.12: Mô tả ca sử dụng xem tất cả phản hồi của bộ thẻ 28](#_Toc512216538)

[Bảng 4.1: Mô tả bảng User 30](#_Toc512216539)

[Bảng 4.2: Mô tả bảng Decks 30](#_Toc512216540)

[Bảng 4.3: Mô tả bảng Cards 30](#_Toc512216541)

[Bảng 4.4: Mô tả bảng Lessons 31](#_Toc512216542)

[Bảng 4.5: Mô tả bảng User\_Lessons 31](#_Toc512216543)

[Bảng 4.6: Mô tả bảng Feedbacks 31](#_Toc512216544)

[Bảng 4.7: Mô tả bảng Favorites 31](#_Toc512216545)

[Bảng 4.8: Mô tả bảng Relationships 32](#_Toc512216546)

[Bảng 4.9: Mô tả bảng Admins 32](#_Toc512216547)

[Bảng 4.10: Bảng định nghĩa các định tuyến của ứng dụng 32](#_Toc512216548)

[Bảng 4.11: Bảng mô tả API 34](#_Toc512216549)

[Bảng 5.1: Phản hồi của người dùng 49](#_Toc512216550)

**DANH SÁCH KÝ HIỆU, CHỮ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| ***Ký hiệu*** | ***Diễn giải*** | ***Tiếng Việt*** |
| *HTML* | *HyperText Markup Language* | *Ngôn ngữ đánh dấu siêu văn bản* |
| *CSS* | *Cascading Style Sheets* | *Ngôn ngữ định dạng các ngôn ngữ tạo ra bởi ngôn ngữ đánh dấu* |
| *API* | *Application Programming Interface* | *Giao diện lập trình ứng dụng* |
| *ASP* | *Active Server Pages* | *Trang máy chủ hoạt động* |
| *CGI* | *Computer-generated imagery* | *Máy tính tạo ra hình ảnh* |
| *JSP* | *JavaServer Pages* | *Trang máy chủ java* |
| *JS* | *Javascript* |  |
| *AI* | *Artificial intelligence* | *Trí tuệ nhân tạo* |
| *UI* | *User Interface* | *Giao diện người dùng* |
| *OOP* | *Object-oriented programming* | *Lập trình hướng đối tượng* |
| *REST* | *Representational state transfer* | *Chuyển trạng thái biểu hiện* |
| *ORM* | *Object-relational mapping* | *Bản đồ quan hệ giữa các đối tượng* |
| *SQL* | *Structured Query Language* | *Ngôn ngữ truy vấn cấu trúc* |
| *TCP* | *Transmission Control Protocol* | *Giao thức điều khiển truyền* |
| *BDD* | *Behavior-driven Development* | *Phát triển hành vi* |
| *MMORPG* | *Massively Multiplayer Online Role-Playing Game* | *Trò chơi nhập vai trực tuyến nhiều người chơi* |

# Lời mở đầu

* 1. Lý do chọn đề tài

**Trong những năm gần đây, con số các công cụ được thiết kế để hỗ trợ cho các kĩ sư phát triển phần mềm thực sự bùng nổ. Những frameworks giúp bạn kết nối các ứng dụng lại nhanh hơn, các thư viện mã nguồn mở giúp bạn tránh phải dựng và tái dựng những phần nhỏ nhặt thông thường. Đồng thời, các dịch vụ đám mây giúp cho việc thêm các đặc tính dễ dàng và tiết kiệm hơn rất nhiều so với khi bạn tự dựng.**

Nhưng nếu bạn muốn học cách làm thế nào để sử dụng tất cả các công nghệ đó thì nó đồng nghĩa với việc bạn phải đọc sách rất nhiều. Đọc hàng tá các tập tài liệu có thể là một việc quá khó khăn và dễ gây chán nản, đặc biệt là nếu tài liệu không cho bạn biết những gì bạn cần biết. Nhiều devs cuối cùng đành phải tìm đến các trang web để được giúp đỡ. Một trong số đó là trang web Viblo, một trang web có tiếng nơi mà các kĩ sư có thể chia sẻ kinh nghiệm, kiến thức về một lĩnh vực công nghệ nào đó hay đặt câu hỏi và tìm lời giải đáp cho một vấn đề thắc mắc.

Dựa trên mục đích đó tôi đã xây dựng một ứng dụng chia sẻ kiến thức với giao diện dễ dàng sử dụng và giúp cho mọi người có thể tiếp cận, trao đổi nhanh nhất với một lượng dữ liệu lớn.

Để xây dựng ứng dụng tốt nhất,tôi đã sử dụng công nghệ Laravel kết hợp với Vuejs. Tôi sẽ trình bày chi tiết về các công nghệ ở các chương sau của khóa luận.

* 1. Phạm vi và đối tượng nghiên cứu
* Phạm vi địa lý của khóa luận: Trên lãnh thổ Việt Nam.
* Đối tượng nghiên cứu: Những người có nhu cầu tìm hiểu hay chia sẻ kiến thức về lập trình.
* Các công nghệ sử dụng trong khóa luận:
* Laravel
* Vuejs
  1. Mục tiêu khóa luận

Bài toán đặt ra yêu cầu một giải pháp để giải quyết yêu cầu bài toán, do đó việc xây dựng và phát triển hoàn thiện ứng dụng là một vấn đề không nhỏ cần nhiều thời gian và công sức. Nội dung và kết quả của khóa luận là tập trung phân tích và thiết kế các chức năng cơ bản, cần thiết nhất đáp ứng yêu cầu cơ bản được đặt ra của người dùng và giải quyết các vấn đề mà các ứng dụng tương tự còn gặp phải. Đầu ra của ứng dụng là một ứng dụng web đáp ứng yêu cầu của bài toán.

* 1. Khó khăn và giải pháp

Hiện tại, có nhiều ứng dụng web để chia sẻ kiến thức. Tôi cần phải xây dựng một hệ thống vừa giúp người dùng dễ dàng sử dụng vừa đáp ứng được đầy đủ chức năng. Do đó cần xây dựng trang web với một giao diện thân thiện, tăng trải nghiệm cho người dùng.

* 1. Nội dung khóa luận

Khóa luận gồm 6 chương có cấu trúc như sau:

Chương 1: Lời mở đầu - Trình bày tính cần thiết, động lực phát triển của đề tài.

Chương 2: Cơ sở lý thuyết và công nghệ - Trình cày cơ sở lý thuyết, giới thiệu các công nghệ sử dụng để xây dựng ứng dụng.

Chương 3: Phân tích yêu cầu bài toán - Trình bày các yêu cầu của hệ thống, mô tả và phân tích yêu cầu các chức năng của bài toán.

Chương 4: Giải quyết bài tóan với Vuejs - Trình bày cách áp dụng công nghệ vào bài toán để triển khai và xây dựng ứng dụng.

Chương 5: Thực nghiệm - Trình bày thực nghiệm của ngừoi dùng thử và mô tả cách tiếp cận ứng dụng.

Chương 6: Kết luận – Trình bày kết quả mà khóa luận đạt được, nhưng điều bản thân đã học được khi làm khóa luận và đưa ra định hướng tiếp theo của ứng dụng.

# Cơ sở lý thuyết và công nghệ

Ứng dụng chia sẻ kiến thức được xây dựng dựa trên các công nghệ Laravel và Vuejs cùng với cơ sở dữ liệu MySQL. Chương 2 sẽ tập trung giới thiệu và phân tích những công nghệ này.

* 1. Tổng quan về lập trình web và phát triển ứng dụng web

Một ứng dụng web là một trình ứng dụng mà có thể tiếp cận qua web thông qua mạng như Internet hay intranet. Chẳng hạn, ứng dụng web được dùng để hiện thực Webmail, bán hàng trực tuyến, đấu giá trực tuyến, wiki, diễn đàn thảo luận, Weblog, MMORPG, hệ quản trị nội dung, phần mềm quản lý nguồn nhân lực và nhiều chức năng khác.

Dù có nhiều biến thể, một ứng dụng Web thông thường được cấu trúc như một ứng dụng ba lớp. Ở dạng phổ biến nhất, một trình duyệt Web là lớp thứ nhất, một bộ máy sử dụng một vài công nghệ nội dung Web động (như ASP, ASP.NET, CGI, ColdFusion, JSP/Java, PHP, Python, hoặc Ruby On Rails) là lớp giữa, và một cơ sở dữ liệu là lớp thứ ba. Trình duyệt sẽ gửi yêu cầu đến lớp giữa, lớp giữa sẽ phục vụ bằng cách tạo ra truy vấn và cập nhật cơ sở dữ liệu và tạo ra giao diện người dùng.

Ranh giới giữa một ứng dụng web và một website thực ra rất mong manh, và giữa 2 khái niệm có nhiều sự giao thoa dễ gây ra sự nhầm lầm. Để phân biệt 2 khái niệm này, chúng ta điểm lại một chút về khái niệm website trước.

Một website được viết lên dựa trên khung là HTML và các tài nguyên như hình ảnh, âm thanh, video,…, nó bao gồm nhiều trang web khác nhau. Tập hợp các webpage đơn lẻ như vậy được liên kết với nhau qua các đường dẫn HTML và hợp thành một website lớn và sử dụng chung một tên miền được gọi là Website. Website có tính tương tác thấp, ít chức năng, chỉ dùng để lưu giữ và hiển thị thông tin cho người truy cập chứ không có khả năng tương tác giữa người dùng với hệ thống hoặc giữa người dùng với nhau.

Trong khi đó, một ứng dụng web có thể hiểu là một ứng dụng chạy trên nền Web. Mà ứng dụng là một loại chương trình có khả năng làm cho máy tính thực hiện trực tiếp một công việc nào đó người dùng muốn thực hiện. Từ đó ta có thể hiểu ứng dụng web được tạo ra với mục đích cung cấp khả năng thực hiện những điều người dùng mong muốn, tăng khả năng tương tác giữa người dùng với hệ thống cũng như giữa người dùng với nhau. Webapp được tạo ra gồm cả 2 phần: phần Backend xây dựng bằng các ngôn ngữ lập trình web như PHP, Python, Java, .NET… còn phía người dùng sử dụng HTML/CSS/JS…

Lấy một ví dụ cho dễ hiểu, một trang báo mạng như Dân trí chẳng hạn. Người đọc báo khi truy cập vào dantri.com.vn thì họ được quyền xem thông tin các bài báo và bình luận về nội dung… cho nên đối với họ thì Dân trí là một website. Tuy nhiên admin hoặc biên tập viên của báo Dân trí thì sẽ coi nó là ứng dụng web. Bởi vì họ có quyền quản trị và được thực hiện nhiều thao tác trên hệ thống cũng như khả năng truy cập và sử dụng dữ liệu của dantri.com.vn.

Lập trình web là công việc tiếp nhận kết quả giao diện web của bên phía thiết kế Web kết hợp với yêu cầu của khách hàng để chuyển thành một hệ thống Website hoàn chỉnh. Website này có thể là Web tĩnh (chỉ có HTML, CSS, JS, không có người dùng hay là tương tác với cơ sở dữ liệu) hoặc là có sự tương tác với cơ sở dữ liệu cũng như khả năng tương tác với người dùng (hay còn gọi là web động – sử dụng các ngôn ngữ lập trình web như PHP, .NET, Ruby…).

Khác với thiết kế Web là công việc của người thiết kế – những người có khả năng sử dụng các phần mềm đồ họa và thiết kế như là Photoshop, AI, Flash… để xây dựng nên giao diện cho website dưới dạng ảnh hoặc một dạng tường minh chi tiết theo ý tưởng và yêu cầu của khách hàng.

Như vậy, chúng ta có thể hình dung luồng quy trình trong lập trình Web sẽ như sau: Những người thiết kế làm việc với khách hàng để cho ra giao diện Web bằng ảnh hoặc tệp photoshop, sau đó người lập trình viên sẽ biến giao diện này thành một website hoàn chỉnh theo yêu cầu khách hàng.

* 1. Vuejs
     1. Single-page Application (SPA)

SPA là một ứng dụng web hoặc trang web tương tác với người dùng bằng cách tự động thay đổi một phần nội dung của trang hiện tại thay vì tải toàn bộ trang mới từ máy chủ. Cách tiếp cận này tránh gián đoạn trải nghiệm người dùng giữa các trang tiếp theo, làm cho ứng dụng hoạt động giống như một ứng dụng máy tính để bàn. Trong một SPA, hoặc là tất cả các mã cần thiết - HTML, JavaScript, và CSS - được lấy ra với một trang tải hoặc tài nguyên thích hợp được tự động nạp và thêm vào trang khi cần thiết, thường là để phản hồi với hành động của người dùng. Trang không tải lại ở bất kỳ thời điểm nào trong quy trình, cũng như không kiểm soát chuyển sang trang khác.

Có rất nhiều bộ khung (framework) của javascript có thể triển khai một ứng dụng SPA như là VueJs, Angular, React,... , ở khóa luận này tôi đã sử dụng Angular 2.

* + 1. Giới thiệu về Angular 2

Angular 2 là một framework (bộ khung) phát triển trên nền tảng javascript của google kế thừa các đặc điểm của AngularJs và phát triển một phương thức tiếp cận việc xây dựng ứng dụng hoàn toàn mới, phương pháp hướng thành phần(Component).

"AngularJS là một framework có cấu trúc cho các ứng dụng web động. Nó cho phép bạn sử dụng HTML như là ngôn ngữ mẫu và cho phép bạn mở rộng cú pháp của HTML để diễn đạt các thành phần ứng dụng của bạn một cách rõ ràng và súc tích. Hai tính năng cốt lõi: Data binding (liên kết dữ liệu) và Dependency injection (nhúng phụ thuộc) giúp loại bỏ phần lớn mã nguồn mà bạn thường phải viết. Nó xảy ra trong tất cả các trình duyệt, làm cho nó trở thành đối tác lý tưởng của bất kỳ công nghệ Server nào."

Angular 2 là đơn giản hơn so với AngularJS và khái niệm của nó làm cho nó dễ dàng hơn để hiểu. Bạn có thể cập nhật các bộ dữ liệu lớn với bộ nhớ tối thiểu. Nó sẽ tăng tốc độ tải ban đầu thông qua cơ chế rendering trên server.

Trên tất cả Angular 2 nổi bật lên bởi các tính năng như đa nền tảng, tốc độ và hiệu năng cao, hỗ trợ phát triển toàn diện

Đầu tiên, Angular 2 phát triển ứng dụng trên mọi nền tảng. Với Angular 2, lập trình viên có thể xây dựng ứng dụng web hiện đại với hiệu năng cao, có khả năng ngoại tuyến. Bên cạnh đó, nó còn giúp phát triển các ứng dụng trên thiết bị di động giống như Ionic Framework, NativeScript hay React Native. Không chỉ vậy, với Angular 2 lập trình viên có thể dễ dàng xây dựng ứng dụng cài đặt trên hệ điều hành Mac, Windows, Linux với những phương thức dành cho ứng dụng web cùng với khả năng truy cập vào API gốc của hệ điều hành [CITATION Seb16 \l 1033].

Về tốc độ và hiệu năng, Angular 2 đã đạt được tốc độ tối đa nhất có thể trên nền tảng web ngày nay và tiếp tục trong tương lai, nhờ vào Web Worker và phần rendering phía máy chủ. Ngôn ngữ của Angular 2 phù hợp với ngữ nghĩa, cú pháp viết mã dễ dàng làm giảm độ phức tạp cho công việc viết mã. Angular có cấu trúc đơn giản, dễ học và có nhiều khuôn mẫu. Việc định tuyến các đường dẫn cũng khá đơn giản. Ngoài ra, Angular 2 còn hỗ trợ điều hướng, hiệu ứng chuyển động và cơ chế tự động lấy dữ liệu (lazy loading) cho trình duyệt di động.

Một đặc tính nổi bật của Angular 2 phải kể đến đó là hỗ trợ phát triển toàn diện. Đầu tiên, về kiểm thử, Angular 2 sử dụng Karma cho các unit tests, bạn có thể biết về nó nếu bạn làm hỏng thứ gì đó mỗi lần bạn lưu lại ứng dụng. Không chỉ có Karma[ CITATION Kar17 \l 1033 ], Angular 2 còn tích hợp Protractor [ CITATION Pro17 \l 1033 ] để giúp các kịch bản test chạy nhanh và ổn định hơn. Về hiệu ứng chuyển động, Angular cung cấp khả năng tạo ra hiệu ứng chuyển động phức tạp một cách dễ dàng thông qua API, cho phép xử lý thay đổi hiệu ứng theo những tương tác từ người dùng.

* + 1. Những thành phần chính của Angular 2
* Component:

Là thành phần chính của angular 2. Component là các khối xây dựng lên ứng dụng Angular 2. Nó biểu diễn một phần có thể tái sử dụng của UI, thường là một một phần tử tùy chỉnh HTML. Một component thành lập bởi ít nhất một phần HTML cái được biết là mẫu (template), một class đóng gói dữ liệu và các tương tác với template, và selector (tên phần tử tùy chỉnh siêu văn bản). Angular 2 sẽ tạo, cập nhật hoặc hủy bỏ component thông qua thao tác của người sử dụng. Ứng dụng sẽ thực hiện hành động đó dựa theo vòng đời có thể được cài đặt thêm.

* Module:

Các ứng dụng Angular 2 mang tính mô hình hóa và Angular 2 có hệ thống module được gọi là Angular Modules hay NgModules. Mỗi hệ thống Angular 2 phải có ít nhất một module gốc, gọi là root module. Các module giúp tổ chức ứng dụng thành các khối chức năng liên kết chặt chẽ với nhau.

* Template:

Khi định nghĩa một component, ta sẽ có template riêng biệt đi liền với nó. Template này sẽ tồn tại dưới dạng HTML và sẽ cho Angular biết phải chuyển đến component hiển thị ra sao. Một template sẽ tương tự với HTML thông thường, ngoại trừ một số điểm khác biệt như cách hiển thị các dữ liệu từ component, truyền dữ liệu vào component khác, dữ liệu trong vòng lặp, các sự kiện cần xử lý.

* Data binding:

Là cơ chế phối hợp giữa từng bộ phận của template và bộ phận của component. Trong quá trình thao tác dữ liệu khi có bất kỳ sự thay đổi nào ở model, dữ liệu sẽ được phản ánh ngay tức và thể hiện dễ dàng cho người dùng nhìn thấy đó chính là Template và ngược lại

Two-way binding sử dụng dữ liệu hai chiều (tự động đồng bộ giữa model và view)

* Service:

Là một lớp đóng gói một chức năng và cung cấp nó cho phần còn lại của ứng dụng. Nó chỉ là một lớp nhỏ định nghĩa mục đích cần được cung cấp.

* Dependency injection:

Là cách để tạo các đối tượng phụ thuộc vào các dối tượng khác. Hệ thống này cung cấp các đối tượng phụ thuộc (hay gọi là các thánhf phần phụ thuộc) khi nó tạo ra một thể hiện của đối tượng. Đa số các phụ thuộc đó là dịch vụ. Angular 2 sử dụng dependent injection để cung cấp cho component nhưng dịch vụ mà chúng cần.

Khi Angular tạo một component, đầu tiên nó sẽ yêu cầu về injector cho service mà component đó yêu cầu. Một injector duy trì một gói các thực thể dịch vụ mà nó tạo ra trước đó. Nếu một thực thể service được yêu cầu không chứa trong gói đó, injector tạo ra nó và thêm nó vào gói trước khi trả về service cho Angular. Khi tất cả các yêu cầu về service được giải quyết và trả về, Angular có thể gọi khởi tạo với các service như biến truyền vào.

* 1. Ruby và Ruby on Rails Framework
     1. Ruby là gì?

Ruby là một ngôn ngữ lập trình hướng đối tượng được tạo ra bởi Yukihiro Matsumoto. Không giống như các ngôn ngữ tiền nhiệm, khi mà vẫn có gì đó liên quan đến lập trình cấu trúc (C++, C#) hoặc vẫn tồn tại các kiểu dữ liệu nguyên thuỷ (Java). Có thể nói Ruby là một ngôn ngữ thuần hướng đối tượng, khi mà cả số 0 hay 1 đều là một đối tượng. Ruby được thiết kế dựa trên con người hơn là máy tính, các công việc được tối giản đến mức thấp nhất có thể, các tên gọi của biến, phương thức đều rất gần gũi với ngôn ngữ tự nhiên. Ruby là một ngôn ngữ thông dịch, tức là thay vì cần phải biên dịch ra một tệp tin thực thi (như C/C++, C# ...) hoặc ra mã bytecode (như Java), nó hoạt động tương tự như Perl hoặc Python, các dòng mã sẽ được thực thi ngay tại thời điểm gọi đến. Ruby ra đời dựa trên đặc tính cơ bản của 2 ngôn ngữ, nó kế thừa cú pháp từ Perl và ngữ nghĩa của Smalltalk, và mang một triết lý khác hẳn với Python. Ngày nay, với sự phổ biến của nguyên lý lập trình hướng đối tượng (Object Oriented Programming-OOP), Ruby là một ngôn ngữ nên được cân nhắc để các lập trình viên làm quen với xu thế này, nhưng không nhất định. Cái làm cho Ruby trở nên nổi tiếng có lẽ không phải bản thân ngôn ngữ này mà là kho thư viện bổ sung của nó - hệ thống thư viện (GEM). Điều này có lẽ không còn xa lạ gì với các lập trình viên Ruby, đặc biệt với sự ra đời và phát triển không ngừng của một gem đặc biệt - Ruby on Rails. Đây có lẽ là Framework nổi tiếng nhất trong cộng đồng lập trình viên Ruby/Rails.

Ruby đã tạo nên những nét đặc biệt nhất của ngôn những ngôn ngữ lập trình.

Những nét chính đó là:

Sức mạnh: là sự kết hợp hoàn hảo giữa hướng đối tượng thuần túy của ngôn ngữ hướng đối tượng với sức diễn đạt và sự tiện lợi của ngôn ngữ kịch bản (Script) của Perl. Chương trình Ruby rất nhỏ gọn, dễ đọc.

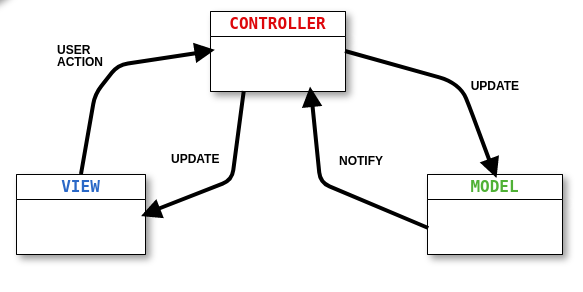
Tính đơn giản: Cú pháp và nghĩa của nó rất trực quan và dễ đọc (clean). Và cũng không có “những trường hợp đặc biệt” nào mà ta cần phải ghi nhớ. Như các instance, số nguyên (integer), hay các lớp (class), vừa đủ giống với các ngôn ngữ khác. Mỗi khi ta học những cái cơ bản, nó rất dễ dàng để đoán trước được cách làm những cái mới hơn.

Tính vô hình: Ruby giải phóng người lập trình khỏi sự cực nhọc trong việc nhồi nhét của các trình biên dịch. Ruby luôn năm trong vùng kiểm soát, do đó ta có thể tập trung giải quyết các lỗi bằng tay.

Sẵn có: Ruby là nguồn mở nên có thẻ sử dụng một cách tự do đối với người dùng hay người phát triển. Không giống như nhiều ngôn ngữ khác, Ruby không giới hạn về vấn đề hệ điều hành và pháp lý. Ruby có thể chạy trên Unix hay Linux, Microsoft- Windows, MacOS.

* + 1. Ruby on Rails framework
       1. Khái niệm

Ruby On Rails hay còn được gọi là ROR hay Rails là một framwork phát triển ứng dụng web được viết bằng ruby. Nó được thiết kế để làm cho các ứng dụng web được lập trình dễ dàng hơn bằng cách giả lập ra những gì nhà phát triển cần để bắt đầu bao gồm nhiều thư viện liên kết.

* + - 1. Các tính năng chính
* Kiến trúc MVC: Rails sử dụng kiến trúc theo mô hình MVC - mô hình (Model) – Xem (View) – bộ điều khiển (Controller).
* - Model: Là thành phần chịu trách nhiệm xử lý các thao tác trên database. Và gởi trả kết quả thông qua view.
* - View: Là phần hiển thị thông tin trên website, sau khi đi qua controller và nhận kết quả từ phía model thì view là bước cuối cùng để chuyển thông tin tới người dùng.
* - Controller: Là phần điều hướng các request tới những tác vụ tương ứng. Controller là một phần không thể thiếu ở bất cứ framework nào. Vì nó có trách nhiệm gởi và nhận request từ hệ thống tới người sử dụng.
* 

Hình 2.1: Mô hình MVC

* Thư viện truy xuất dữ liệu:
* Debugging
* Representational Sate Transfer (REST) cho các web services
* Hỗ trợ nhiều hệ quản trị cơ sở dữ liệu lớn như MySQL, Oracle, MS SQL, PostgreSQL…
* Ngôn ngữ script Ruby được viết bên phía Server
* Sử dụng các quy ước thay vì phải cấu hình
* Có các bộ sinh script để tự động thực hiện các task
  + - 1. Các thành phần của Rails
* Action Pack

Là một thư viện (gem) đơn bao gồm Action Controller, Action View, Controller Dispatch. Cung cấp các lớp ở dưới tầng controller và view trong mô hình MVC. Những module này tiếp nhận yêu cầu từ trình duyệt và sau đó ánh xạ đến các action tương ứng được định nghĩa ở controller và sau đó các action này sẽ hiển thị lên trình duyệt.

* Action Mailer:

Module này chịu trách nhiệm cung cấp các dịch vụ về mail, chúng ta có thể sử dụng để nhận và xử lý Email gửi đến và gửi đi một đoạn văn đơn giản hoặc email có nhiều dữ liệu phức tạp dựa trên các khuôn mẫu linh hoạt.

* Active Model

Định nghĩa interface giữa Action Pack và Active Record. Giao diện của active record được dùng ở bên ngoài Rails, cung cấp các chức năng ORM(Object relation mapping) – là một kỹ thuật giúp chuyển đồi dữ liệu giữa các hệ thống không tương thích được viết bởi các ngôn ngữ hướng đối tượng.

* Active Record

Được sử dụng để quản lý dữ liệu trong các cơ sở dữ liệu quan hệ thông qua các đối tượng, là cơ sở cho các model trong ứng dụng Rails, nó cung cấp dữ liệu độc lập, khả năng tìm kiếm dữ liệu nâng cao, kết nối các model lại với nhau.

* Active Resource

Quản lý các kết nối giữa RESTful web services và các đối tượng nghiệp vụ, giúp tối ưu mã nguồn và số lượng mã nguồn. Nó thực hiện ánh xạ tài nguyên dựa trên web tới những đối tượng cục bộ.

* Active Suport

Là các tiện ích và các thư viện chuẩn của ruby, nó rất hữu ích trong quá trình phát triển.

* Railties

Đảm nhận việc kết nối các module lại với nhau, là lõi để xây dựng ứng dụng rails.

* 1. MySQL
     1. Hệ quản trị cơ sở dữ liệu

Cơ sở dữ liệu là tập hợp các dữ liệu được tổ chức theo một cấu trúc nhất định để có thể dễ dàng quản lý. Hệ quản trị cơ sở dữ liệu là một chuơng trình phần mềm hay hệ thống được thiết kế để quản trị một cơ sở dữ liệu. Hệ quản trị cơ sở dữ liệu hỗ trợ khả năng lưu trữ, sửa chữa, xóa và tìm kiếm thông tin trong một cơ sở dữ liệu. Có nhiều loại hệ quản trị cơ sở dữ liệu khác nhau như MySQL, Oracle, PostgreSQL, SQL Server,…

* + 1. MySQL là gì

MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập cơ sở dữ liệu trên internet...

Đặc điểm:

* Là một mã nguồn mở.
* Là một chương trình rất mạnh mẽ
* Sử dụng form chuẩn của ngôn ngữ dữ liệu SQL
* Làm việc trên nhiều hệ điều hành với nhiều ngôn ngữ
* Làm việc nhanh và khỏe với các tập dữ liệu lớn.
* Rất thân thiện, dễ dàng tìm hiểu để phát triển web
* Hỗ trợ các cơ sở dữ liệu lớn.

# Phân tích yêu cầu bài toán

Chương này mô tả tổng quan về bài toán, thu thập các yêu cầu của người dùng về phần mềm, thiết kế các ca sử dụng và phân tích chi tiết ca sử dụng.

* 1. Mô tả bài toán

Bài toàn đặt ra là cần xây dựng một ứng dụng dựa trên ý tưởng của phần mềm tinycards của Duolingo giúp người dùng học nhanh kiến thức thông qua các thẻ học. Ứng dụng có người dùng là người học từ vựng, khái niệm muốn trau dồi vốn từ vựng, kiến thức và chia sẻ với cộng đồng những kiến thức của bản thân.

* 1. Thu thập yêu cầu

Yêu cầu người dùng

* Là người dùng, tôi muốn đăng ký tài khoàn vào ứng hệ thống để có thể sử dụng hệ thống
* Là người dùng, tôi muốn đăng nhập vào ứng dụng để sử dụng ứng dụng
* Là người dùng, tôi muốn tạo ra các bộ thẻ học từ vựng cho tôi và mọi người cùng học trên ứng dụng
* Là người dùng tôi muốn chỉnh sửa bộ thẻ mà chính tôi tạo ra
* Là người dùng tôi muốn xem được thông tin về tài khoản, họ tên, email của mình đề đảm bảo thông tin chính xác của tôi
* Là người dùng tôi muốn chỉnh sửa thông tin về tên, email, ảnh đại diện và mật khẩu của tài khoản của tôi.
* Là người dùng tôi muốn được học các bài học theo từng bộ thẻ của ứng dụng
* Là người dùng tôi muốn đăng xuất ra khỏi ứng dụng
* Là người dùng tôi muốn tìm kiếm các bộ thẻ theo tên là mô tả của bộ thẻ.

Yêu cầu người quản trị

* Là người quản trị tôi muốn xem thông tin tất cả người dùng trong hệ thống
* Là người quản trị tôi muốn xem tất cả các bộ thẻ trong hệ thống.
* Là người quản trị tôi muốn thống kê số lượng người dùng, số bộ thẻ, số thẻ trong một tuần hoặc một tháng
* Là người quản trị tôi muốn xem các phản hồi của người dùng về các bộ thẻ.
* Là người quản trị tôi muốn xóa các bộ thẻ có nội dung nhạy cảm, không phù hợp
  1. Phân tích tác nhân chức năng của hệ thống

Sau khi thu thập yêu cầu của hệ thống, tôi đã xác định được các tác nhân của hệ thống gồm: Người dùng và người quản trị.

\* Yêu cầu chức năng

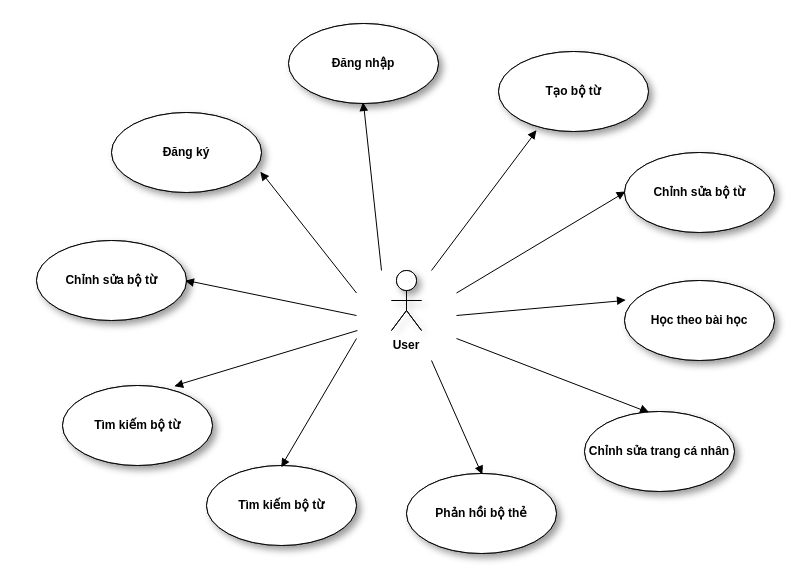
Để diễn tả yêu cầu chức năng của hệ thống, tôi chia chia chức năng theo từng vai trò của người sử dụng cho hợp lý.

Người quản trị:

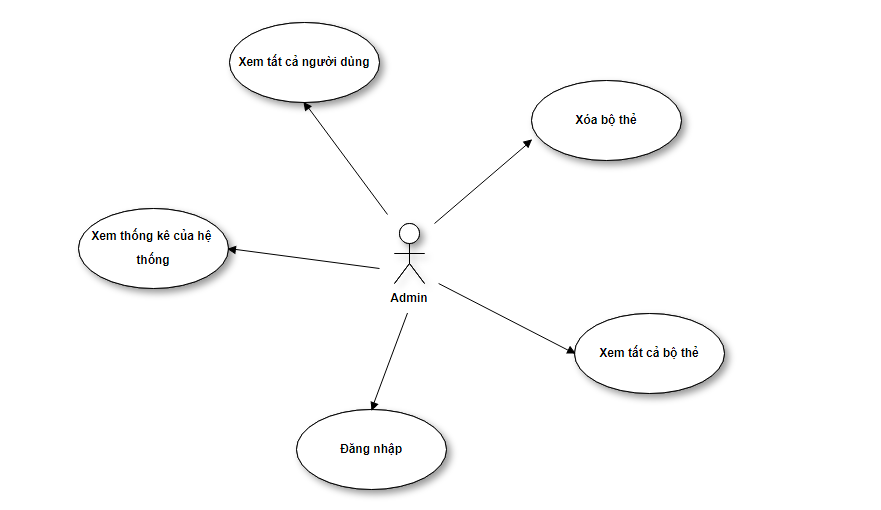
* Đăng nhập: Hệ thống mặc đinh sẽ tạo sẵn một tài khoản cho người quản trị, có thể tạo nhiều hơn một tài khoản người quản trị.
* Quản lý các bộ thẻ: Người quản trị có trách nhiệm quản lý các bộ thẻ của người dùng. Nhận phản hồi của người dùng để kiểm tra nội dung của các bộ thẻ xem bộ thẻ đó có nội dùng xấu, nhạy cảm không để quản lý nó dễ dàng.
* Quản lý người dùng: Người quản trị có trách nhiệm quản lý các người dùng.

Người dùng:

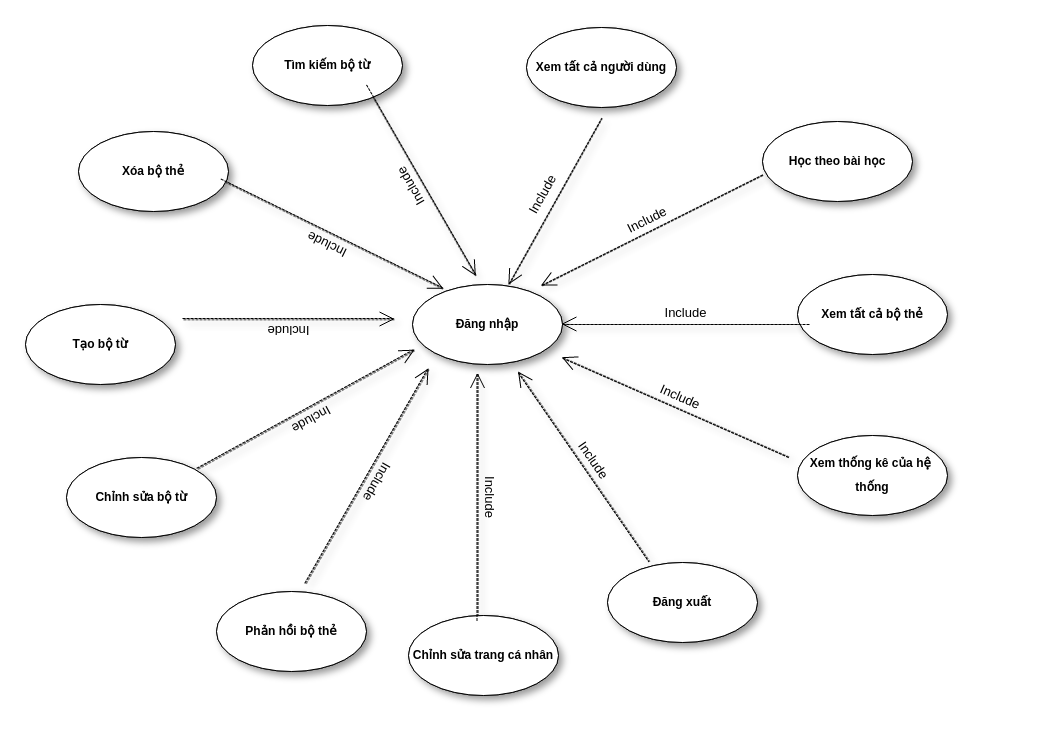
* Đăng nhập: Đăng nhập khi đã có tài khoản trên hệ thống.
* Đăng ký: Đăng ký tài khoản để sử dụng hệ thống.
* Đăng xuất: Đăng xuất tài khoản ra khỏi hệ thống.
* Tạo bộ thẻ mới bằng cách thủ công theo mẫu của hệ thống (form): Người dùng có thể tạo mới một bộ thẻ, người dùng có thể xem được chi tiết nội dung của bộ thẻ. Khi người dùng tạo ra bộ thẻ thì các bài học cũng sẽ được tạo ra dựa trên số thẻ của bộ thẻ.
* Sửa bộ thẻ: Khi bộ thẻ do chính người dùng tạo ra thì người dùng sẽ có quyền chỉnh sửa bộ thẻ đó. Sau khi chỉnh sửa, mọi dữ liệu của các bài học được tạo ra từ dữ liệu của bộ thẻ cũ sẽ bị mất. Sau khi thay đổi bộ thẻ các bài học cũng được cập nhật lại.
* Nhập dữ liệu từ google sheet để tạo các bộ thẻ: Khi người dùng có nhiều từ nhưng không muốn mất nhiều thời gian để nhập vào khi sử dụng cách thủ công thì người dùng có thể tạo dữ liệu có sẵn ở google sheet và nhập đường dẫn vào hệ thống để tự động các các bộ thẻ.
* Cập nhật thông tin của người dùng: Khi người dùng muốn xem hoặc thay đổi thông tin cá nhân, hoặc người dùng có thể thay đổi mật khẩu cá nhân. Người dùng muốn cập nhật thông tin chỉ cần vào phần cài đặt để tiến hành xem và chỉnh sửa thông tin.
* Tìm kiếm bộ thẻ: Người dùng có thể tìm kiếm các bộ thẻ theo chủ đề hoặc mô tả về bộ thẻ để học và luyện theo chủ đề bằng cách tìm kiếm ở trên thanh công cụ của ứng dụng.
* Học và luyện các bộ thẻ theo từng bài học: Mục đích của ứng dụng là học và luyện kiến thức nên các bộ thẻ sẽ có các bài học để luyện tập ghi nhớ các khái niệm, các kiến thức của bộ thẻ đó. Người dùng sẽ vào bài học ở trong bộ thẻ, người dùng chỉ được phép học các bài học khi mà các bài học trước của cùng một bộ thẻ đã học xong.
* Xóa bộ thẻ: Quản trị viên có thể xóa các bộ thẻ của người dùng khi nội dung của bộ thẻ không phù hợp. Người dùng chỉ xóa được những bộ thẻ do chính người dùng tạo ra. Để xóa bộ thẻ người dùng vào chỉnh sửa bộ thẻ và nhấn vào xóa để xóa bộ thẻ.
* Theo dõi người dùng: Khi người dùng muốn những bộ thẻ của những người mà mình thấy sở hữu những bộ thẻ có ích thì người dùng sẽ theo dõi bằng cách nhấn vào nút theo dõi (Follow) ở trang cá nhân của người đó.
* Phản hồi với những bộ thẻ có nội dùng không phù hợp: Trong quá trình người dùng học và luyện các bộ thẻ, nếu người dùng thấy bộ thẻ nào có nội dung không phù hợp, nội dung nhạy cảm. Người dùng phản hồi bằng cách ở trang giao diện của bộ thẻ đó, người dùng vào tất cả các thẻ của bộ thẻ sẽ có nút để phản hồi, người dùng chọn nội dung phản hồi tương ứng với bộ thẻ đó.
* \* Yêu cầu phi chức năng
* Hệ thống cần được xây dựng với các yêu cầu phi chức năng như sau:
* Ruby on Rails: Hệ thống được xây dựng trên nên tảng framework Ruby on Rails với phiên bản 5.1 của Ruby với phiên bản 2.4 và Angular 2 phiên bản 2
* Xác thực và phần quyền: Hệ thống cần có sự bảo mật với cái vai trò nhất định. Người dùng sẽ chỉ có quyền tạo bộ thẻ và ôn luyện chúng, không thể thực hiện các chức năng của người quản trị như xem toàn bộ người dùng hoặc xóa các bộ thẻ không phù hợp
* Triển khai hệ thống: Mã nguồn của hệ thống được lưu giữu trên github và đã được triển khai trên thực tế để người dùng sử dụng.
* Hiệu năng hoạt động của hệ thống: Hệ thống cần hoạt động ổn định với dữ liệu lên tới hàng nghìn người cùng truy cập và sử dụng, thời gian tương tác nhanh, tài nguyên sử dụng hợp lý.
* Tương thích: Tường tích trên trình duyệt của các thiết bị di dộng mà máy tính để bàn.
  1. Phân tích yêu cầu
     1. Thiết kế ca sử dụng



Hình 3.2: Mô hình ca sử dụng của người dùng



Hình 3.3: Mô hình ca sử dụng của quản trị viên



Hình 3.4 Ràng buộc giữa các use-case

* + 1. Phân tích biểu đồ ca sử dụng

Ca sử dụng đăng nhập

Bảng 3.1: Mô tả ca sử dụng đăng nhập

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Đăng nhập vào ứng dụng | |
| **Tác nhân** | Người dùng | |
| **Mô tả** | Người dùng đăng nhập vào ứng dụng với tài khoản đã đăng ký | |
| **Điều kiện tiên quyết** | Thông tin của người dùng đã có trong cơ sở dữ liệu | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Người dùng nhấn vào đăng nhập (Login) ở thanh công cụ |
| 2 | Người dùng nhập thông tin email và mật khẩu đã đăng ký vào biểu mẫu có sẵn |
| 3 | Người dùng bấm vào nút đăng nhập (Login) trên màn hình |
| 4 | Hệ thống trả về đăng nhập thành công |
| 5 | Hệ thống lưu sesson cho người dùng |
| **Luồng thay thế** | Tại bước 3 nếu thông tin người dùng nhập không chính xác hệ thống sẽ trả về màn hình đăng nhập | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng đăng ký

Bảng 3.2: Mô tả ca sử dụng đăng ký

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Đăng ký tài khoản vào ứng dụng để có quyền thực hiện tất cả các chức năng của người dùng | |
| **Tác nhân** | Người dùng | |
| **Mô tả** | Người dùng đăng ký tài khoản vào ứng dụng | |
| **Điều kiện tiên quyết** | Thông tin người dùng chưa có trong cơ sở dữ liệu | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Người dùng vào phần đăng ký tài khoản (Sign up) |
| 2 | Người dùng nhập thông tin về tên, email, mật khẩu theo đúng định dạng trên biểu mẫu ở màn hình |
| 3 | Người dùng bấm vào nút đăng ký (Sign Up) |
| 4 | Hệ thống trả về đăng ký thành công |
| **Luồng thay thế** | Tại bước 3 nếu người dùng đăng ký tài khoản theo email đã được sử dụng để đăng ký rồi thì hệ thống sẽ không đăng ký thành công và giữ nguyên giao diện đăng ký | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng Đăng xuất

Bảng 3.3: Mô tả ca sử dụng đăng xuất

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Đăng xuất khỏi ứng dụng | |
| **Tác nhân** | Người dùng | |
| **Mô tả** | Người dùng đăng xuất khỏi ứng dụng | |
| **Điều kiện tiên quyết** | Người dùng đã đăng nhập | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Người dùng chọn Đăng xuất (Logout) |
| 2 | Hệ thống trả về đăng xuất thành công và trả về giao diện đăng nhập với người dùng |
| **Luồng thay thế** | Không có | |
| **Điều kiện sau** | Hệ thống gửi các client đã tải thông tin về người dùng trước đó thông điệp là người dùng này đã thoát khỏi hệ thống | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng Tạo bộ thẻ

Bảng 3.4: Mô tả ca sử dụng tạo bộ thẻ

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Tạo ra một bộ các thẻ để học và chia sẻ cho mọi người theo từng chủ để | |
| **Tác nhân** | Người dùng | |
| **Mô tả** | Người dùng có thể tự tạo bộ thẻ theo những chủ đề | |
| **Điều kiện tiên quyết** | Không có | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Người dùng chọn Tạo bộ thẻ (Create Deck) ở thanh công cụ |
| 2 | Hệ thống trả về giao diện Tạo bộ thẻ |
| 3 | Người dùng nhập thông tin của bộ thẻ bao gồm chủ đề(title), mô tả(description) và thêm các thẻ học có các thông tin là mặt trước, mặt sau và ảnh của thẻ theo đúng định dạng. |
| 4 | Người dùng bấm vào nút Tạo (Create) trên giao diện |
| 5 | Hệ thống trả về tạo bộ thẻ thành công và trả về giao diện của bộ thẻ vừa mới tạo |
| **Luồng thay thế** | Tại bước 4 nếu thông tin của bộ thẻ chưa đầy đủ hoặc thiếu thẻ thì hệ thống giữ nguyên giao diện và không tạo thành công | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng học theo bài học

Bảng 3.5: Mô tả ca sử dụng học theo bài học

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Người dùng luyện tập theo những bài học từ vựng | |
| **Tác nhân** | Người dùng | |
| **Mô tả** | Người dùng có thể luyện tập để ôn lại kiến thức từ vựng theo những bài học của hệ thống | |
| **Điều kiện tiên quyết** | Người dùng đã đăng nhập vào hệ thống | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Trên giao diện của một bộ thẻ, người dùng bấm vào bài học muốn học. |
| 2 | Hệ thống trả về giao diện của bài học đó |
| 3 | Số các câu hỏi của một bài học tỷ lệ thuận với số thẻ mà nó chứa. Người dùng và hệ thống sẽ luân phiên trả lời các câu hỏi của bài học và kiểm tra độ chính xác của câu hỏi cho đến khi hoàn thành khóa học.  - Mở đầu của một bài học, người dùng sẽ được học nội dung của các thẻ bằng cách bấm vào thể để xem nội dung 2 mặt của thẻ.  - Kế tiếp đó sẽ là các câu hỏi kiểm tra lại nội dung vừa học (các câu hỏi có thể câu hỏi điền từ, câu hỏi chọn đáp án đúng) – hệ thống đảm bảo rằng luôn có đầy đủ các dạng câu hỏi cho mỗi thẻ.  - Khi trả lời câu hỏi nếu người dùng trả lời đúng ở lần đầu tiên với thẻ được hỏi thì thanh trạng thái thể hiện điểm số sẽ tăng lên còn nếu trả lời sai thì thanh trạng thái sẽ giữ nguyên và hệ thống sẽ hoặc là cho người dùng biết đáp án ( với dạng trả lời sai câu hỏi điền từ) hoặc là sẽ đợi cho đến khi người dùng trả lời đúng thì thôi (với dạng câu hỏi chọn đáp án đúng) và sau đó cho người dùng học lại thẻ đã trả lời sai.  - Bài học được coi là hoàn thành khi mà người dùng hoàn thành các câu hỏi của bài học đó (thanh trạng thái hoàn thành). |
| 4 | Hệ thống chuyển đến trang bộ thẻ đang học |
| **Luồng thay thế** | Tại bước 3, nếu người dùng không muốn tiếp tục học bài học thì bấm vào hình dấu nhân để thoát ra bài học. Sau đó hệ thống sẽ trở lại trang bộ thẻ | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng Chỉnh sửa bộ thẻ

Bảng 3.6: Mô tả ca sử dụng chỉnh sửa bộ thẻ

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Người dùng có thể chỉnh sửa những bộ thẻ do chính người dùng tạo ra | |
| **Tác nhân** | Người dùng | |
| **Mô tả** | Người dùng có thể chỉnh sửa các bộ thẻ để thêm các thẻ hoặc chỉnh sửa nội dùng của bộ thẻ | |
| **Điều kiện tiên quyết** | Người dùng đã đăng nhập hệ thống | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Trên giao diện của một bộ thẻ, người dùng bấm vào nút chỉnh sửa (icon biểu thị nút chỉnh sửa) |
| 2 | Hệ thống trả về giao diện của trang chỉnh sửa bộ thẻ |
| 3 | Người dùng chỉnh sử nội dung của bộ thẻ là thông tin chủ để, mô tả bộ thẻ và thông tin của các thẻ, xóa thẻ cũ thêm thẻ mới sau đó nhấn nút Thay đổi (Update) |
| 4 | Hệ thống cập nhật lại thông tin của bộ thẻ và chuyển đến trang bộ thẻ vừa chỉnh sửa |
| **Luồng thay thế** | Không có | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng Tìm kiếm bộ thẻ

Bảng 3.7: Mô tả ca sử dụng tìm kiếm bộ thẻ

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Người dùng có thể tìm kiếm bộ thẻ muốn tìm nhanh hơn bằng các từ khóa | |
| **Tác nhân** | Người dùng | |
| **Mô tả** | Người dùng có thể tìm kiếm các bộ thẻ bằng từ khóa là tiêu để hoặc mô tả của bộ thẻ | |
| **Điều kiện tiên quyết** | Người dùng đã đăng nhập và hệ thống | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Trên giao diện trang web, người dùng gõ từ khóa tìm kiếm vào thanh công cụ tìm kiếm ở đầu trang |
| 2 | Hệ thống tìm kiếm và trả về tất cả các bộ thẻ theo từ khóa tìm kiếm của người dùng. |
| 3 | Hệ thống chuyển đến trang chứa tất cả bộ thẻ đã tìm được. |
| **Luồng thay thế** | Không có | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng Chỉnh sửa trang cá nhân

Bảng 3.8: Mô tả ca sử dụng chỉnh sửa trang cá nhân

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Người dùng có thể chỉnh sửa những thông tin trang cá nhân của mình | |
| **Tác nhân** | Người dùng | |
| **Mô tả** | Người dùng có thể chỉnh sửa thông tin về email, tên, mật khẩu và mô tả người dùng | |
| **Điều kiện tiên quyết** | Người dùng đã đăng nhập vào hệ thống | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Trên giao diện của một bộ trang cá nhân, người dùng bấm vào nút cài đặt (Settings) |
| 2 | Hệ thống trả về giao diện của trang chỉnh sửa thông tin cá nhân của người dùng |
| 3 | Người dùng chỉnh sửa thông tin cá nhân gồm email tên, mô tả thông tin và nhấn vào nút Thay đổi (Change). Nếu người dùng muốn thay đổi mật khẩu thì người dùng vào tab đổi mật khẩu nhập các thông tin mật khẩu hiện tại, mật khẩu mới để thay đổi. |
| 4 | Hệ thống cập nhật thông tin và chuyển đến trang cá nhân của người dùng |
| **Luồng thay thế** | Không có | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng phản hồi của người dùng

Bảng 3.9: Mô tả ca sử dụng phản hồi của người dùng

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Người dùng có thể phản hồi những bộ thẻ không phù hợp | |
| **Tác nhân** | Người dùng | |
| **Mô tả** | Trong quá trình học và luyện, người dùng nhận thấy bộ thẻ có nội dung không phù hợp để người quản trị xem xét loại bỏ bộ thẻ ra khỏi hệ thống | |
| **Điều kiện tiên quyết** | Không có | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Trên giao diện của một bộ thẻ, người dùng chuyển sang tab tất cả thẻ |
| 2 | Hệ thống trả về giao diện tất cả thẻ, người dùng bấm vào nút “…” |
| 3 | Hệ thống sẽ hiển thị lên một hộp thoại chứa các loại phản hồi như để người dùng chọn loại phản hồi |
| 4 | Người dùng nhấn chọn loại để phản hồi và nhấn vào gửi (Submit) |
| 5 | Hệ thống thông báo phản hồi thành công |
| **Luồng thay thế** | Không có | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng xem tất cả bộ thẻ

Bảng 3.10: Mô tả ca sử dụng xem tất cả bộ thẻ

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Người quản trị viên có thể xem thông tin của tất cả các bộ thẻ có trong hệ thống | |
| **Tác nhân** | Người quản trị viên | |
| **Mô tả** | Để nắm rõ hệ thống và quản lý tốt hơn, quản trị viên có thể xem tất cả các bộ thẻ của hệ thống chứa các thông tin về chủ đề, mô tả, tổng số thẻ của bộ thẻ. | |
| **Điều kiện tiên quyết** | Không có | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Trên giao diện của quản trị viên, quản trị viên truy cập vào mục tất cả bộ thẻ (List Decks) |
| 2 | Hệ thống trả về giao diện của trang chứa tất cả các bộ thẻ của hệ thống. |
| **Luồng thay thế** | Không có | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng xem tất cả người dùng

Bảng 3.11: Mô tả ca sử dụng xem tất cả người dùng

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Quản trị viên có thể xem thông tin của tất cả người dùng trong hệ thống | |
| **Tác nhân** | Quản trị viên | |
| **Mô tả** | Quản trị viên có thể xem thông tin chi tiết của người dùng để có thể quản lý | |
| **Điều kiện tiên quyết** | Không có | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Trên giao diện của quản trị viên, quản trị viên truy cập vào tất cả người dùng (List Users) |
| 2 | Hệ thống trả về giao diện của trang tất cả người dùng |
| **Luồng thay thế** | Không có | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng xem tất cả phản hồi bộ thẻ

Bảng 3.12: Mô tả ca sử dụng xem tất cả phản hồi của bộ thẻ

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Quản trị viên sẽ xem tất cả các phản hồi của người dùng về bộ thẻ | |
| **Tác nhân** | Quản trị viên | |
| **Mô tả** | Quản trị viên xem tất cả các phản hồi để xem xét có xóa các bộ thẻ không | |
| **Điều kiện tiên quyết** | Không có | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Trên giao diện trang web của người quản trị, người dùng vào trang tất cả phản hồi (List Reports) |
| 2 | Hệ thống trả về giai diện tất cả phản hồi của người dùng chứa các thông tin về người phản hồi, bộ thẻ bị phản hồi. |
| **Luồng thay thế** | Không có | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

# Giải quyết bài toán với Ruby On Rails, Angular 2

Để thực hiện các chức năng, yêu cầu như trên, tôi đã sử dụng các công nghệ mới như Angular 2, Ruby on Rails để giải quyết. Ứng dụng gồm có hai phần là frontend và backend. Trước tiền, phần frontend sử dụng Angular 2, Angular 2 sẽ giúp hệ thống cải thiện được tốc độ và hiệu năng sự dụng. Đối với backend, tôi sử dụng Ruby, cụ thể là Ruby on Rails framework để kết nối với cơ sở dữ liệu sử dụng là MySQL. Cuối cùng là sử dụng rspec làm công cụ kiểm thử phía máy chủ.

* 1. Thiết kế phần mềm
     1. Thiết kế giao diện

Giao diện được thiết kế sao cho đơn giản, hiệu quả và dễ sử dụng thân thiện với người dùng. Vì ứng dụng được sử dụng trên mọi thiết bị máy tính hay điện thoại di động nên giao diện sẽ được thiết kế để phù hợp với tất cả các thiết bị.

Các màn hình được thiết kế sẽ được giới thiệu ở phần Kết quả thực nghiệm ở chương 5.

* + 1. Thiết kế cơ sở dữ liệu

Hệ quản trị cơ sở dữ liệu được sử dụng là MySQL. Nó rất linh động, dễ mở rộng. Mô hình cơ sở dữ liệu được giải thích trực quan như dưới đây:

Ảnh có chứa ảnh chụp màn hình

Mô tả được tạo với mức tin cậy rất cao

Hình 4.5: Mô hình dữ liệu trong cơ sở dữ liệu MySQL

Users

Bảng này lưu các thông tin về người dùng.

Bảng 4.13: Mô tả bảng User

|  |  |
| --- | --- |
| Từ khóa | Ý Nghĩa |
| **Id** | Định danh người dùng |
| **Name** | Tên người dùng |
| **Email** | Địa chỉ email người người dùng |
| **Avatar** | Ảnh đại diện của người dùng |
| **Bio** | Miêu tả những thông tin, giới thiệu khái quát về bản thân người dùng |

Decks

Bảng này lưu các thông tin về các bộ thẻ học. Mỗi bộ thẻ phải thuộc về một người dùng.

Bảng 4.14: Mô tả bảng Decks

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh bộ thẻ |
| **user\_id** | Định danh người dùng tạo ra bộ thẻ |
| **cover\_image** | Ảnh của bộ thẻ |
| **Title** | Chủ đề của bộ thẻ |
| **Description** | Miêu ta ngắn gọn về bộ thẻ |

**Cards**

Bảng này lưu thông tin về các thẻ trong một bộ thẻ. Mỗi một thẻ phải thuộc về một bộ thẻ

Bảng 4.15: Mô tả bảng Cards

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh thẻ |
| **lesson\_id** | Định danh bài học chứ thẻ |
| **deck\_id** | Định danh bộ thẻ chứa thẻ |
| **Front** | Mặt trước của thẻ |
| **Back** | Mặt sau của thẻ |
| **Picture** | Ảnh mô tả của thẻ |

Lessons

Bảng này lưu thông tin về các bài học của một bộ thẻ. Mỗi một bài học phải thuộc một bộ thẻ

Bảng 4.16: Mô tả bảng Lessons

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh bài học |
| **deck\_id** | Định danh của bộ thẻ chứa bài học |

User\_lessons

Bảng này lưu các thông tin về các bài học của mỗi người dùng thuộc về một bộ thẻ. Mỗi User\_lesson thuộc về một người dùng và một bộ thẻ

Bảng 4.17: Mô tả bảng User\_Lessons

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh bài học của người dùng |
| **user\_id** | Định danh của người dùng thực hành bài học |
| **lesson\_id** | Định danh bài học của bộ thẻ |
| **practice\_at** | Thời gian thực hiện bài học |

Feedbacks

Bảng này lưu các thông tin về nội dung phản ảnh của người dùng về bộ thẻ. Mỗi feedback thuộc về một người dùng phản ảnh về một bộ thẻ

Bảng 4.18: Mô tả bảng Feedbacks

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh phản hồi |
| **user\_id** | Định danh người dùng phản hồi |
| **deck\_id** | Định danh của bộ thẻ đựoc phản hồi |
| **Status** | Nội dung phản hồi |

Favorites

Bảng này lưu các thông tin về người dùng yêu thích bộ thẻ.

Bảng 4.19: Mô tả bảng Favorites

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh |
| **user\_id** | Định danh của ngừoi dùng |
| **deck\_id** | Định danh của bộ thẻ |

Relationships

Bảng này lưu các thông tin về quan hệ theo dõi giữa nhưng người dùng.

Bảng 4.20: Mô tả bảng Relationships

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh quan hệ của 2 người dùng |
| **follower\_id** | Định danh của người dùng theo dõi |
| **followed\_id** | Định danh của ngừoi dùng được theo dõi |

Admins

Bảng này lưu các thông tin về người quản lý hệ thống

Bảng 4.21: Mô tả bảng Admins

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh của người quản lý |
| **Email** | Email của người quản lý hệ thống |

* 1. Triển khai phần mềm:
     1. Ứng dụng Angular 2 để xây dựng phía máy khách

Xây dựng cấu trúc ứng dụng

Angular 2 chịu trách nhiệm lấy các dữ liệu để xử lý hoặc trả về cho phía máy chủ để hiển thị giao diện cho người dùng.

Router

Các router định tuyến ứng dụng theo các thao tác của người dùng. Chúng được khai báo trong tập tin app.module.ts. Các router của ứng dụng được liệt kê dưới bảng sau:

Bảng 4.22: Bảng định nghĩa các định tuyến của ứng dụng

|  |  |  |
| --- | --- | --- |
| **STT** | **Đường dẫn** | **Chức năng** |
| 1 | /signup | Đăng ký tài khoản của người dùng |
| 2 | /login | Đăng nhập vào hệ thống |
| 4 | /users/{id} | Thông tin cá nhân của người dùng |
| 5 | /users/{id}/setting | Chỉnh sửa thông tin cá nhân của người dùng |
| 6 | /createDeck | Tạo mới một bộ thẻ |
| 7 | /users/createDeckAdvance | Tạo mới một thẻ nâng cao |
| 8 | /deck/{id} | Thông tin về một bộ thẻ |
| 9 | /decks/{id}/edit | Chỉnh sửa thông tin của bộ thẻ |
| 10 | /decks/{id}/lessons/{id} | Phần học một bài học của một bộ thẻ |
| 11 | /search?q={key} | Tìm kiếm bộ thẻ theo từ khóa "key" |
| 12 | /logout | Đăng xuất ra khỏi hệ thống |

Service (Dịch vụ)

Các dịch vụ trong ứng dụng đóng gói các chức năng đảm bảo việc kết nối với ruby on rails để trao đổi dữ liệu. Dữ liệu trả về theo phuơng thức Promise hoặc Observable để tránh việc không đồng bộ.

Promise để xử lý kết quả của một hành động cụ thể, kết quả của hành động sẽ là thành công hoặc thất bại. Tại một thời điểm, một promise sẽ có 3 trạng thái: chờ hoặc từ chối (pending), thành công (fulfilled) và thất bại (rejected). Trong ứng dụng này tôi sử dụng promise với các phương thức get và post dữ liệu để lấy các dũe liệu về các bộ thẻ và dữ liệu của người dùng.

Observables hỗ trợ cung cấp truyền thông tin giữa bên cung cấp và bên lắng nghe, nó giải quyết việc không đồng bộ và xử lý nhiều giá trị. Observable là một khai báo, bạn định nghĩa một hàm tạo các giá trị nhưng nó sẽ không thực thi cho đến khi phía lắng nghe không đăng ký nó. Observables cũng gần tương tự như Promise. Nhưng Observable có khả năng xử lý theo thời gian, phù hợp với các dữ liệu thời gian thực các sự kiện và các loại sự kiện theo dòng. Với những đặc điểm này, tôi đã áp dụng nó trong việc lấy dữ liệu về người học và dữ liệu các bài học.

Các dịch vụ tôi đã sử dụng:

* DeckService: Lấy dữ liệu các bộ thẻ, các thẻ.
* UserService: Lấy dữ liệu của người dùng
* LessonPracticeService: Lấy dữ liệu của các bài học của người dùng
* AuthenticatesService: Lấy dữ liệu để cử lý đăng nhập đăng ký và lưu phiên đăng nhập của người dùng

Component

Angular 2 có cấu trúc hướng thành phần với các thành phần trong ứng dụng. Ứng dụng của tôi có các component sau:

* HomeComponent: Hiển thị giao diện trang chủ với các thanh tiêu đề để người dùng dễ dàng tuơng tác.
* LoginCompnent: Quản lý người dùng có thể đăng nhập
* SignupComponent: Quản lý người dùng có thể đăng ký
* UserComponent: Quản lý trang thông tin và trang cá nhân của người dùng
* CreateDeckComponent: Tạo ra bộ thẻ mới
* DeckDetailsComponent: Hiện thị dữ liệu của một bộ thẻ
* EditDeckComponent: Chỉnh sửa một bộ thẻ
* SearchDeckComponent: Tìm kiếm bộ thẻ theo từ khóa
* LessonComponent: Quản lý các bài học theo từng bộ thẻ của người dùng.
* ImportDeckComponent: Nhập dữ liệu từ google sheet để tạo các bộ thẻ tự động
  + 1. Xây dựng backend với ruby on rails

Máy chủ sử dụng ruby để kết nối phía client với MySQL. Với ruby tôi đã sử dụng Ruby On Rails là framework để xây dựng các router, và các mô hình dữ liệu theo đúng chuẩn, sau đây là bảng mô tả API của hệ thống.

Bảng 4.23: Bảng mô tả API

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **API** | **Phương thức** | **Giải thích** |
| 1 | /api/auth/users | POST | Thêm người dùng mới vào cơ sở dữ liệu |
| 2 | /api/auth/sign\_in | POST | Đăng nhập vào hệ thống |
| 3 | /api/auth/sign\_out | POST | Đăng xuất ra khỏi hệ thống |
| 4 | /api/decks/new | GET | Trả về trang tạo mới bộ thẻ |
| 5 | /api/decks | POST | Thêm một bộ thẻ mới vào cơ sở dữ liệu |
| 6 | /api/decks | GET | Lấy tất cả bộ thẻ của hệ thống |
| 7 | /api/decks/:id | GET | Trả về một bộ thẻ cụ thể |
| 8 | /api/decks/:id | PATCH | Chỉnh sửa một bộ thẻ |
| 9 | /api/decks/:id/edit | GET | Trả về một bộ thẻ |
| 10 | /api/user\_lessons/:id | POST | Tạo một bài học theo từng người dùng |
| 11 | /api/user\_lessons | DELETE | Xóa một bài học của người dùng |
| 12 | /api/favorite/decks | POST | Thêm một bộ thẻ yêu thích cho người dùng |
| 13 | /api/favorite/decks/:id | DELETE | Xóa bộ thẻ yêu thích của người dùng |
| 14 | /api/following\_people/decks | POST | Người dùng theo dõi một người dùng khác |
| 15 | /api/following\_people/decks/:id | DELETE | Người dùng bỏ theo dõi một người dùng khác |
| 16 | /api/trending/decks | GET | Trả về những bộ thẻ có số lượng yêu thích nhiều |
| 17 | /api/search/decks | GET | Trả về những bộ thẻ theo chủ để hoặc mô tả mà người dùng tìm kiếm |
| 18 | /api/lessons/:id | GET | Trả về một bài học của một bộ thẻ |

**Nhập dữ liệu từ danh sách có sẵn ở google sheet**

Tôi đã sử dụng thư viện google\_drive để đọc ghi một tệp tin hay một bảng tính được lưu trên Google Drive hay Google Docs cho ứng dụng của mình, cụ thể là đọc các hàng dữ liệu ở trong google sheet để tự động tạo ra các bộ thẻ khớp với

Bất kỳ một ứng dụng Rails nào muốn truy cập tới Google Drive để sử dụng các chức năng đọc/ghi tệp tin..., thì trước hết chúng ta cần phải Authorization cho những tài khoản người sử dụng ứng dụng đó.

**Xuất dữ liệu của người dùng và bộ thẻ ra định dạng csv hoặc excel**

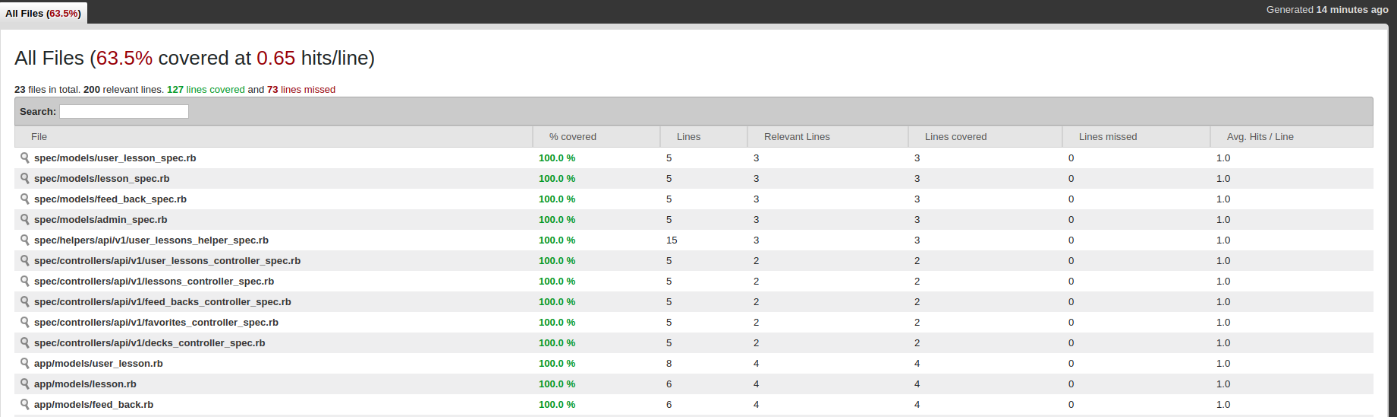
Sử dụng thư viện csv để tạo ra dữ liệu csv, khi xuất dữ liệu có thể xuất từng hàng hoặc xuất cả bảng tùy thuộc vào cấu hình, ở khóa luận này tôi đã cấu hình để toàn bộ dữ liệu được xuất ra.

Chức năng phản hồi của người dùng:

Sử dụng websocket qua action cable để xây dựng ứng dụng thười gian thực. Đầu tiên, websockets là giao thức được xây dựng dựa trên TCP protocol. Nó sẽ thiết lập kết nối giữa máy khách và máy chủ, duy trì kết nối đó liên tục để khi có người dùng phản hồi ngay lập tức quản trị viên sẽ nhận được thông báo. ActionCable là phần trung gian giữa máy chủ mà máy khách, nó giúp thiết lập và quản lý kết nối hiệu quả hơn.

* 1. Kiểm thử tự động

Trong qúa trình xây dựng và phát triển phần mềm, kiểm thử có một ý nghĩa và vai trò vô cùng quan trọng. Môi trường phát triển hiện nay đang hướng tới: "Quá trình phát triển ngắn", "kinh phí thấp", "chất lượng cao". Hiện nay, quá trình test đã chiểm đến gần 40% thời gian phát triển phần mềm Nhằm khắc phục thực trạng này, kiểm thử tự động càng được chú ý đến, đặc biệt là phương pháp kiểm thử "unit test". Tôi đã sử dụng Rspec để kiểm thử các chức năng của ứng dụng của tôi dựa theo mô hình BDD (Behavior Driven Development) để làm quy trình phát triển phần mềm. BDD được hiểu đơn giản là quá trình vòng lặp với 3 quá trình: viết kiểm thử lỗi bởi vì lúc này viết mã kiểm thử nhưng chưa có mã nguồn nên sẽ bị lỗi, sửa mã nguồn để kiểm thử thành công và chỉnh sửa lại mã nguồn để tối ưu hóa và loại bỏ mã dư thừa. Trong quá trình phát triển phần mềm, tôi nhận thấy càng đầu tư nhiều thời gian để viết kiểm thử thì việc chỉnh sửa mã nguồn và quản lí toàn bộ dự án càng trở nên dễ dàng, giảm gánh nặng khi cần chỉnh sửa code, đồng thời giam thiểu được lỗi của ứng dụng. Rspec là một công cụ kiểm thử dành cho Ruby, tuy nó rất mạnh nhưng lại có thể sử dụng đơn giản và hiệu quả. Tôi đã áp dụng gem simplecov (một thư viện chuyên dùng để đánh giá chất lượng kiểm thử trong dự án) để đánh giá hiệu quả của việc kiểm thử ứng dụng của tôi. Nó thống kê số lần mỗi dòng của mã ứng dụng được chạy trong quá trình viết kiểm thử. Kết quả đạt được thể hiện trong hình dưới đây.



Hình 4.6: Kết quả chạy kiểm thử tự động

# Thực nghiệm

Chương 5 mô tả môi trường thực nghiệm, dữ liệu thực nghiệm và kết quả thực nghiệm trên các môi trường cùng với sự phản hồi của những người dùng thử ứng dụng.

* 1. Môi trường thực nghiệm

Môi trường thử nghiệm bao gồm như sau:

* Laptop hệ điều hành ubuntu với các trình duyệt chrome, firefox.
* Oppo Neo 7
* Ipad mini
  1. Dữ liệu thực nghiệm

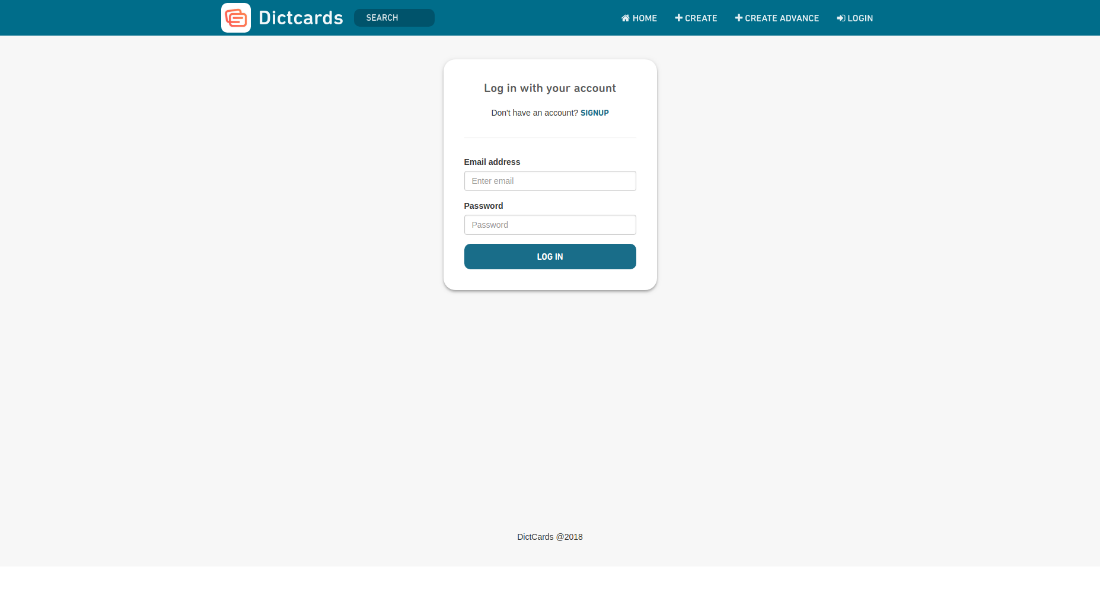
Dữ liệu dùng để thực nghiệm được lấy từ google sheets như hình 5.1



Hình 5.7: Dữ liệu dưới dạng excel của google sheets

* 1. Kết quả thực nghiệm
     1. Xác thực và ủy quyền ngừoi dùng

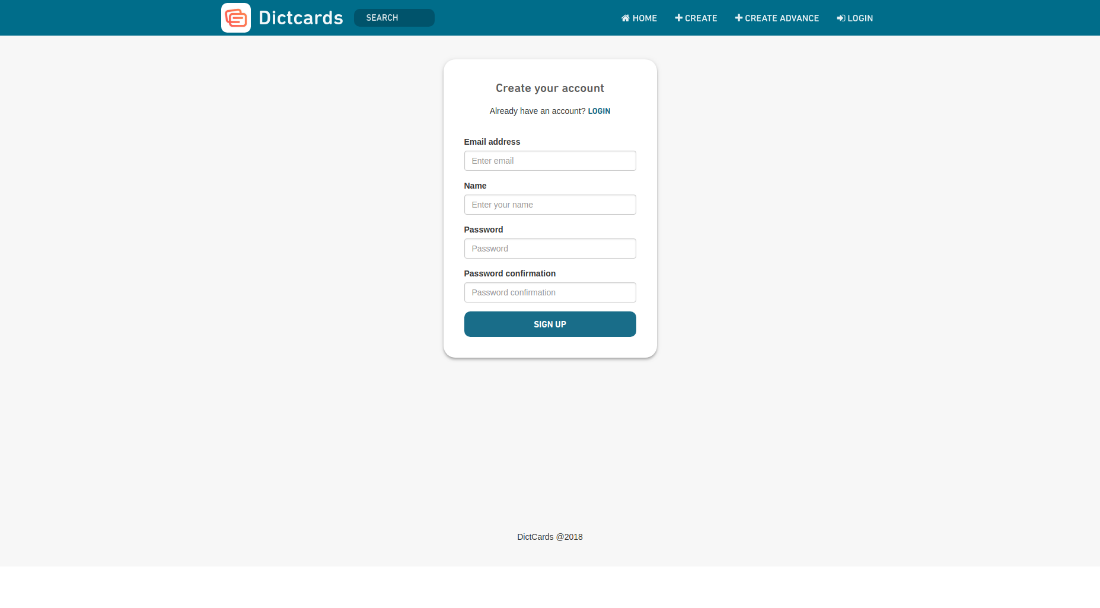
Khi một người dùng là khách vãng lai đăng nhập vào hệ thống thì một giao diện như thế này sẽ hiện ra:



Hình 5.8: Giao diện khi chưa đăng nhập

Lúc này hệ thống cần bạn đăng nhập hoặc đăng ký để có thể truy cập và sử dụng các tính năng của hệ thống.

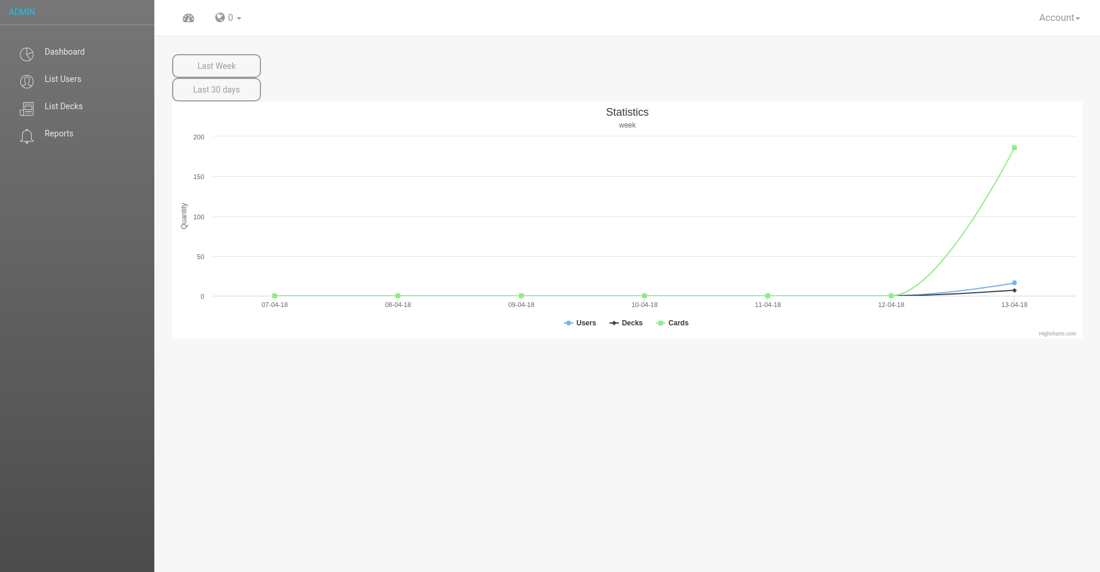
Nếu bạn chưa có tài khoản trong hệ thống, bạn sẽ nhấn vào đăng ký để tạo tài khoản. Giao diện đăng ký tài khoản như sau:



Hình 5.9: Giao diện đăng ký tài khoản của ngừoi dùng

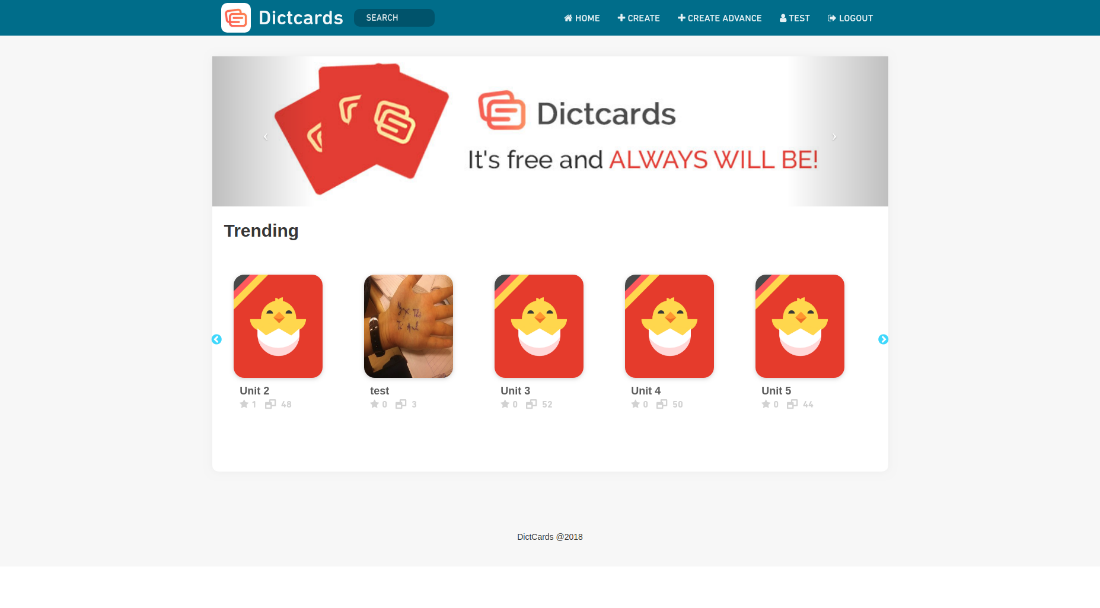
Hệ thống sửa dụng chung một form đăng nhập đăng ký cho tất cả ngừoi dùng, bao gồm cả tài khoản quản trị viên và người dùng. Với mỗi người dùng khác nhau hệ thống sẽ có cơ chế tự nhận biết và xác thực khác nhau thông qua đường định tuyến routes. Chẳng hạn, với tài quản quản trị viên, sau khi đăng nhập hệ thống sẽ chuyển hướng tới khu vực quản trị. Khu vực này người dùng không có quyền truy cập.

Giao diện sau khi đăng nhập dành cho quản trị viên



Hình 5.10: Giao diện trang chủ dành cho quản trị viên

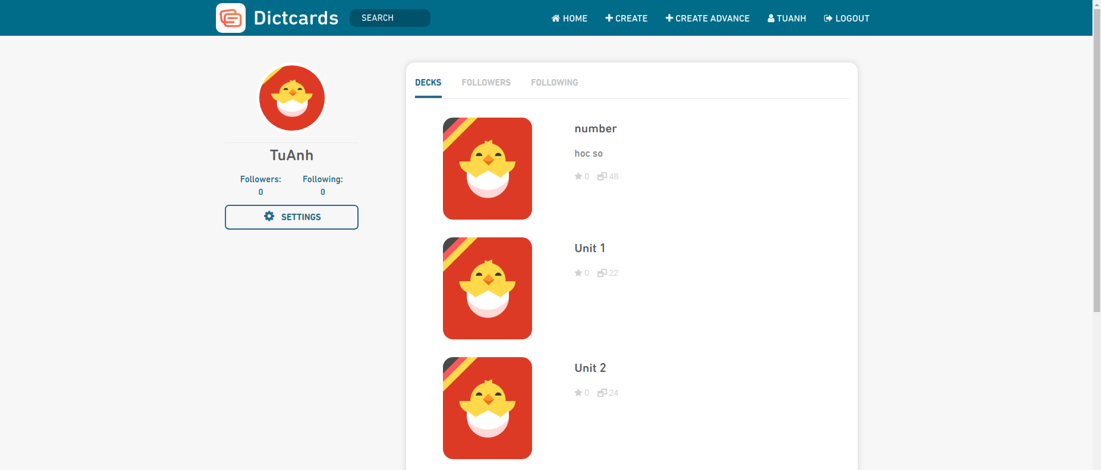
Giao diện sau khi đăng nhập của người dùng:



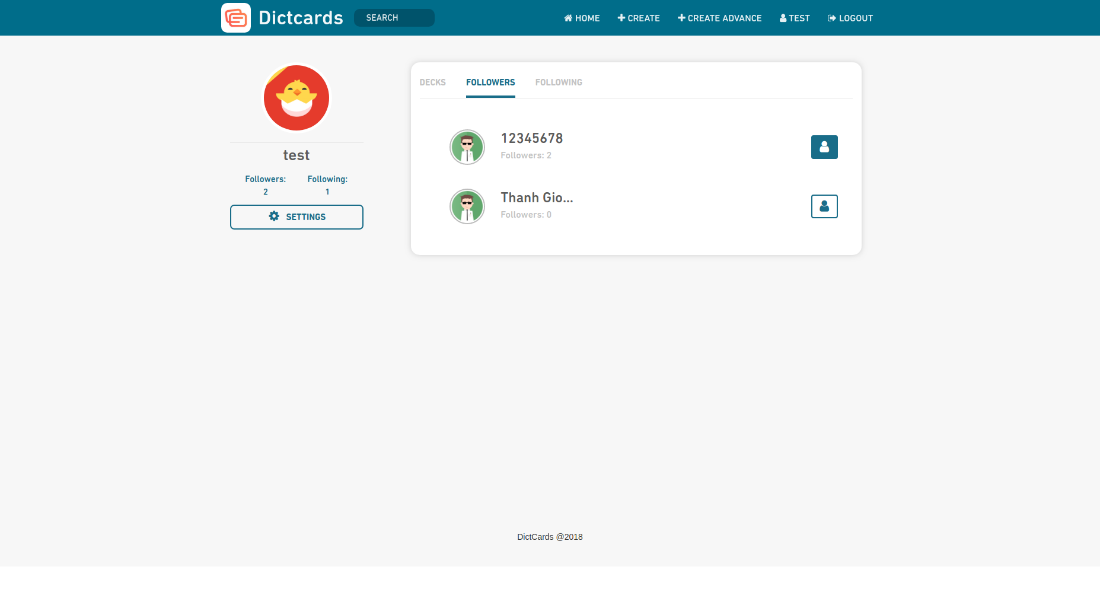
Hình 5.11: Giao diện trang chủ dành cho người dùng

* + 1. Dành cho người dùng

Sau khi đăng nhập, người dùng có thể chọn thông tin cá nhân để xem các thông tin cơ bản của mình trên hệ thống. Các thông tin về những bộ thẻ của người dùng, những người đang theo dõi và những người đang theo dõi. Tại đây với danh sách những người đang theo dõi, người dùng có thể hủy theo dõi bất cứ ai.

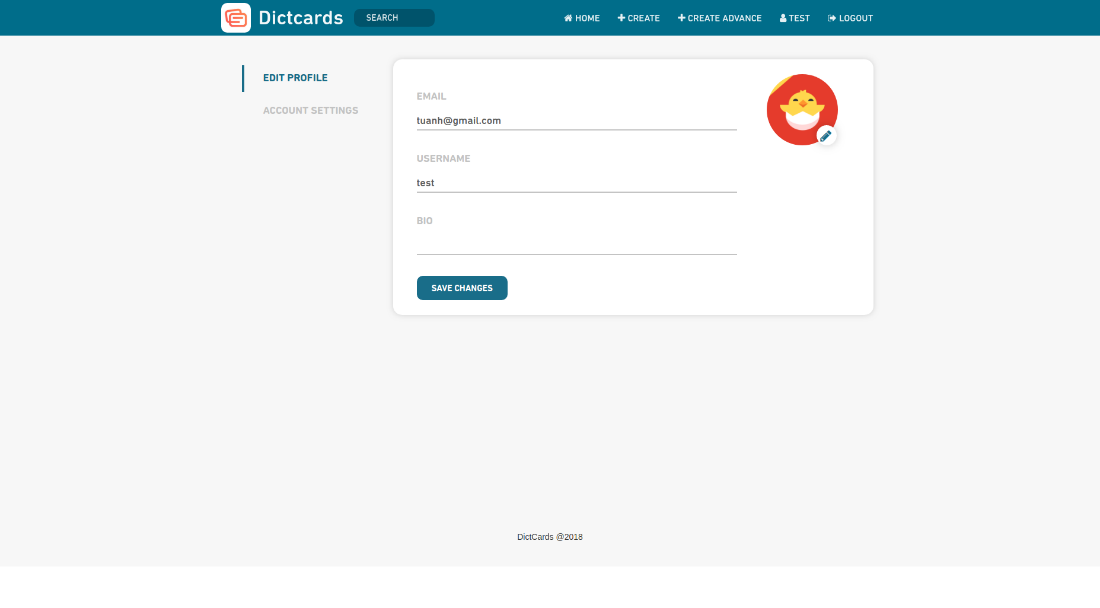


Hình 5.12: Giao diện trang cá nhân của người dùng



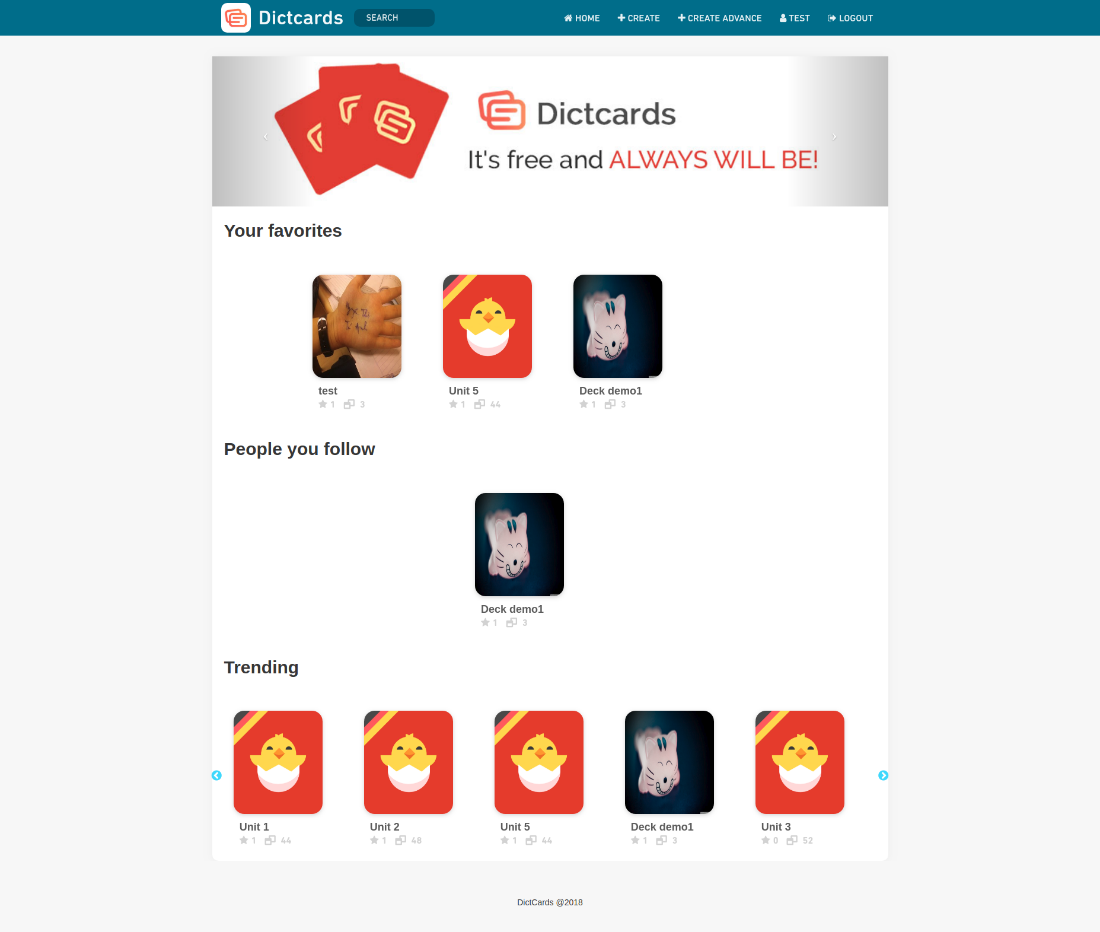
Hình 5.13: Giao diện trang cá nhân của người dùng

Trong trang cũng có nút cài đặt (Settings), nếu người dùng bấm vào nút này giao diện sẽ chuyển đến giao diện chỉnh sửa thông tin cá nhân về emai, tên, mô tả bản thân và thay đổi mật khẩu tài khoản, giao diện chuyển trang như sau



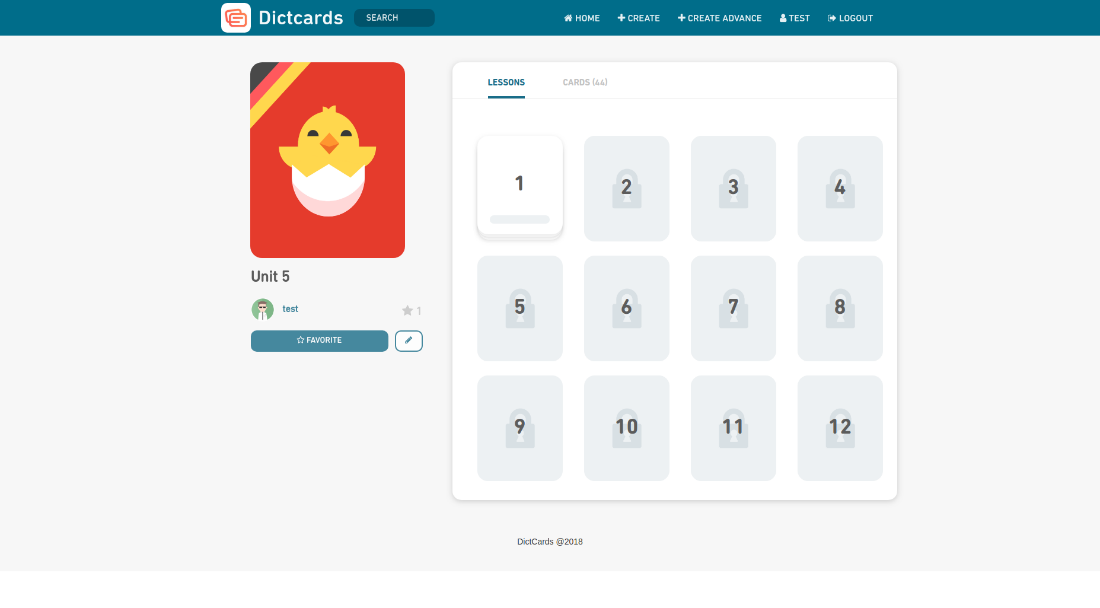
Hình 5.14: Giao diện xem và chỉnh sửa thông tin người dùng

Giao diện trang chủ sau khi đăng nhập gồm một thanh công cụ sẽ hiển thị các bộ thẻ để người dùng dễ dàng tuơng tác. Đầu tiên sẽ là danh sách những bộ thẻ mà người dùng yêu thích và thấy có ích với mình, tiếp theo là danh sách những bộ thẻ của người khác mà người dùng đã theo dõi, cuối cùng là danh sách những bộ thẻ có số lượt yêu thích nhiều nhất.



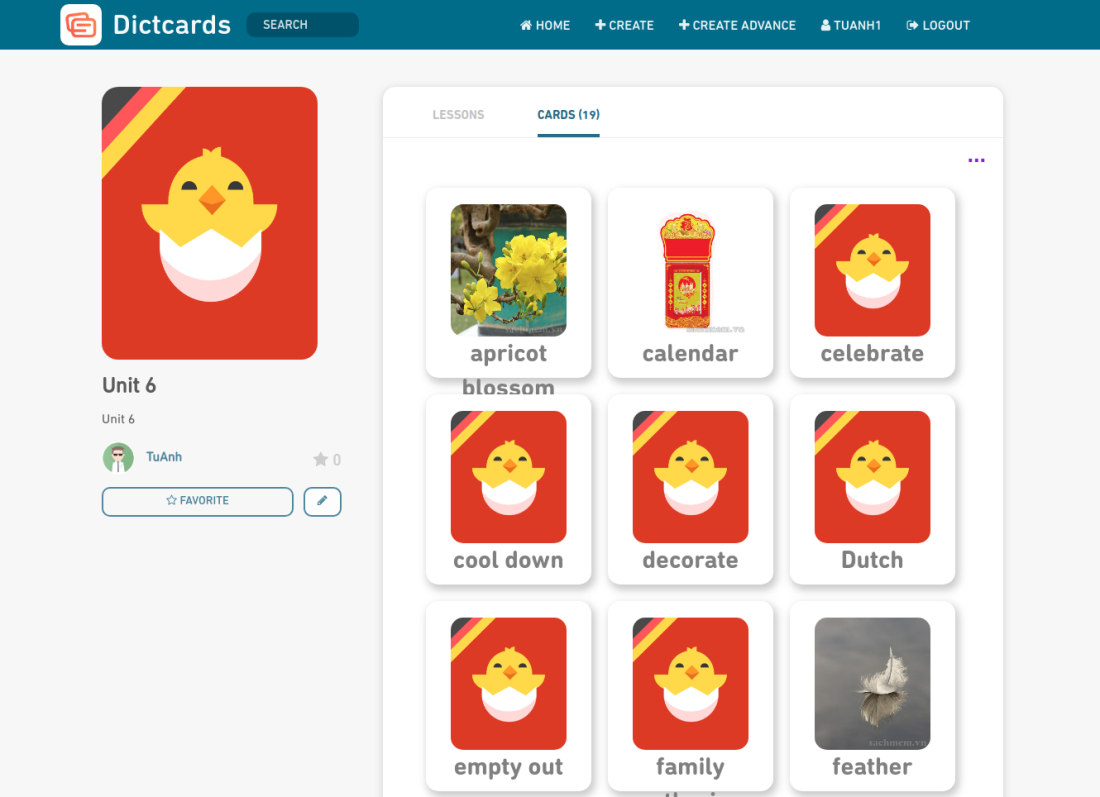
Hình 5.15: Giao diện trang chủ của người dùng

Khi chọn vào một bộ thẻ, hệ thống sẽ chuyển sang giao diện của bộ thẻ đó như hình:



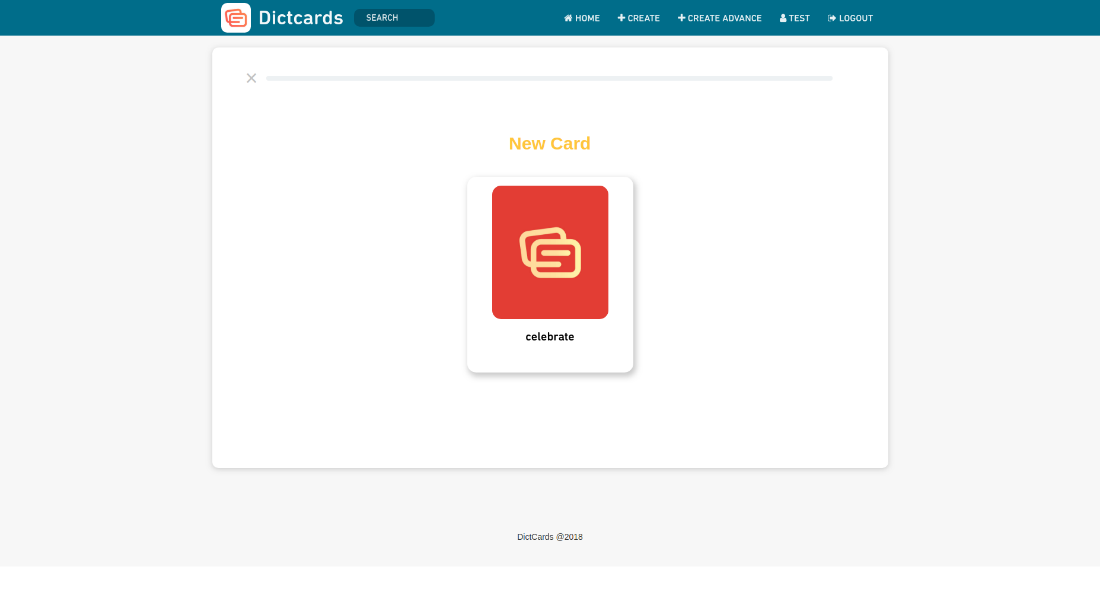
Hình 5.16: Giao diện của một bộ thẻ với các bài học

Bộ thẻ sẽ có những bài học, và bài học nào chưa có quyền truy cập sẽ bị khóa lại nếu bài học trước đó chưa học xong. Một bộ thẻ có nhiều thẻ nên sẽ có một giao diện hiển thị tất cả các thẻ của bộ thẻ đó như hình:



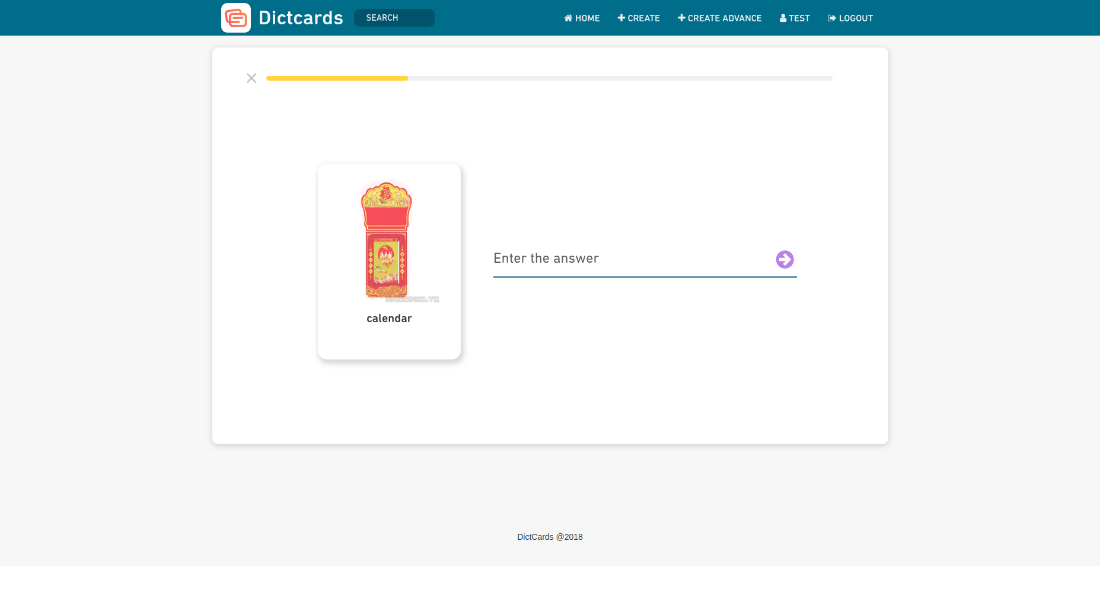
Hình 5.17: Giao diện của một bộ thẻ với các thẻ

Khi người dùng nhấp vào một bài học thì bên phía server sẽ tạo câu hỏi và trả về câu hỏi để phía client hiển thị, giao diện của bài học đó sẽ được hiển thị như hình... Khi trả lời đúng thanh ghi điểm sẽ tăng lên, điểm Có 3 dạng câu hỏi. Đầu tiên là dạng câu hỏi để học từ, nó sẽ hiển thị mặt trước và mặt sau khi nhấp vào thẻ, khi người dùng nhớ thẻ này thì người cũng sẽ nhấp tiếp tục sang câu hỏi khác.



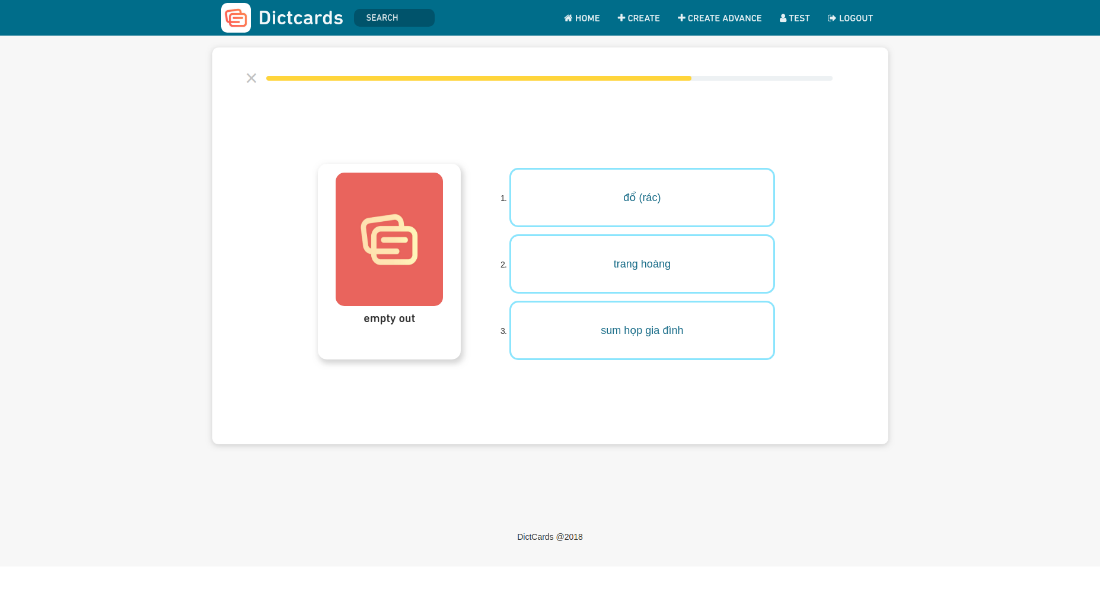
Hình 5.18: Giao diện bài học với Dạng câu hỏi học từ

Thứ hai là dang câu hỏi viết, người dùng sẽ nhớ xem với mạt của thẻ đựoc hiển thị thì mạt tuơng ứng của nó sẽ là gì. Nếu người dùng trả lời đúng hệ thống sẽ chuyển sang câu hỏi khác. Nếu người dùng trả lời không đúng, hệ thống sẽ hiển thị câu trả lời đúng. Và chuyển sang câu hỏi khác



Hình 5.19: Giao diện bài học với Dạng câu hỏi viết

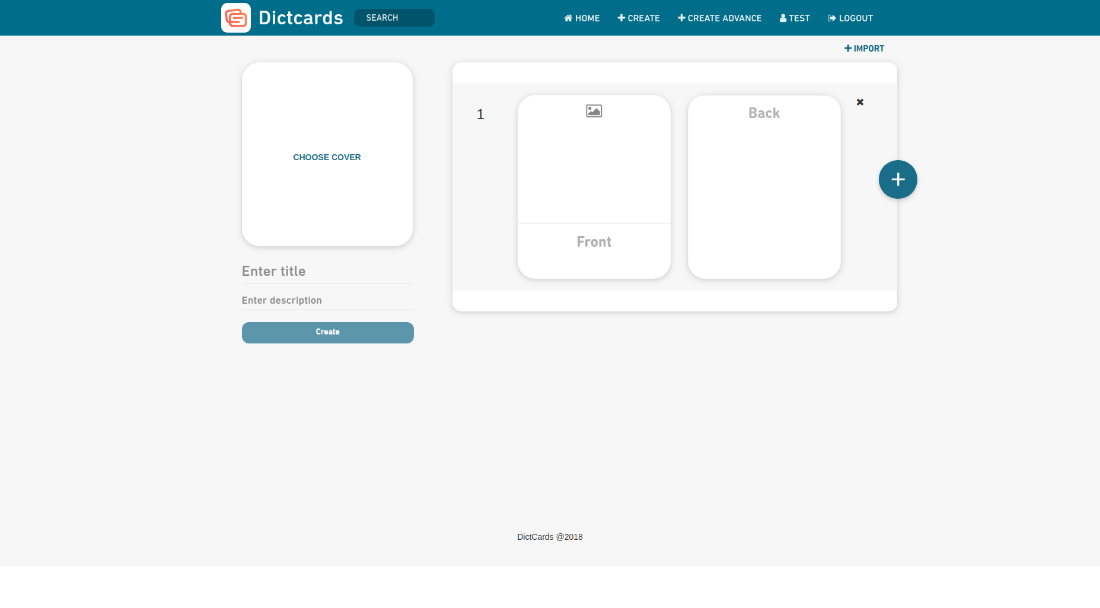
Thứ 3 là dạng câu hỏi chọn thẻ để nhớ các mặt. Hệ thống sẽ hiển thị một mặt của 1 thẻ và có 3 đáp án để người dùng chọn một trong số đó khớp với mặt kia.



Hình 5.20: Giao diện bài học với Dạng câu hỏi nhớ các mặt

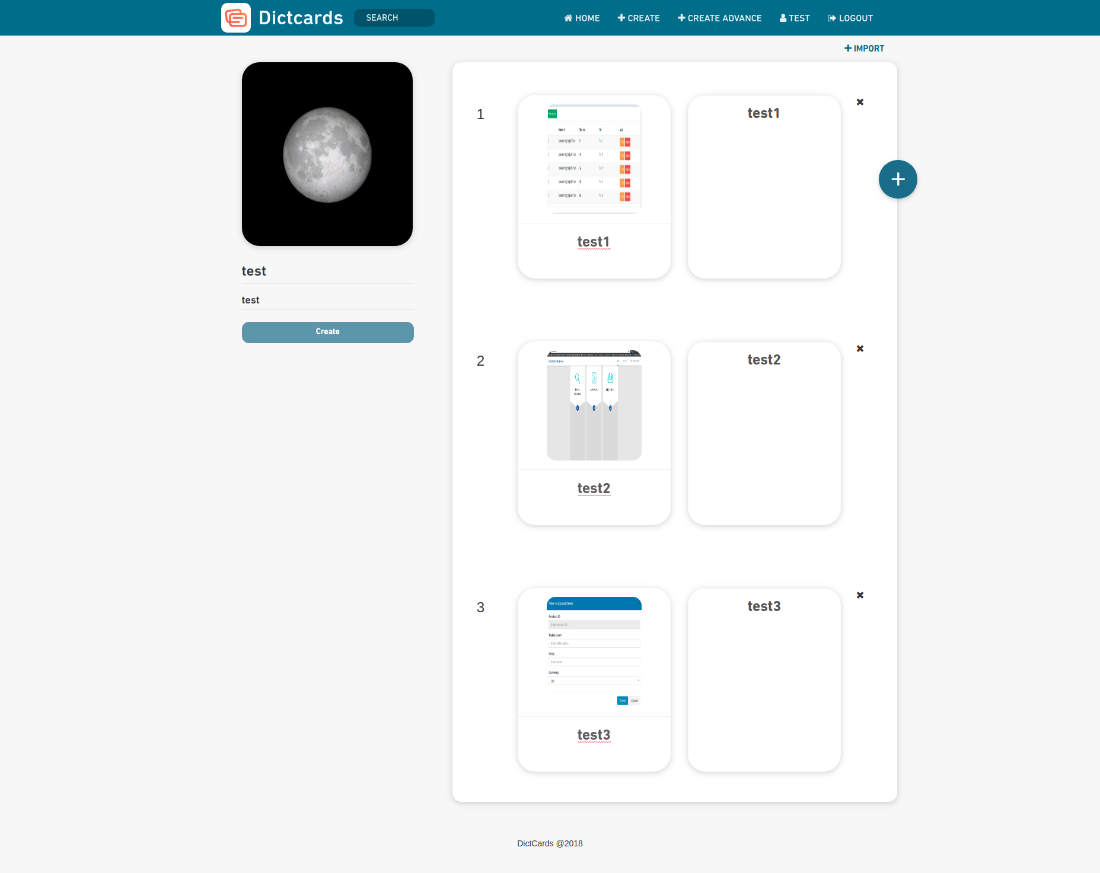
Tạo một bộ thẻ mới

Trên thanh công cụ người dùng bấm vào "Create", hệ thống sẽ chuyển sang giao diện tạo bộ thẻ như hình..



Hình 5.21: Giao diện Tạo bộ thẻ

Người dùng sẽ điền đầy đủ thông tin của bộ thẻ và thêm dữ liệu của các thẻ. Sau đó bấm vào "Create".

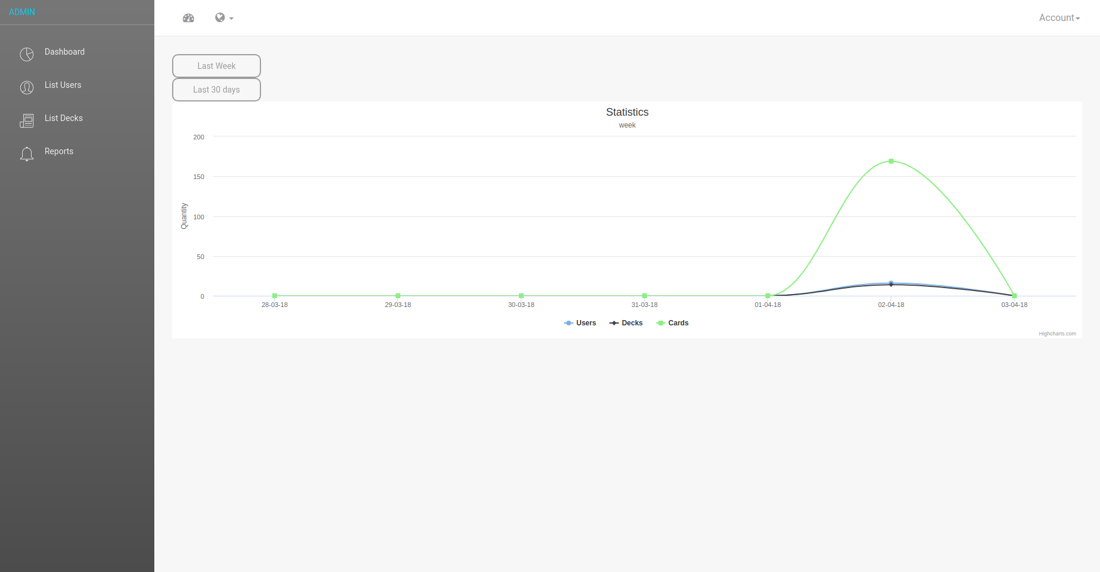


Hình 5.22: Giao diện Nhập dữ liệu màn tạo bộ thẻ

Sau khi tạo hệ thống sẽ chuyển đến trang của bộ thẻ vừa mới tạo và thông báo thành công.

* + 1. Khu vực quản lý của quản trị viên

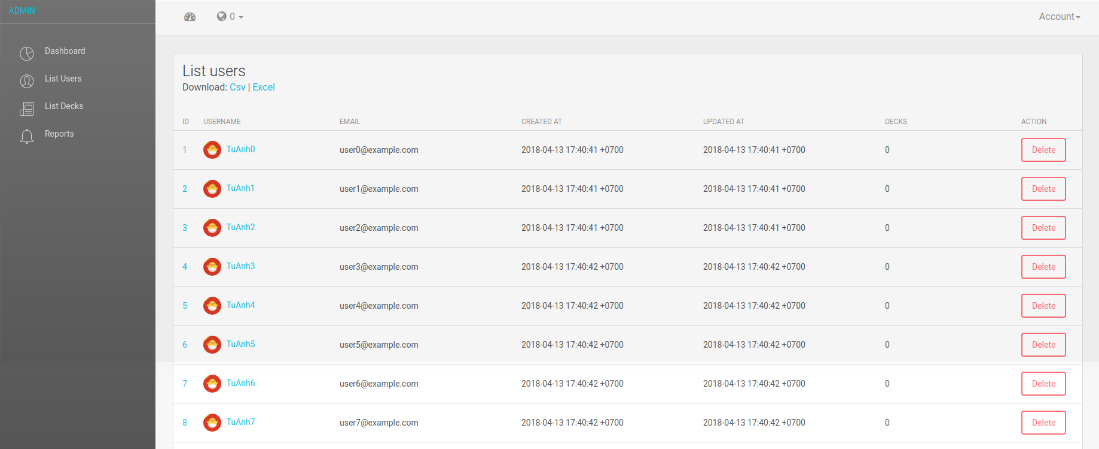
Sau khi người quản trị đăng nhập hệ thống hệ thống sẽ chuyển đến trang biểu đồ của hệ thống



Hình 5.23: Giao diện sau khi đăng nhập của quản trị viên

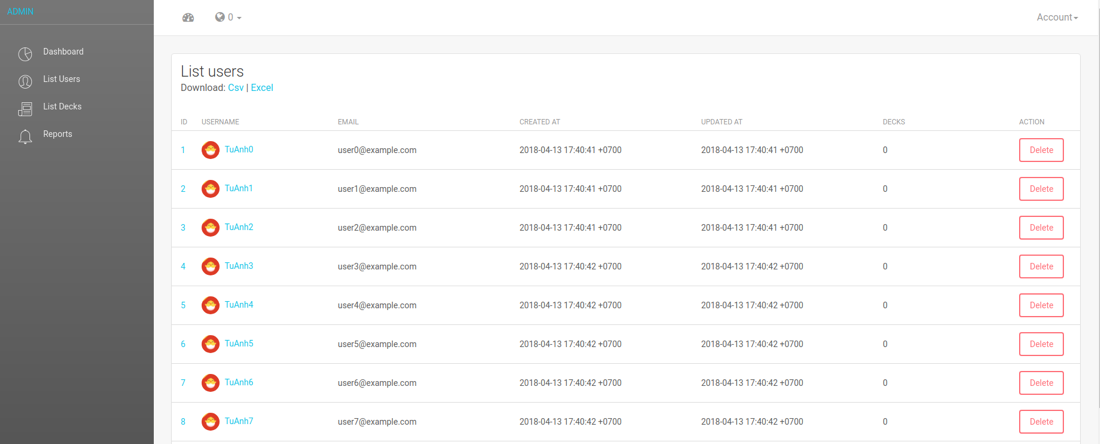
Dữ liệu về người dùng, số bộ thẻ, số thẻ sẽ đựoc thống kê theo ngày và hiện thị trong một tuần hoặc một tháng.

Trên thanh dọc công cụ sẽ có các mục để người quản trị nhấp vào. Về người dùng, sẽ hiển thị tất cả ngừoi dùng với đây đủ các thống tin về tên, email, số lựong bộ thẻ.



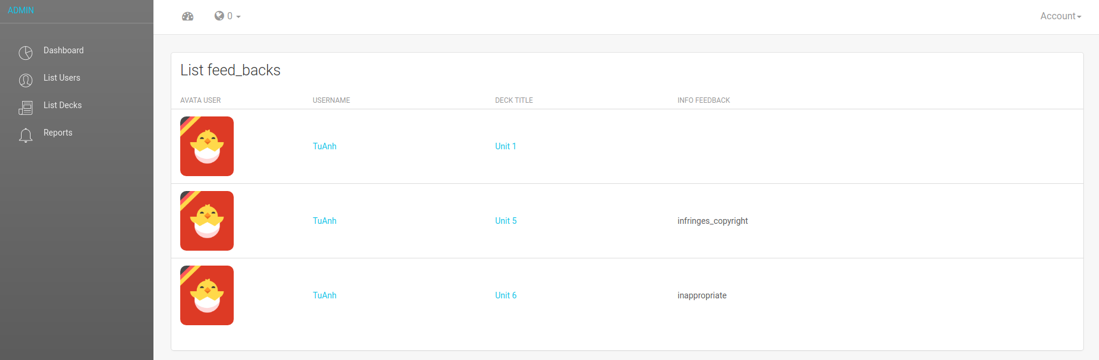
Hình 5.24: Giao diện tất cả người dùng

Về bộ thẻ, sẽ hiện thị toàn bộ thẻ với thông tin về ảnh của bộ thẻ, tiêu đề và số lượng thẻ cũng nhưng người đã tạo ra bộ thẻ đó.



Hình 5.25: Giao diện tất cả bộ thẻ

Về phản hồi bộ thẻ của người dùng, ở thanh công cụ nếu người quản trị bấm vào danh sách phản hồi (Reports) sẽ hiển thị người tạo phản hồi và nội dung phản hồi và bộ thẻ được phản hồi đề người quản lý kiểm tra để xem xét có xóa bộ thẻ hay không.



Hình 5.26: Giao diện tất cả phản hồi

* 1. Nhận xét, đánh giá ứng dụng

Hiện tại, ứng dụng cơ bản đã hoàn thành. Chức năng và giao diện được kiểm thử thông qua một số ngừoi dùng và đã nhận được các nhận xét, phần hồi khách quan. Đa số những người dùng thủ đều hài lòng với chức năng và giao diện của ứng dụng. Kết quả ở bảng sau:

Bảng 5.24: Phản hồi của người dùng

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Nghề nghiệp** | **Bạn thấy ứng dụng có hữu ích không?** | **Bạn thấy ứng dụng có dễ dàng sử dụng không?** | **Bạn thấy ứng dụng có chạy ổn định không?** | **Giao diện của ứng dụng như thế nào?** | **Ý kiến đóng góp của bạn** |
| 1 | Nguyễn Anh Tú | Lập trình viên | Có | Có | Ứng dụng chạy mượt mà, Tốc độ load trang nhanh | Đẹp, hiệu ứng tốt | Chưa có nhiều người dùng nên số lượng thẻ ít, chưa phong phú nội dung |
| 2 | Võ Mỹ Phương | Học sinh | Có | Có | Có | Mình thích những ứng dụng có giao diện đơn giản như thế này | Bổ sung tính năng thiết lập mục tiêu hàng ngày, nhắc nhở người dùng khi chưa đạt mục tiêu, hiển thị tiến độ so với mục tiêu đã đề ra. Ngoài ra nếu có thêm chức năng điểm số, xem được điểm số của bạn bè sẽ tạo ra tính ganh đua hiệu quả hơn. |
| 3 | Nguyễn Thị Hồng Lam | Sinh viên | Có | Có | Có | Giao diện đẹp, dễ dùng |  |
| 4 | Vũ Thanh Tùng | Lập trình viên | Có | Có | Có, chạy mượt, không bị giật | Giao diện đơn giản, đẹp | Ứng dụng tốt, dễ sử dụng, thích hợp với mọi lứa tuổi |
| 5 | Nguyễn Thị Hải | Giáo viên | Có | Có | Có | Khá đẹp | Ứng dụng còn ít dữ liệu |
| 6 | Trần Đức Anh | Sinh viên | Có | Có | Có | Đơn giản, dễ sử dụng |  |

# Kết luận

* 1. Kết quả đạt được

Xây dựng ứng dụng công đồng học nhanh qua thẻ là một ứng dụng áp dụng ý tưởng từ một ứng dụng có sẵn và có phần sáng tạo bổ sung thêm các chức năng mới giúp người dùng dễ sử dụng và thuận tiện hơn trong việc học và ghi nhớ các khái niệm. Việc sử dụng Angular giúp cho hệ thống có tốc độ tải nhanh và thân thiện với ngừoi dùng. Đa số ngừoi dùng thử nghiệm đều hài lòng với giao diện và chức năng của ứng dụng. Vì ứng dụng mở cho tất cả mọi người dùng có thể tạo bộ thẻ nên mọi người có thể cùng học và chia sẻ những kiến thức bổ ích với cộng đồng. Cách tạo ra các bài học để luyện kiến thức có khoa học giúp người dùng dễ nhớ và cảm thấy hấp dẫn thú vị hơn khi học. Bên cạnh các ưu điểm thì ứng dụng cũng có một số nhược điểm như các dạng câu hỏi chưa đa dạng để ôn luyện, số lượng người dùng chưa nhiều nên số lượng bộ thẻ còn ít.

Trong quá trình thực hiện khóa luận này, tôi đã tích lũy, học hỏi được những kiến thức về các công nghệ Ruby on Rails, Angular 2... Bên cạnh đó việc làm khóa luận cũng giúp tôi hiểu được quy trình phát triển phần mềm, các cách thức làm việc nhóm để giúp tôi tích lũy thêm nhiều kinh nghiệp phục vụ có công việc sau này.

* 1. Định hướng phát triển

Trong thời gian tới ứng dụng sẽ được phát triển theo các hướng sau:

* Đa dạng loại hình của thẻ như thẻ động, có nhiều mặt hơn.
* Xây dựng biểu đồ học tập cho người dùng để người dùng nắm được qua trình học của bản thân.
* Cấu trúc ứng dụng thân thiện hơn với người dùng.

TÀI LIỆU THAM KHẢO

**Tiếng Anh**

[1] "Angular 2," Google, [Online]. Available: https://angular.io/. [Accessed 12 February 2018].

[2] "Ruby on Rails," Google, [Online]. Available: rubyonrails.org. [Accessed 12 February 2018].

[3] “Single page application,” Google, [Online]. Available: https://en.wikipedia.org/wiki/Single-page\_application. [Accessed 12 February 2018].

[4] "Tinycards," [Online], Available: https://tinycards.duolingo.com/. [Accessed 12 February 2018]

[5] "Angular 2 - Service" Google, [Online]. Available: https://www.tutorialspoint.com/angular2/angular2\_services.html. [Accessed 12 February 2018]

**Tiếng Việt**

[6] "Ứng dụng web," Google, [Online]. Available: <https://vi.wikipedia.org/wiki/Ứng_dụng_web>. [Truy cập lần cuối 12 tháng 2 năm 2018]

[7] "Học lập trình web bằng ruby on rails," Google, [Online]. Available: <https://techmaster.vn/posts/7951/hoc-lap-trinh-web-bang-ruby-on-rails-2>. [Truy cập lần cuối 12 tháng 2 năm 2018]

[8] "[A2] Giới thiệu về Angular 2," Google, [Online]. Available: https://github.com/truonganhhoang/int3507-2016/wiki/%5BA2%5D-Giới-thiệu-về-Angular-2. [Truy cập lần cuối 12 tháng 2 năm 2018]

[9] "Hệ quản trị cơ sở dữ liệu," Google, [Online]. Available: https://vi.wikipedia.org/wiki/Hệ\_quản\_trị\_cơ\_sở\_dữ\_liệu. [Truy cập lần cuối 12 tháng 2 năm 2018]

[10] "ActionCable trong Rails 5," Google, [Online]. Available:

https://viblo.asia/p/actioncable-trong-rails-5-nwmkyZbbvoW. [Truy cập lần cuối 12 tháng 2 năm 2018]

[11] "Tìm hiểu về kiểm thử tự động," Google, [Online]. Available: https://viblo.asia/p/tim-hieu-ve-kiem-thu-tu-dong-dich-1l0rvmxDGyqA. [Truy cập lần cuối 12 tháng 2 năm 2018]