Mục lục

[**I.** **Các công nghệ áp dụng** 2](#_Toc514280138)

[**1.** **Angular 5** 2](#_Toc514280139)

[1.1 Giới thiệu chung 2](#_Toc514280140)

[1.2 Các đặc điểm chính 2](#_Toc514280141)

[**2.** **Nodejs** 2](#_Toc514280142)

[2.1 Giới thiệu 2](#_Toc514280143)

[2.2 Framework Express 2](#_Toc514280144)

[**3.** **Bootstrap** 2](#_Toc514280145)

[**4.** **MongoDB** 2](#_Toc514280146)

[**II.** **Phân tích yêu cầu** 2](#_Toc514280147)

[**1.** **Đặc tả yêu cầu chức năng** 2](#_Toc514280148)

[**2.** **Đặc tả yêu cầu phi chức năng** 2](#_Toc514280149)

[**3.** **Tóm tắt yêu cầu từ người sử dụng** 3](#_Toc514280150)

[**4.** **Thiết kế ca sử dụng** 3](#_Toc514280151)

[**5.** **Phân tích ca sử dụng** 4](#_Toc514280152)

[**III.** **Mô tả cài đặt ứng dụng** 6](#_Toc514280153)

[**1.** **Cài đặt Frontend** 6](#_Toc514280154)

[***a.*** ***Thư mục homepage*** 6](#_Toc514280155)

[***b.*** ***Thư mục card*** 6](#_Toc514280156)

[***c.*** ***Thư mục create-card*** 6](#_Toc514280157)

[***d.*** ***Thư mục navigation-bar*** 6](#_Toc514280158)

[***e.*** ***Thư mục learnpage*** 6](#_Toc514280159)

[***f.*** ***Thư mục loginpage*** 6](#_Toc514280160)

[***g.*** ***File cookiee.ts*** 6](#_Toc514280161)

[***h.*** ***File app-routing.module.ts*** 6](#_Toc514280162)

[***i.*** ***File user.guard.ts*** 6](#_Toc514280163)

[***j.*** ***File data.service.ts*** 6](#_Toc514280164)

[**2.** **Cài đặt Backend** 6](#_Toc514280165)

[**3.** **Cài đặt cơ sở dữ liệu** 6](#_Toc514280166)

[***a.*** ***Các collection*** 6](#_Toc514280167)

[**b.** **Document mẫu cho các collection** 6](#_Toc514280168)

# **Các công nghệ áp dụng**

## **Angular 5**

### 1.1 Giới thiệu chung

### 1.2 Các đặc điểm chính

Ứng dụng Angular được xây dựng trên nền tảng module, được gọi là Angular

module hay NgModules.

Một Angular module là một lớp dữ liệu kiểu NgModule. Siêu dữ liệu thể hiện những công việc sau:

* Khai báo component, directive và pipe thuộc về module.
* Đưa ra một số trạng thái công khai để các template của các component khác có thể sử dụng chúng.
* Khai báo các module khác cần cho chính module này.
* cung cấp các service ở mức ứng dụng.

*Component*

Một component điều khiển một phần màn hình gọi là view.

component xác định logic cho một lớp trong ứng dụng. Lớp này tương tác với view thông qua các thuộc tính hay phương thức.

*Template*

Template xác định giao diện cho component dưới dạng HTML.

*Data binding*

Angular hỗ trợ data binding (ràng buộc dữ liệu), cơ chế giúp ràng buộc các thành phần của template với component.

Có bốn dạng cú pháp data binding, mỗi dạng cáo một hướng: tới DOM, từ DOM; hoặc cả hai hướng.

*Pipe*

Pipe có nhiệm vụ nhận dữ liệu đầu vào và biến đổi nó thành dạng đầu ra mong muốn trong một template.

*Attribute directive*

Có ba loại directive trong Angular:

1. Component-directive đi cùng với template.

2. Structural directive- thay đổi giao diện cấu trúc DOM bằng việc thêm bớt các phần tử DOM.

Component là directive thông dụng nhất trong ba loại.

*Router*

Router của Angular cho phép điều hướng từ một view này sang view khác khi người dùng thực hiện thao tác trên ứng dụng.

*Service*

Service là nền tảng cho mọi ứng dụng Angular. Các component sẽ sử dụng service lặp lại rất nhiều lần, tăng khả năng tái sử dụng mã nguồn.

## **Nodejs**

### 2.1 Giới thiệu

Node.js là môi trường chạy Javascript mã nguồn mở thực thi mã JavaScript.

Node.js cho phép tạo các web server, công cụ mạng sử dụng JavaScript và một bộ các thư viện “module” kiểm soát các chức năng cốt lõi

### 2.2 Framework Express

Express.js hay đơn giản là Express, là một framnework cho Node.js. Nó được thiết kế để xây dựng ứng dụng web và các API. Express cùng với Node.js cho phép xây dựng một website hoàn toàn với JavaScript.

## **Bootstrap**

Bootstrap là một framework cho phép thiết kế website responsive nhanh hơn và dễ dàng hơn

Bootstrap bao gồm các HTML templates, CSS templates và Javascript tao ra những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm các plugin Javascript trong nó. Giúp cho việc thiết kế responsive của bạn dễ dàng hơn và nhanh chóng hơn.

Nó tương thích với tất cả các trình duyệt như Chrome, Firefox, Internet Explorer, Safari, Opera...Tương tác tối với smartphone.

## **MongoDB**

MongoDB là hệ quản trị cơ sở dữ liệu mã nguồn mở. Được sắp xếp là hệ quản trị SQL, MongoDB sử dụng dữ liệu dưới dạng các tài liệu tương tự như JSON với lược đồ được định nghĩa trước.

MongoDB hiện nay đa hỗ trợ trên 10 ngôn ngữ lập trình như: Java, C++, Python…

MongoDB lưu trữ dữ liệu - document- bên trong các tập hợp collection.

# **Phân tích yêu cầu**

Tiếng Anh rất quan trọng trong cuộc sống và công việc ngày nay. Nó giúp chúng ta giao tiếp dễ dàng với mọi người trên thế giới và khiến cuộc sống và công việc của chúng ta dễ dàng hơn. Qua đó ta thấy tầm quan trọng của việc học tiếng Anh. Yêu cầu đặt ra là ta xây dựng được một ứng dụng học tiếng Anh miễn phí và đáp ứng đầy đủ nhu cầu học tập của mọi người. Ứng dụng của nhóm 4 đáp ứng một trong các nhu cầu đó là học từ vựng tiếng Anh.

## **Đặc tả yêu cầu chức năng**

Ứng dụng có đầy đủ các phần :

* Cho phép người dùng học từ mới thông qua các chủ đề
* Mỗi chủ đề gồm các từ mới và hình ảnh minh họa. Người dùng có thể click vào ảnh hoặc từ để đổi 2 thẻ cho nhau.
* Người dùng có thể tự tạo chủ đề của mình, thêm từ mới, thêm ảnh.
* Người dùng có thể thêm các chủ đề vào mục ưa thích để tiện cho việc học lần sau.

## **Đặc tả yêu cầu phi chức năng**

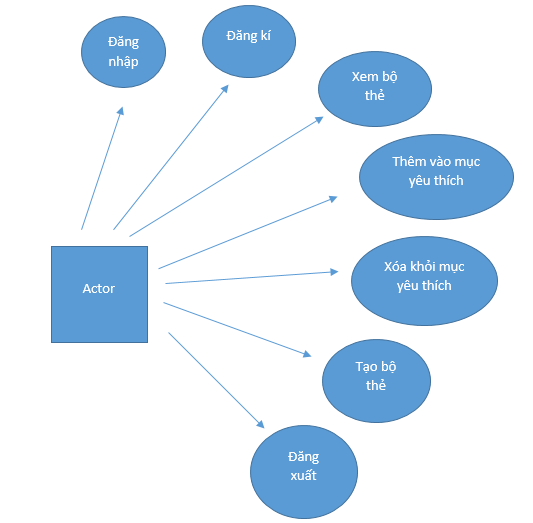
* Tính khả dụng :Giao diện đơn giản, thân thiện với người dùng, dễ dàng đăng ký với đăng nhập
* Tính tin cậy : ứng dụng hoạt động đúng với các chức năng đã được thiết kế.
* Khả năng tiếp cận : chỉ yêu cầu cài đặt trình duyệt Web cơ bản.
* Hiệu năng: Ổn định, sẵn sàng hoạt động mọi lúc,phản hồi theo thao tác người dùng.
* Tính bảo mật : giữ bảo mật mọi thông tin người dùng.

## **Tóm tắt yêu cầu từ người sử dụng**

* Là người dùng, tôi muốn đăng nhập, đăng kí dễ dàng..
* Là người dùng, tôi muốn có thể tự thêm các chủ đề..
* Là người dùng, tôi muốn các chủ đề được đa dạng và có thể đánh dấu là đã học.

## **Thiết kế ca sử dụng**

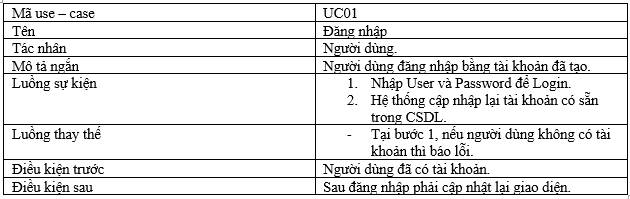
Hình dưới đây mô tả các ca sử dụng. Và phần tiếp theo sẽ phân tích và làm rõ các ca sử dụng này.



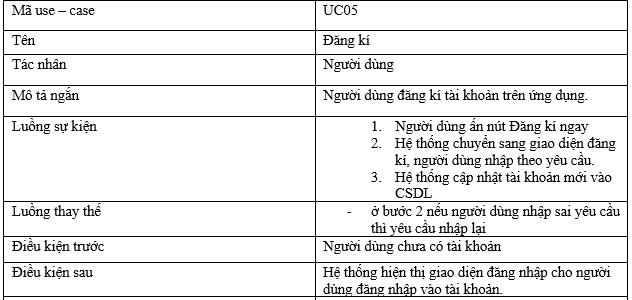
## **Phân tích ca sử dụng**

Phần sau đây sẽ mô tả từng ca sử dụng của ứng dụng.

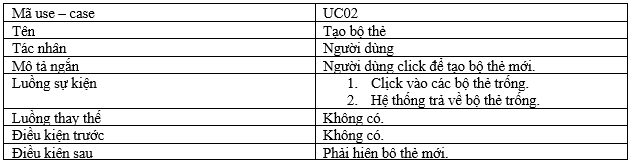
2.5.0: Ca sử dụng đăng nhập:



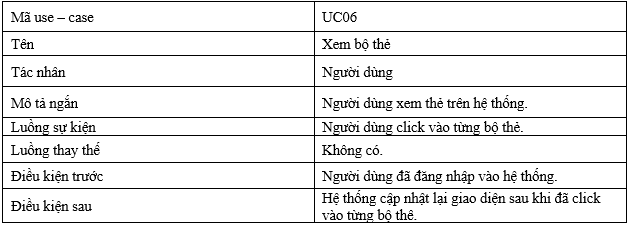
2.5.1: Ca sử dụng đăng kí:



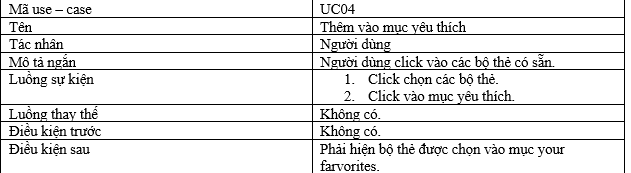
2.5.2: Ca sử dụng Tạo bộ thẻ:



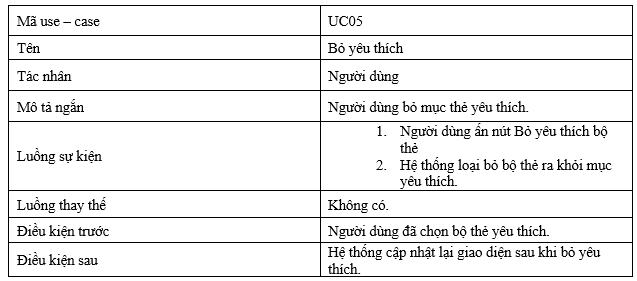
2.5.3: Xem bộ thẻ:



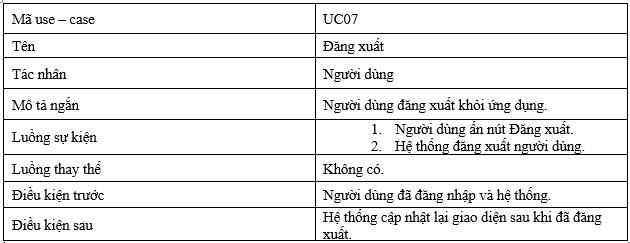
2.5.4: Ca sử dụng Thêm vào mục yêu thích:



2.5.5: Xóa khỏi mục yêu thích:



2.5.6: Ca sử dụng Đăng xuất:



# **Mô tả cài đặt ứng dụng**

## **Cài đặt Frontend**

Phần cài đặt chức năng và giao diện chính của ứng dụng được đặt trong thư mục app của Angular, cấu trúc của thư mục app được nêu ở bảng 3a:

|  |  |
| --- | --- |
| **Thư mục/ file** | **Mô tả** |
| Thư mục homepage | Chứa component cho trang chủ |
| Thư mục card | Chứa component card giúp hiển thị các bộ thẻ trên trang chủ |
| Thư mục create-card | Chứa component cho trang thêm bộ thẻ |
| Thư mục navigation-bar | Chức component cho thanh điều hướng |
| Thư mục learnpage | Chứa component cho trang học 1 bộ thẻ |
| Thư mục loginpage | Chứa component cho trang đăng nhập/ đăng ký |
| File Cookiee.ts | Chứa các hàm chức năng đọc, ghi cookie trình duyệt |
| File app-routing.module.ts | Chứa module route thực hiện chức năng điều hướng trang web |
| File user.guard.ts | Chứa module bảo vệ cho router, ngăn người dùng chưa đăng nhập truy cập vào 1 số route |
| File data.service.ts | Chứa các service( cụ thể là các hàm) thực hiện chức năng trao đổi dữ liệu, gửi request với server |

*bảng 3a- Cấu trúc thư mục app*

Các phần dưới đây sẽ mô tả từng phần của thư mục app

### ***Thư mục homepage***

* ***Thành phần homepage.component.html*** *để hiển thị trang chủ với các bộ thẻ theo 2 mục: “Your favorites”\_ chứa các bộ thẻ mà người dùng đánh dấu là ưa thích và “All cards”\_ chứa tất cả các bộ thẻ có trên hệ thống*
* ***Thành phần homepage.component.ts*** *có chức năng lấy các bộ thẻ từ dataservice để hiện thị*

### ***Thư mục card***

* ***Thành phần card.component.html*** *để hiển thị bộ thẻ với đầu vào là tiêu đề bộ thẻ, avatar của bộ thẻ, kiểu của avatar (img hoặc text)*
* ***Thành phần card.component.ts*** *chứa các @Input() để hỗ trợ chức năng nhận đầu vào dưới dạng property của html*

### ***Thư mục create-card***

* ***Thành phần create-card.component.html*** *để hiển thị giao diện thêm bộ thẻ với các nút “Thêm thẻ khác”, “Xóa”, “Tạo bộ thẻ này” và các ô nhập liệu cho Tên bộ thẻ, ảnh đại diện , mặt trước, mặt sau với các option miêu tả kiểu dữ liệu của thẻ/ ảnh đại diện là văn bản hay đường dẫn tới hình ảnh*
* ***Thành phần create-card.component.ts*** *chứa các hàm quan trọng :*
  + *createItem() thực hiện chức năng thêm 1 form nhập để nhập 1 thẻ mới vào trong bộ thẻ trên giao diện*
  + *deleteRow() giúp xóa 1 thẻ khỏi danh sách các thẻ được gọi khi nhấn vào nút “Xóa”*
  + *sendNewCardCollection() thực hiện việc lấy dữ liệu từ form nhập và yêu cầu dataservice gửi dữ liệu lên server*

### ***Thư mục navigation-bar***

* ***Thành phần navigation-bar.component.html*** *để hiển thị thanh điều hướng với các item “Tinycards\_N4” hoặc biểu tượng ngôi nhà ở màn hình nhỏ \_ dẫn đến trang chủ ; “Create” hoặc biểu tượng dấu cộng ở màn hình nhỏ \_ dẫn đến trang tạo bộ thẻ mới hoặc trang đăng nhập nếu người dùng chưa đăng nhập; “Đăng nhập” hoặc biểu tượng đăng nhập ở màn hình nhỏ (chỉ hiển thị khi người dùng chưa đăng nhập) \_ dẫn đến trang đăng nhập; “Đăng xuất” hoặc biểu tượng đăng xuất ở màn hình nhỏ \_ thực hiện chức năng đăng xuất và đưa ứng dụng về màn hình trang chủ; Phần hiển thị tên người dùng đã đăng nhập*
* ***Thành phần navigation-bar.component.ts*** *chứa các biến logedIn để lưu trạng thái đăng nhập, đăng xuất của người dùng, username để lưu tên người dùng đã đăng nhập; hàm logOut() thực hiện việc đăng xuất khi người dùng bấm vào nút đăng xuất*

### ***Thư mục learnpage***

* ***Thành phần learnpage.component.html*** *để hiển thị giao diện học bộ thẻ gồm:*
  + *Nút “X” để quay lại trang chủ*
  + *Nút “Thêm vào mục yêu thích” để thêm bộ thẻ đang học vào mục yêu thích*
  + *Nút “Xóa khỏi mục yêu thích” để bỏ bộ thẻ đang học khỏi mục yêu thích*
  + *Phần hiển thị thẻ gồm 2 mặt trước và sau có thẻ lật bằng cách click chuột vào*
  + *Nút “Next” để hiển thị thẻ tiếp theo*
  + *Nút “Back” để hiển thị thẻ trước đó*

***- Thành phần learnpage.component.ts*** *gồm các hàm quan trọng*

*- addToFavorite() thực hiện khi click vào nút Thêm vào mục ưa thích*

*- removeOfFavorite() thực hiện khi click vào nút Xóa khỏi mục ưa thích*

*- nextCard() thực hiện khi ấn vào nút next*

*- backCard() thực hiện khi ấn vào nút back*

### ***Thư mục loginpage***

* ***Thành phần loginpage.component.html*** *để hiển thị giao diện form đăng nhập hoặc form đăng ký tài khoản, người dùng có thể chuyển qua lại hai form bằng nút đăng ký ngay hoặc nút đăng nhập*
* ***Thành phần loginpage.component.ts*** *gồm các hàm quan trọng:*
  1. *login() thực hiện gửi nội dung form đăng nhập lên server khi người dùng bấm nút đăng nhập*
  2. *signup() thực hiện gửi nội dung form đăng ký lên server khi người dùng bấm nút đăng ký*
  3. *samPassword() thực hiện kiểm tra người dùng có nhập lại mật khẩu ở form đăng ký có giống với mật khẩu đã nhập trước đó không*

### ***File cookiee.ts***

* *export hàm getCookie() nhận vào tên cookie và trả về giá trị của cookie đó*
* *export hàm deleteAllCookie() thực hiện xóa tất cả các cookie đã lưu*

### ***File app-routing.module.ts***

* *Chứa mảng route gồm các path: /home, /create, learn/:type/:cardID, /login và các redirect path*
* *export module RouterModule cho module AppModule sử dụng*

### ***File user.guard.ts***

* *implement interface CanActivate có sẵn của Angular trả về kết quả true nếu người dùng đã đăng nhập để bảo vệ route /create*

### ***File data.service.ts***

* *Hàm logIn() gửi request post lên server chứa username và password và nhận về username nếu đăng nhập thành công*
* *Hàm signUp() gửi request post lên server với nội dung giống với hàm logIn và nhận response với status code 201 nếu đăng kí thành công*
* *Hàm getListFavoriteCollection() gửi request get kèm username của người dùng đang đăng nhập và nhận về mẳng các bộ thẻ người dùng đó yêu thích*
* *Hàm getListCardCollection() gửi request get đến server và nhận về tất cả bộ thẻ có trên hệ thống*
* *Hàm sendNewCardCollection() gửi request post chứa nội dung bộ thẻ mà người dùng muốn tạo và nhận về status code 201 nếu bộ thẻ được tạo thành công*
* *Hàm addFavorite() gửi request get kèm username và id của bộ thẻ yêu cầu thêm bộ thẻ vào danh sách yêu thích của người dùng hiện tại và nhận về status code 201 nếu thành công*
* *Hàm removeFavorite() gửi request get kèm username và id của bộ thẻ yêu cầu xóa bộ thẻ khỏi danh sách của người dùng hiện tại và nhận status code 201 nếu thành công*

## **Cài đặt Backend**

Server được cài đặt bằng Nodejs với framework Express để xây dựng các api cụ thể theo bảng sau:

|  |  |  |
| --- | --- | --- |
| **Đường dẫn** | **Phương thức** | **Chức năng** |
| /login | POST | Xác thực người dùng bằng thông tin gửi đến |
| /signup | POST | Tạo 1 tài khoản mới bằng thông tin gửi đến |
| /newCardCollection | POST | Thêm mới một bộ thẻ vào database với dữ liệu gửi lên |
| /list/CardCollection | GET | Lấy tất cả các bộ thẻ có trong database |
| /list/FavoriteCardCollection/userName=:userName | GET | Lấy tất cả các bộ thẻ yêu thích của người sử dụng có tên tài khoản là <username> |
| /CardCollection/cardID=:cardID | GET | Lấy tất cả các thẻ có trong bộ thẻ có id của bộ thẻ là <cardID> |
| /addFavorite/collectionID=:collectionID/userName=:userName | GET | Thêm bộ thẻ có id là <collectionID> vào danh sách yêu thích của người dùng có tên đăng nhập là <userName> |
| /removeFavorite/collectionID=:collectionID/userName=:userName | GET | Xóa bộ thẻ có id là <collectionID> khỏi danh sách yêu thích của người dùng có tên đăng nhập là <userName> |

## **Cài đặt cơ sở dữ liệu**

### ***Các collection***

* + Cơ sở dữ liệu bao gồm 4 collection được miêu tả trong bảng dưới đây:

|  |  |
| --- | --- |
| **Tên collection** | **Mô tả** |
| users | Lưu trữ thông tin của người sử dụng |
| cards | Lưu trữ thông tin của các thẻ độc lập |
| cardcollections | Lưu trữ thông tin của các bộ thẻ |
| user\_collections | Lưu trữ quan hệ người sử dụng yêu thích một bộ thẻ |

### **Document mẫu cho các collection**

* + **users**

*{*

*"\_id": {*

*"$oid": "5af8e156973d3e577802e0f2"*

*},*

*"userName": "tranquanglinh.pt@gmail.com",*

*"userPassword": "87654321",*

*"\_\_v": 0*

*}*

* + **cards**

*{*

*"\_id": {*

*"$oid": "5af7aac96212543f443c4c07"*

*},*

*"cardOf": "5af7a3b52c3176134c95f762",*

*"typeBack": "text",*

*"typeFront": "img",*

*"back": "Owl",*

*"front": "https://d9np3dj86nsu2.cloudfront.net/image/8a7907739579128f58cd41f59b5b500c",*

*"\_\_v": 0*

*}*

* + **cardcollections**

*{*

*"\_id": {*

*"$oid": "5af7a3b52c3176134c95f762"*

*},*

*"title": "Birds",*

*"avatar": "https://d9np3dj86nsu2.cloudfront.net/image/8249e80335efb41dadbd5c8936755183",*

*"typeAvatar": "img",*

*"\_\_v": 0*

*}*

* + **user\_collections**

*{*

*"\_id": {*

*"$oid": "5afa9a152e2f460004dc51ed"*

*},*

*"userName": "quanvu143@gmail.com",*

*"collectionID": "5af7a3b52c3176134c95f762",*

*"\_\_v": 0*

*}*