**BDRPC187TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

****

**KHÓA LUẬN TỐT NGHIỆP**

**Đề tài: Xây dựng ứng dụng thi trắc nghiệm tập luật trong sát hạch bằng lái xe trên nền NodeJS và Android**

**Giáo viên hướng dẫn: Th.S Trần Thanh Huân**

**Sinh viên thực hiện: Vũ Hải Đăng**

**Lớp : Kỹ thuật phần mềm 2 khóa 7**

**Mã sinh viên : 0741360122**

***Hà Nội, tháng 05 năm 2016***

**LỜI NÓI ĐẦU**

Trong quá trình thực hiện em đã gặp phải không ít khó khăn và trở ngại, tuy nhiên được sự góp ý từ các bạn cũng như thầy em đã hoàn thiện đề tài. Để hoàn thiện được đề tài “**Xây dựng ứng dụng thi trắc nghiệm tập luật trong sát hạch bằng lái xe trên nền NodeJS và Android**”. Em xin được gửi lời cảm ơn chân thành đến những người bạn đã góp ý và giúp đỡ em đặc biệt là thầy giáo hướng dẫn Thầy Trần Thanh Huân đã hết lòng giúp đỡ, hướng dẫn, chỉ dạy tận tình. Rất mong rằng Thầy và các bạn vẫn tiếp tục đồng hành cùng em và có những lời nhận xét nhiều hơn nữa để đề tài có thể ngày càng hoàn thiện hơn.

Xin chân thành cảm ơn!

|  |  |
| --- | --- |
|  | Sinh viên thực hiện:  Vũ Hải Đăng |

***Mục lục***

Chương 1 Mở đầu 9

1.1 Tên đề tài 9

1.2 Lý do chọn đề tài 9

1.3 Mục đích, đối tượng và phạm vi nghiên cứu. 9

1.3.1 Mục đích. 9

1.3.2 Đối tượng nghiên cứu. 9

1.3.3 Phạm vi nghiên cứu. 10

1.3.4 Kết cấu của đề tài. 10

1.4 Công cụ và công nghệ sử dụng 10

Chương 2 Cơ sở lý thuyết 11

2.1 Giới thiệu ngôn ngữ kịch bản JavaScript và ngôn ngữ lập trình Android. 11

2.1.1 Ngôn ngữ kịch bản JavaScript. 11

2.1.2 Ngôn ngữ lập trình Java và Android. 15

2.2 Giới thiệu các Framework và Platform. 20

2.2.1 Hệ quản trị cơ sở dữ liệu MongoDB. 20

2.2.2 Angular Client Side Javascript. 22

2.2.3 NodeJS – Javascript Platform. 25

2.2.4 Module Socket.IO, Agenda, Express Framework. 27

Chương 3 Xây dựng chương trình 29

3.1 Khảo sát 29

3.1.1 Khảo sát đề tài. 29

3.1.2 Yêu cầu về nghiệp vụ. 29

3.1.3 Các chức năng 30

3.2 Phân tích thiết kế hệ thống 32

3.2.1 Các tác nhân của hệ thống 32

3.2.2 Các ca sử dụng của hệ thống 33

3.2.3 Phân tích từng ca sử dụng 36

3.2.4 Thiết kế cơ sở dữ liệu 80

3.3 Triển khai ứng dụng trên máy chủ CentOS và kết quả. 84

3.3.1 Triển khai ứng dụng trên máy chủ CentOS. 84

3.3.2 Kết quả. 84

3.3.3 Giao diện 85

Tài liệu tham khảo 87

***Danh mục hình ảnh***

[Hình 2.1 Lưu trữ dữ liệu trong MongoDB. 21](#_Toc449666176)

[Hình 2.2 Sơ đồ hoạt động của MongoDB 21](#_Toc449666177)

[Hình 3.1 Biểu đồ ca sử dụng chính 33](#_Toc449666178)

[Hình 3.2 Biểu đồ ca sử dụng khách hàng 34](#_Toc449666179)

[Hình 3.3 Biểu đồ ca sử dụng của Quản trị hệ thống website 35](#_Toc449666180)

[Hình 3.4 Biểu đồ Use Case thành viên 36](#_Toc449666181)

[Hình 3.5 Biểu đồ hoạt động ca sử dụng đăng nhập 38](#_Toc449666182)

[Hình 3.6 .Biểu đồ lớp tham gia ca sử dụng đăng nhập 39](#_Toc449666183)

[Hình 3.7 Biểu đồ trình tự ca sử dụng đăng nhập. 40](#_Toc449666184)

[Hình 3.8 Biểu đồ hoạt động ca sử dụng Quản lý thành viên – hành động xóa thành viên 42](#_Toc449666185)

[Hình 3.9 Biểu đồ hoạt động ca sử dụng Quản lý thành viên – hành động sửa thông tin thành viên 43](#_Toc449666186)

[Hình 3.10 Biểu đồ lớp tham gia ca sử dụng Quản lý thành viên 44](#_Toc449666187)

[Hình 3.11 Biểu đồ trình tự ca sử dụng Quản lý thành viên – hành động xóa thành viên 45](#_Toc449666188)

[Hình 3.12 Biểu đồ trình tự ca sử dụng Quản lý thành viên – hành động sửa thành viên 46](#_Toc449666189)

[Hình 3.13 Biểu đồ hoạt động ca sử dụng Quản lý danh mục bài viết – hành động them danh mục. 49](#_Toc449666190)

[Hình 3.14 Biểu đồ hoạt động ca sử dụng Quản lý danh mục bài viết – hành động sửa danh mục. 50](#_Toc449666191)

[Hình 3.15 Biểu đồ hoạt động ca sử dụng Quản lý danh mục bài viết – hành động xóa danh mục. 51](#_Toc449666192)

[Hình 3.16 Biểu đồ lớp tham gia ca sử dụng Quản lý danh mục bài viết. 52](#_Toc449666193)

[Hình 3.17 Biểu đồ trình tự ca sử dụng Quản lý danh mục bài viết – hành động them danh mục. 52](#_Toc449666194)

[Hình 3.18 Biểu đồ trình tự ca sử dụng Quản lý danh mục bài viết – hành động sửa danh mục. 53](#_Toc449666195)

[Hình 3.19 Biểu đồ trình tự ca sử dụng Quản lý danh mục bài viết – hành động xóa danh mục. 54](#_Toc449666196)

[Hình 3.20 Biểu đồ hoạt động ca sử dụng Quản lý bài viết – hành động them bài viết. 56](#_Toc449666197)

[Hình 3.21 Biểu đồ hoạt động ca sử dụng Quản lý bài viết – hành động sửa bài viết. 57](#_Toc449666198)

[Hình 3.22 Biểu đồ hoạt động ca sử dụng Quản lý bài viết – hành động xóa bài viết. 58](#_Toc449666199)

[Hình 3.23 Biểu đồ trình tự ca sử dụng Quản lý bài viết – hành động them bài viết 59](#_Toc449666200)

[Hình 3.24 Biểu đồ trình tự ca sử dụng Quản lý bài viết – hành động sửa bài viết 60](#_Toc449666201)

[Hình 3.25 Biểu đồ trình tự ca sử dụng Quản lý bài viết – hành động xóa bài viết 62](#_Toc449666202)

[Hình 3.26 Biểu đồ hoạt động ca sử dụng đăng ký thành viên 64](#_Toc449666203)

[Hình 3.27 Biểu đồ lớp tham gia ca sử dụng đăng ký thành viên 65](#_Toc449666204)

[Hình 3.28 Biểu đồ trinh tự ca sử dụng đăng ký thành viên 66](#_Toc449666205)

[Hình 3.29 Biểu đồ hoạt động ca sử dụng Quản lý đề thi – hành động them đề thi. 69](#_Toc449666206)

[Hình 3.30 Biểu đồ hoạt động ca sử dụng Quản lý nhóm – hành động sửa đề thi. 70](#_Toc449666207)

[Hình 3.31 Biểu đồ hoạt động ca sử dụng Quản lý đề thi – hành động xóa đề thi. 72](#_Toc449666208)

[Hình 3.32 Biểu đồ lớp tham gia ca sử dụng Quản lý đề thi. 73](#_Toc449666209)

[Hình 3.33 Biểu đồ trình tự ca sử dụng Quản lý đề thi – hành động them đề thi. 74](#_Toc449666210)

[Hình 3.34 Biểu đồ trình tự ca sử dụng Quản lý đề thi – hành động sửa đề thi. 75](#_Toc449666211)

[Hình 3.35 Biểu đồ trình tự ca sử dụng Quản lý đề thi – hành động xóa đề thi. 76](#_Toc449666212)

[Hình 3.36 Biểu đồ hoạt động ca sử dụng quản lý comment. 77](#_Toc449666213)

[Hình 3.37 Biểu đồ lớp tham gia ca sử dụng quản lý comment. 77](#_Toc449666214)

[Hình 3.38 Biểu đồ trình tự ca sử dụng quản lý comment. 78](#_Toc449666215)

[Hình 3.39 Biểu đồ trình tự ca sử dụng làm bài thi. 79](#_Toc449666216)

[Hình 3.40 Biểu đồ lớp tham gia ca sử dụng làm bài thi. 80](#_Toc449666217)

[Hình 3.41 Biểu đồ trình tự ca sử dụng làm bài thi. 81](#_Toc449666218)

[Hình 3.42 Collection Member. 82](#_Toc449666219)

[Hình 3.43 Collection Example. 82](#_Toc449666220)

[Hình 3.44 Collection Question. 83](#_Toc449666221)

[Hình 3.45 Collection Answer. 83](#_Toc449666222)

[Hình 3.46 Collection Comment. 84](#_Toc449666223)

[Hình 3.47 Collection TypeExamp. 84](#_Toc449666224)

[Hình 3.48 Collection Post. 85](#_Toc449666225)

[Hình 3.49 Giao diện ứng dụng trên nền WEB. 86](#_Toc449666226)

[Hình 3.50 Giao diện ứng dụng trên Mobile nền tảng Android. 87](#_Toc449666227)

**DANH SÁCH BẢNG BIỂU**

[Bảng 2.1 Toán tử và chức năng 14](#_Toc449686922)

[Bảng 2.2 Kiểu dữ liệu. 15](#_Toc449686923)

[Bảng 2.3 Các thành phần trong AngularJS 26](#_Toc449686924)

[Bảng 3.1 Chức năng của quản trị viên. 32](#_Toc449686925)

[Bảng 3.2 Chức năng người dùng. 33](#_Toc449686926)

[Bảng 3.3 Tác nhân của hệ thống. 34](#_Toc449686927)

[Bảng 3.4 Đặc tả ca sử dụng đăng nhập. 37](#_Toc449686928)

[Bảng 3.5 Đặc tả ca sử dụng quản lý thành viên. 40](#_Toc449686929)

[Bảng 3.6 Đặc tả ca sử dụng xóa thành viên. 41](#_Toc449686930)

[Bảng 3.7 Đặc tả ca sử dụng Quản lý thành viên – hành động sửa thông tin thành viên. 41](#_Toc449686931)

[Bảng 3.8 Đặc tả ca sử dụng quản lý danh mục bài viết. 47](#_Toc449686932)

[Bảng 3.9 Đặc tả ca sử dụng quản lý danh mục bài viết – hành động thêm danh mục. 47](#_Toc449686933)

[Bảng 3.10 Đặc tả ca sử dụng quản lý danh mục bài viết – hành động sửa danh mục. 48](#_Toc449686934)

[Bảng 3.11 Đặc tả ca sử dụng quản lý danh mục bài viết – hành động xóa danh mục. 48](#_Toc449686935)

[Bảng 3.12 Đặc tả ca sử dụng quản lý bài viết. 54](#_Toc449686936)

[Bảng 3.13 Đặc tả ca sử dụng quản lý bài viết- hành động thêm bài viết. 55](#_Toc449686937)

[Bảng 3.14 Đặc tả ca sử dụng quản lý bài viết- hành động sửa bài viết. 55](#_Toc449686938)

[Bảng 3.15 Đặc tả ca sử dụng quản lý bài viết- hành động xóa bài viết. 55](#_Toc449686939)

[Bảng 3.16 Đặc tả ca sử dụng đăng ký thành viên. 63](#_Toc449686940)

[Bảng 3.17 Đặc tả ca sử dụng quản lý đề thi. 67](#_Toc449686941)

[Bảng 3.18 Đặc tả ca sử dụng quản lý đề thi - hành động thêm đề thi. 67](#_Toc449686942)

[Bảng 3.19 Đặc tả ca sử dụng quản lý đề thi – hành động sửa đề thi. 68](#_Toc449686943)

[Bảng 3.20 Đặc tả ca sử dụng quản lý đề thi – hành động xóa đề thi. 68](#_Toc449686944)

[Bảng 3.21 Đặc tả ca sử dụng quản lý comment. 77](#_Toc449686945)

[Bảng 3.22 Đặc tả ca sử dụng làm bài thi. 79](#_Toc449686946)

**DANH SÁCH TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| Từ viết tắt | Định nghĩa |
| QTV | Quản trị viên |
| JS | Javascript |
| OOP | Hướng đối tượng |
| FW | Khung làm việc |
| Lib | Thư viện |
| I/O | Vào/ra |

# Mở đầu

## Tên đề tài

“Xây dựng ứng dụng thi trắc nghiệp tập luật trong sát hạch bằng lái xe trên nền NodeJS và Android.”

## Lý do chọn đề tài

Ngày nay, việc học để có một chiếc bằng lái xe máy trở nên rất phổ biến. Một trong những nhu cầu khiến nhiều người học lái xe là rất nhiều công việc đòi hỏi phải có bằng lái xe. Do đó, một chiếc bằng lái sẽ mở ra thêm nhiều cơ hội việc làm cho mọi người. Nhưng với nhịp sống hối hả của cuộc sống không phải ai cũng có thời gian, điều kiện để tham gia các lớp học lý thuyết thi sát hạch lái xe. Bên cạnh đó, việc học lý thuyết trên những trang sách cũng khiến nhiều người nhàm chán. Vì thế chúng ta cần phải có phương tiện nào đó để việc học lý thuyết trở nên dễ dàng và tiện lợi.  
Với sự phát triển ngày càng nhanh chóng của khoa học công nghệ,Website không chỉ đơn giản là những phương tiện cung cấp tin tức, mua sắm,…và điện thoại di động không còn đơn thuần là thiết bị liên lạc, mà trở thành một công cụ hỗ trợ và giải trí đa phương tiện và một vật không thể thiếu của mọi người. Chính vì sự phát triển mạnh mẽ của thiết bị di động, ý tưởng được đưa ra là xây dựng một ứng dụng trên thiết bị di động giúp mọi người có thể học thi lý thuyết sát hạch bằng lái xe trở nên đơn giản, tiết kiệm thời gian và học bất cứ lúc nào rảnh rội ngay trên chiếc điện thoại của mình

## Mục đích, đối tượng và phạm vi nghiên cứu.

### Mục đích.

* Phân tích, thiết kế, xây dựng hệ thống thi trắc nghiệm tập luật dựa trên tập luật đường bộ dành cho người tham gia sát hạch bằng lái xe trên nền web và di động.
* Tìm hiểu và nắm vững lập trình ứng dụng trên nền Web với NodeJS và lập trình trên Mobile với Android.

### Đối tượng nghiên cứu.

* Ngôn ngữ kịch bản JavaScript.
* Ngôn ngữ lập trình Java.
* Hệ điều hành Android
* NodeJS Javascript Platform.

### Phạm vi nghiên cứu.

* Câu hỏi trắc nghiệm trong thi sát hạch bằng lái xe.
* Lưu trữ câu hỏi và đáp án.
* Triển khai ứng dụng trên nền Web và Mobile sử dụng hệ điều hành Android.

### Kết cấu của đề tài.

Nội dung gồm 4 chương

* CHƯƠNG 1: MỞ ĐẦU
* Tên đề tài.
* Lý do chọn đề tài.
* Mục đích – mục tiêu
* Công cụ và công nghệ sử dụng.
* CHƯƠNG 2: CƠ SỞ LÝ THUYẾT
* Giới thiệu ngôn ngữ kịch bản JavaScript và ngôn ngữ lập trình Java và Android.
* Giới thiệu các Framework, Platform.
* CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH
* Khảo sát.
* Phân tích thiết kế hệ thống.
* Thực nghiệm và kết quả..

## Công cụ và công nghệ sử dụng

* Trình soạn thảo CODE Sublime Text 3.
* Hệ quản trị cơ sở dữ liệu MongoDB.
* Máy chủ NodeJS.
* Ngôn ngữ kịch bản : JavaScript.
* Ngôn ngữ lập trình Android.
* Framework CSS Bootstrap 3.
* Framework NodeJS Express v4.4.
* AngularJS Client Side Javascript v1.32.
* Module SocketIO, Agenda, Mongoose, Multer,…

# Cơ sở lý thuyết

## Giới thiệu ngôn ngữ kịch bản JavaScript và ngôn ngữ lập trình Android.

### Ngôn ngữ kịch bản JavaScript.

#### Lịch sử phát triển?

Từ lâu, các nhà phát triển Java™ đã xem JavaScript như một ngôn ngữ đồ chơi, vừa quá nhẹ để lập trình thực sự và vừa quá vụng về để sử dụng thay thế cho việc tạo kịch bản lệnh. Vậy mà JavaScript vẫn còn đâu đó và nó là cơ sở của các công nghệ web thú vị như GWT và Node.js. Trong bài đăng này về [phát triển Java 2.0](https://www.ibm.com/developerworks/vn/views/java/libraryview.jsp?search_by=Ph%C3%A1t+tri%E1%BB%83n+Java+2.0:), Andrew Glover giải thích tại sao JavaScript là một công cụ quan trọng đối với các nhà phát triển Java hiện đại. Sau đó ông sẽ khiến bạn bắt tay vào cú pháp cần dùng để xây dựng các ứng dụng hạng nhất cho các trang web ngày nay, bao gồm cả các biến, các kiểu, các hàm và các lớp của JavaScript.

JavaScript được Netscape giới thiệu vào năm 1995 và nó nhanh chóng được công chúng yêu thích. Lý do có liên quan nhiều đến sự xuất hiện của các trang web như là một nền tảng môi trường truyền thông thương mại: Với JavaScript, bạn có thể tác động đến hành vi của trang web trong trình duyệt bằng lập trình. Chỉ điều đó thôi cũng đã thú vị rồi! Việc xác nhận hợp lệ của dạng HTML và mánh khóe hình ảnh hạn chế đã là vấn đề quan trọng hàng đầu.

Kể từ đó, JavaScript đã trải qua một vài thay đổi. Vào lúc đó Netscape đã có một sản phẩm là Netscape Application Server (Máy chủ ứng dụng Netscape), dựa trên JavaScript phía máy chủ để xây dựng các ứng dụng web. Vài năm sau đó, sự ra đời của Ajax và các thư viện tiện ích như JQuery, Prototype và ExtJS đã tiếp tục châm ngòi cho các mối quan tâm về JavaScript. Gần đây hơn, JavaScript phía máy chủ đã trở lại với Node.js, một framework I/O dựa trên sự kiện dùng để xây dựng các ứng dụng web phía máy chủ bằng cách sử dụng công cụ V8 JavaScript của Google.

Để chăm lo cho tương lai của sản phẩm của mình, Netscape đã gửi JavaScript lên tổ chức quốc tế về tiêu chuẩn hóa Ecma. Đó là lý do tại sao một số người gọi JavaScript là ECMAScript. Quan trọng hơn, đó là lý do tại sao hầu hết các trình duyệt web đều hỗ trợ ECMAScript. Kết quả là, các ứng dụng web gặp khó khăn đều không sử dụng JavaScript ở đâu đó và JavaScript dường như không thể thay đổi được điều đó. Hiện tại chưa có ngôn ngữ tạo kịch bản lệnh nào khác tương thích trình duyệt xuất hiện.

Ngày nay, mặc dù tiếng xấu của nó vẫn còn sót lại, nhưng JavaScript được cho là một trong những ngôn ngữ được sử dụng nhiều nhất (và có ích) trên hành tinh.

#### Phân loại và môi trường chạy.

**Javascript** được chia thành 2 loại theo môi trường hoạt động:

* *Client-side JavaScript* - JavaScript phía máy khách, JavaScript được mở rộng bằng cách cung cấp các Object để quản lý trình duyệt và Document Object Model (DOM)  của nó. Ví dụ, phần mở rộng phía máy khách cho phép một ứng dụng tác động tới các yếu tố trên một trang HTML và phản hồi giống các tác động của người dùng như click chuột, nhập form, và chuyển trang.
* *Server-side JavaScript* - JavaScript phía Server, JavaScript được mở rộng bằng cách cung cấp thêm các đối tượng cần thiết để để chạy JavaScript trên máy chủ. Ví dụ, phần mở rộng phía server này cho phép ứng dụng kết nối với cơ sở dữ liệu (database), cung cấp thông tin một cách liên tục từ một yêu cầu tới phần khác của ứng dụng, hoặc thực hiện thao tác với các tập tin trên máy chủ.

#### Cú pháp, toán tử hàm dữ liệu trong Javascript.

* 1. Cú pháp
* Javascript được viết trong tệp tin có phần mở rộng là \*.js.
* Phân biệt chữ hoa và chữ thường.

JavaScript phân biệt chữ hoa và chữ thường. Cần chú ý điều này khi đặt tên cho biến và sử dụng từ khóa của ngôn ngữ. Một biến được đặt tên là remote sẽ khác biến có tên là Remote hay REMOTE. Tương tự, từ khóa cho vòng lặp là while, nhưng nếu viết thành WHILE hoặc Whilethì sẽ gây ra lỗi.

* Ký tự trắng.

Trong hầu hết trường hợp, JavaScript bỏ qua ký tự trắng nằm giữa các câu lệnh. Bạn có thể thêm ký tự trắng, lùi đầu dòng hoặc viết mã theo bất cứ quy ước mã nào để đoạn mã JavaScript rõ ràng và dễ đọc hơn.

* Chú thích.

Chú thích trong Javascript được đặt trong cặp ký tự /\* và \*/ hoặc đặt sau ký tự //.

* Dấu chấm phẩy.

Trong **JavaScript**, dấu chấm phẩy được sử dụng để phân tách các biểu thức. Về mặt kỹ thuật, dấu chấm phẩy là không bắt buộc trong hầu hết các câu lệnh và biểu thức. Tuy nhiên, những vấn đề khó lường bạn gặp phải khi không sử dụng dấu chấm phẩy có thể gây ra những lỗi không đúng do đó làm tốn thời gian tìm và gỡ lỗi. Trong một số trường hợp, trình thông dịch JavaScript có thể tự động chèn thêm dấu chấm phẩy khi bạn không muốn

* 1. Toán tử.

**Javascript** có các toán tử giống với ngôn ngữ lập trình C/C++.

|  |  |
| --- | --- |
| Toán tử, Logic | Chức năng |
| +, -, \*, / | Phép toán giữa 2 hay nhiều số. |
| % | Tính % của 1 số |
| ++, --, +=,-= | Tăng , giảm 1 số theo 1 giá trị. |
| || | Quan hệ hoặc |
| && | Quan hệ và |

Toán tử và chức năng

* 1. Kiểu dữ liệu, biến, hàm.

Kiểu dữ liệu trong **Javascript**.

|  |  |  |
| --- | --- | --- |
| Kiểu dữ liệu | Định nghĩa | Khai báo |
| String | Kiểu dữ liệu dạng chuỗi | Ten\_bien = “”; |
| Number | Kiểu dữ liệu dạng số. | Ten\_bien = so; |
| Array | Kiểu dữ liệu dạng mảng | Ten\_bien = []; |
| Object | Kiểu dữ liệu dạng đối tượng | Ten\_bien = {}; |

Kiểu dữ liệu.

Hàm trong **Javascript** được tạo ra theo cấu trúc sau:

*function* ten\_ham(arguments){

//thực thi của hàm

}

### Ngôn ngữ lập trình Java và Android.

#### Lịch sử phát triển.

Năm 1990, Sun MicroSystems thực hiện dự án Green nhằm phát triển phần mềm trong các thiết bị dân dụng. James Gosling, chuyên gia lập trình đã tạo ra một ngôn ngữ lập trình mới có tên là Oak. Ngôn ngữ này có cú pháp gần giống như C++ nhưng bỏ qua các tính năng nguy hiểm của C++ như truy cập trực tiếp tài nguyên hệ thống, con trỏ, định nghĩa chồng các tác tử…

#### Khả năng của Java.

Là một ngôn ngữ bậc cao như C, C++, Perl, SmallTalk,.. cho nên có thể được dùng để tạo ra các ứng dụng để giải quyết các vấn đề về số, xử lý văn bản, tạo ra trò chơi, và nhiều thứ khác.

Có các môi trường lập trình đồ họa như  Visual Java, Symantec Cafe, Jbuilder, Jcreator, ...

Có khả năng truy cập dữ liệu từ xa thông qua cầu nối JDBC (Java DataBase Connectivity)

Hỗ trợ các lớp hữu ích, tiện lợi trong lập trình các ứng dụng mạng (Socket) cũng như truy xuất Web.

Hỗ trợ lập trình phân tán  (Remote Method Invocation ) cho phép một ứng dụng có thể được xử lý phân tán trên các máy tính khác nhau.

Và luôn được bổ sung các tính năng cao cấp khác trong các phiên bản sau.

#### Bộ phát triển ứng dụng Java (JDK- Java Development Kit).

JDK là một bộ công cụ cho phép người lập trình phát triển và triển khai các ứng dụng bằng ngôn ngữ java được cung cấp miễn phí bởi công ty JavaSoft (hoặc Sun). Có các bộ Jdk cho các hệ điều hành khác nhau. Các ấn bản của JDK không ngừng được phát hành, các bạn có thể tải về từ địa chỉ [http://java.sun.com](http://java.sun.com/) hoặc [http://www.javasoft.com](http://www.javasoft.com/)

Bộ công cụ này gồm các chương trình thực thi đáng chú ý sau:

javac: Chương trình biên dịch các chương trình nguồn viết bằng ngôn ngữ java ra các tập tin thực thi được trên máy ảo Java.

java: Đây là chương trình làm máy ảo của Java, thông dịch mã Bytecode của các chương trình kiểu application thành mã thực thi của máy thực.

* appletviewer: Bộ thông dịch, thực thi các chương trình kiểu applet.
* javadoc: Tạo tài liệu về chú thích chương trình nguồn một cách tự động.
* jdb: Trình gở rối.
* rmic: Tạo Stub cho ứng dụng kiểu RMI.

rmiregistry: Phục vụ danh bạ (Name Server) trong hệ thống RMI.

#### Các đặc điểm cơ bản của lập trình hướng đối tượng.

Ngôn ngữ hoàn toàn hướng đối tượng.Ngôn ngữ đa nền cho phép một chương trình có thể thực thi trên các hệ điều hành khác nhau (MS Windows, UNIX, Linux) mà không phải biên dịch lại chương trình. Phương châm của java là **"Viết một lần ,  Chạy trên nhiều nền"**  (Write Once, Run Anywhere).Ngôn ngữ đa luồng, cho phép trong một chương trình có thể có nhiều luồng điều khiển được thực thi song song nhau, rất hữu ích cho các xử lý song song. Ngôn ngữ phân tán, cho phép các đối tượng của một ứng dụng được phân bố và thực thi trên các máy tính khác nhau.Ngôn ngữ động, cho phép mã lệnh của một chương trình được tải từ một máy tính về máy của người yêu cầu thực thi chương trình.Ngôn ngữ an toàn, tất cả các thao tác truy xuất vào các thiết bị vào ra đều thực hiện trên máy ảo nhờ đó hạn chế các thao tác nguy hiểm cho máy tính thật.Ngôn ngữ đơn giản, dễ học, kiến trúc chương trình đơn giản, trong sang.

##### Tính đóng gói.

Encapsulation (tính bao đóng) cung cấp cho những đối tượng khả năng che dấu tính chất đặc thù và những hành vi bên trong chúng. Mỗi đối tượng cung cấp một số các phương thức, chúng có thể được truy cập từ những đối tượng khác và có thể bị thay đổi dữ liệu bên trong nó. Trong Java, có 3 mức(modifier) truy cập:  public, private và protected. Mỗi mức bắt buộc những quyền truy cập khác nhau từ những lớp (class) khác, dù là cùng hay khác package. Một vài lợi ích khi sử dụng encapsulation:

* Ẩn những trạng thái (state) của đối tượng cần được bảo vệ.
* Tăng khả năng bảo trì và sử dụng mã lênh, bởi vì những hành vi của đối tượng có thể độc lập khi bị thay đổi hay mở rộng.

- Tăng khả năng modun hoá bằng cách tránh những đối tượng tương tác trực tiếp lẫn nhau, trong cách nhờ vả nhau.

##### Tính đa hình.

Polymorphism (tính đa hình) là khả năng của ngôn ngữ lập trình để biểu thị những interface (giao diện) giống nhau cho những kiểu dữ liệu cơ bản khác nhau. Kiểu đa hình là kiểu mà những xử lý bên trong nó có thể được áp dụng lên giá trị của những kiểu khác.

##### Tính kế thừa

Kế thừa cung cấp cho đối tượng khả năng sử dụng được những trường (field), phương thức (method) từ những lớp khác, lớp đó gọi là lớp cơ bản (lớp cha, base class). Kế thừa cung cấp khả năng tái sử dụng mã lệnh và có thể được sử dụng để thêm những tính năng mới cho một lớp đã có trước nhưng không làm thay đổi lớp đó.

##### Tính trừu tượng.

Abstraction một một quá trình để tách biệt những ý tưởng với những thực thể cụ thể và vì vậy, phát triển lớp trong những điều khoản (term) của các chức năng bên trong nhưng không cài đặt chi tiết những chắc năng đó. Java cung cấp khả năng khởi tạo lớp trừu tượng và những lớp trừu tượng đã có như interfaces (không phải cài đặt chi tiết cho tất cả các phương thức trong lớp trừu tượng). Kỹ thuật trừu tượng nhằm giúp tách biệt việc cài đặt chi tiết của lớp từ những hành vi của nó.

#### Hệ điều hành Android.

##### Giới thiệu về hệ điều hành Android

[Android](https://www.dienmayxanh.com/dien-thoai?g=android) là một hệ điều hành có mã nguồn mở dựa trên nền tảng Linux được thiết kế dành cho các [thiết bị di động](https://www.dienmayxanh.com/dien-thoai) có [màn hình cảm ứng](http://vi.wikipedia.org/wiki/M%C3%A0n_h%C3%ACnh_c%E1%BA%A3m_%E1%BB%A9ng) như điện thoại thông minh và máy tính bảng.

Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google, sau này được chính Google mua lại vào năm 2005 và hệ điều hành Android đã ra mắt vào năm 2007. Chiếc điện thoại đầu tiên chạy Android là HTC Dream được bán vào ngày 22 tháng 10 năm 2008.

Chính mã nguồn mở cùng với giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên được điều chỉnh và phân phối Android một cách tự do. Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới. Android chiếm 75% thị phần [điện thoại thông minh](https://www.dienmayxanh.com/tag/dien-thoai-smartphone)trên toàn thế giới vào thời điểm quý 3 năm 2012, với tổng cộng 500 triệu thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày. Tháng 10 năm 2012, đã có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play (cửa hàng ứng dụng chính của Android) ước tính khoảng 25 tỷ lượt. Hiện nay con số này đã giảm xuống do sự ảnh hưởng lớn của [iOS từ Apple](https://www.dienmayxanh.com/dien-thoai-apple-iphone) và một phần nhỏ của [Windows Phone](https://www.dienmayxanh.com/dien-thoai?g=khoang-4#_), tuy nhiên Android vẫn dẫn đầu thị phần.

Giao diện người dùng của [Android](https://www.dienmayxanh.com/dien-thoai?g=android) dựa trên nguyên tắc tác động trực tiếp, sử dụng cảm ứng chạm tương tự như những động tác ngoài đời thực như vuốt, chạm, kéo dãn và thu lại để xử lý các đối tượng trên màn hình.

Các thiết bị Android sau khi khởi động sẽ hiển thị màn hình chính, điểm khởi đầu với các thông tin chính trên thiết bị, tương tự như khái niệm desktop trên máy tính để bàn. Màn hính chính Android thường gồm nhiều biểu tượng (icon) và tiện ích (widget). Giao diện màn hình chính của Android có thể tùy chỉnh ở mức cao, cho phép người dùng tự do sắp đặt hình dáng cũng như hành vi của thiết bị theo sở thích.

Những ứng dụng do các hãng thứ ba có trên Google Play và các kho ứng dụng khác còn cho phép người dùng thay đổi "chủ đề" của màn hình chính, thậm chí bắt chước hình dáng của hệ điều hành khác như [Windows Phone](https://www.dienmayxanh.com/dien-thoai?g=khoang-4#_) hay iOS. Phần lớn những nhà sản xuất, và một số nhà mạng, thực hiện thay đổi hình dáng và hành vi của các thiết bị Android của họ để phân biệt với các hãng cạnh tranh.

Google đưa ra các bản cập nhật lớn cho [Android](https://www.dienmayxanh.com/dien-thoai?g=android) theo chu kỳ từ 6 đến 9 tháng, mà phần lớn thiết bị đều có thể nhận được qua sóng không dây. Bản cập nhật lớn mới nhất là Android 6.

##### Phát triển ứng dụng nền tảng Android.

**-JDK(Java Development Kit)**  
 Là một môi trường phát triển dùng để viết các ứng dụng Java. JDK gồm môi trường thực thi các ứng dụng Java (JRE), các công cụ hỗ trợ như: trình biên dịch, trình gỡ lỗi và các thư viện tiện ích giúp người lập trình xây dựng ứng dụng dễ dàng.  
**- Android studio**  
 Là một phần mềm miễn phí, được sử dụng để xây dựng ứng dụng Android. Sử dụng Android studio, nhà phát triển có thể tích hợp với nhiều công cụ hỗ trợ khác để có thể được một bộ công cụ hoàn chỉnh.  
**- Android SDK**  
 SDK (Software Development Kit ) đó là bộ công cụ phát triền phần mềm, mà cụ thể là phát triển ứng dụng cho hệ điều hành Android. Sản phẩm này do chính Google xây dựng và phát hành miễn phí cho tất cả những người muốn tìm hiểu và viết các ứng dụng trên nền tảng này.  
**- ADT Plugin**  
 ADT (Android Development Tools) là một thành phần mở rộng của Eclipse, chứa những thành phần trong Android SDK và được thêm vào Eclipse để tạo một môi trường tích hợp xây dựng ứng dụng Android.   
Có nhiều môi trường để phát triển ứng dụng cho Android nhưng Google khuyến khích bạn sử dụng Eclipse bởi vì đây là môi trường được Google xây dựng các tính năng tích hợp khá hoàn chỉnh.  
Để tải bộ công cụ các bạn nên truy cập link sau:  
JDK:  
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

## Giới thiệu các Framework và Platform.

### Hệ quản trị cơ sở dữ liệu MongoDB.

#### MongoDB là gì ?

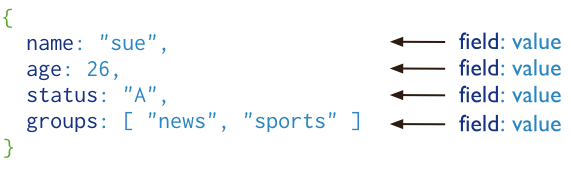
**MongoDB** là một cơ sở dữ liệu mã nguồn mở thuộc họ nhà ***NoSQL*** được phát hành sử dụng miễn phí bởi công ty MongoDB. Nếu bạn là người mới nghiên cứu về NoSQL thì đây là một cơ sở dữ liệu đầu tiên nên tiếp cận bởi nó dễ dàng cài đặt và sử dụng để tiếp cận với những ý tưởng lưu trữ mới. MongoDB hoạt động trên mọi hệ điều hành phổ biến hiện nay như: Windows, Linux, OS X và hỗ trợ đầy đủ các driver cho các ngôn ngữ lập trình để giao tiếp với chúng một cách dễ dàng nhất.

#### Các khái niệm mới trong MongoDB.

* Mỗi một table (bảng dữ liệu) trong SQL sử dụng thì trong MongoDB gọi là collection (tập hợp)
* Mỗi một record (bản ghi) trong MongoDB được gọi là document (tài liệu)

#### Một bản ghi trong MongoDB được lưu trữ như thế nào ?

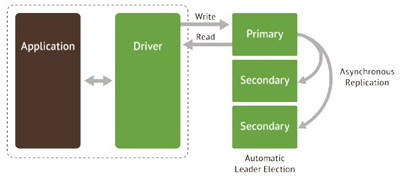
Một bản ghi của MongoDB được lưu trữ dưới dạng ***document*** (tài liệu), nó được ghi xuống với cấu trúc ***field*** (trường) và ***value*** (giá trị). Nó giống như là một đối tượng ***JSON*** có dạng như sau:



Lưu trữ dữ liệu trong MongoDB.

Điều đó giúp bạn có thể dễ dàng ép kiểu sang ***array*** (mảng) để lập trình các ứng dụng một cách dễ dàng hơn. Nói một cách dễ hiểu thì mỗi một bản ghi của MongoDB là một mảng dữ liệu riêng biệt bao gồm các cặp key, value khác nhau do đó cách lưu trữ của MongoDB là phi cấu trúc dữ liệu.

#### MongoDB hoạt động như thế nào ?



Sơ đồ hoạt động của MongoDB

* MongoDB hoạt động dưới một tiến trình ngầm service luôn mở một cổng (Cổng mặc định là 27017) để lắng nghe các yêu cầu truy vấn, thao tác từ các ứng dụng gửi vào sau đó mới tiến hành xử lý.
* Mỗi một bản ghi của MongoDB được tự động gắn thêm một field có tên “\_id” thuộc kiểu dữ liệu ObjectId mà nó quy định để xác định được tính duy nhất của bản ghi này so với bản ghi khác, cũng như phục vụ các thao tác tìm kiếm và truy vấn thông tin về sau.
* Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.
* Mỗi khi có một truy vấn dữ liệu, bản ghi được cache (ghi đệm) lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.
* Khi có yêu cầu thêm/sửa/xóa bản ghi, để đảm bảo hiệu suất của ứng dụng mặc định MongoDB sẽ chưa cập nhật xuống ổ cứng ngay, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng.

#### Ưu – nhược điểm của MongoDB.

* Ưu điểm
* Dữ liệu lưu trữ phi cấu trúc, không có tính ràng buộc, toàn vẹn nên tính sẵn sàng cao, hiệu suất lớn và dễ dàng mở rộng lưu trữ
* Dữ liệu được caching (ghi đệm) lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao
* Nhược điểm.
* Không ràng buộc, toàn vẹn nên không ứng dụng được cho các mô hình giao dịch yêu cầu độ chính xác cao
* Không có cơ chế transaction (giao dịch) để phục vụ các ứng dụng ngân hàng
* Dữ liệu được caching, lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn
* Như đã giới thiệu ở trên, mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

### Angular Client Side Javascript.

#### AngularJS là gì ?

**AngularJS** là một js framework dành xây đựng ứng dụng web động (dynamic web app).

Điều này có nghĩa là AngularJs cho phép chúng ta xây dựng một ứng dụng web bằng cách sử dụng framework được xây dựng trên nền tảng ngôn ngữ client là javascript. AngularJs là một framework thì cũng giống như các framework khác, ắt hẳn nó đã cung cấp các library, mô hình MVC, phương thức giúp ta phát triển một web app nhanh hơn.

Nhờ AngularJs, chúng ta sẽ viết code ít hơn, nhưng làm được nhiều hơn vì AngularJs đã xây dựng sẵn cho chúng ta nhiều thứ.

#### Lịch sử phát triển.

Được phát triển từ năm 2009 bởi Miško Hevery, nhân viên của Google. Trong khi đang tham gia một dự án khác Miško Hevery cùng nhóm đã rút ngắn số dòng code front-end từ 17000 dòng xuống chỉ còn khoảng 1500 dòng với AngularJS.

Phiên bản 1.0 được dưa ra vào tháng 6 năm 2012. Và cho đến thời điểm hiện tại (khi viết bài này) phiên bản đã là 1.3.15.

Mục đích thiết kế AngularJS hướng tới:

* Rút ngắn số dòng code, tổ chức source code tốt hơn.
* Tách riêng phần xử lý ở client của ứng dụng ra khỏi phần xử lý phía serve. Điều này cho phép việc phát triển diễn ra song song và cho phép tái sử dụng ở cả hai phía.
* Tách các thao tác với DOM ra khỏi application logic. Cải thiện khả năng test của mã nguồn.

#### Sơ lược về cấu trúc và các thành phần.

|  |  |
| --- | --- |
| **Nội dung** | **Mô tả** |
| [Template](https://docs.angularjs.org/guide/concepts#template) | Sử dụng ngôn ngữ HTML |
| [Directives](https://docs.angularjs.org/guide/concepts#directive) | Cho phép mở rộng HTML và bạn có thể custom lại các thuộc tính, phần tử (elements) |
| [Model](https://docs.angularjs.org/guide/concepts#model) | Lưu trữ dữ liệu và hiển thị ra ngoài view của người dùng khi có 1 tương tác nào đó |
| [Scope](https://docs.angularjs.org/guide/concepts#scope) | Điểm trung gian, làm nhiệm vụ giao tiếp giữa model, controller, service, view,.. |
| [Expressions](https://docs.angularjs.org/guide/concepts#expression) | Truy cập vào các biến và function từ scope |
| [Compiler](https://docs.angularjs.org/guide/concepts#compiler) | parses template and khởi tạo directives và expressions |
| [Filter](https://docs.angularjs.org/guide/concepts#filter) | Bộ lọc, format lại dữ liệu hiển thị ra ngoài view người dùng |
| [View](https://docs.angularjs.org/guide/concepts#view) | Những gì người dùng có thể thấy được |
| [Data Binding](https://docs.angularjs.org/guide/concepts#databinding) | chuyển đổi dữ liệu giữa model và view |
| [Controller](https://docs.angularjs.org/guide/concepts#controller) | Lớp business logic phía sau views |
| [Dependency Injection](https://docs.angularjs.org/guide/concepts#di) | Liên kết, thống nhất các đối tượng và chức năng |
| [Injector](https://docs.angularjs.org/guide/concepts#injector) | Giống như thùng chứa các biển thể độc lập (hơi khó hiểu, nhưng mà sẽ rõ hơn khi đụng tới :p) |
| [Module](https://docs.angularjs.org/guide/concepts#module) | Cũng giống như các framework khác, module như thùng chứa các thứ như là controller, service, directives,… |
| [Service](https://docs.angularjs.org/guide/concepts#service) | Xây dựng service, mục đích là tái sử dụng lại các phương thức chung có thể sử dụng ở nhiều view hay controller khác nhau |

Các thành phần trong AngularJS

#### Ưu – nhược điểm của AngularJS

* Ưu điểm:
* AngularJS được phát triển bởi google, và là mã nguồn mở viết theo mô hình MVC.
* AngularJS cho phép tạo ra các ứng dụng một cách đơn giản, code sạch, dễ dàng hơn trong việc kiểm thử.
* Tương thích với hầu hết các trình duyệt trên các điện thoại thông minh.
* AngularJS sử dụng cơ chế data-binding tức là khi model thay đổi thì view cũng thay đổi theo và ngược lại.
* Nhược điểm:
* Không an toàn: Được phát triển từ javascript nên nó không an toàn, phía máy chủ phải thường xuyên xác nhận quyền để hệ thống chạy trơn tru.
* Phụ thuộc: Nếu người dùng vô hiệu hóa javascript thì coi vứt đi.

### NodeJS – Javascript Platform.

#### NodeJS là gì ?

Trích dẫn từ wikipedia:

**“Node.js** is an [open source](http://en.wikipedia.org/wiki/Open_source), [cross-platform](http://en.wikipedia.org/wiki/Cross-platform) [runtime environment](http://en.wikipedia.org/wiki/Runtime_system) for server-side and networking applications. Node.js applications are written in [JavaScript](http://en.wikipedia.org/wiki/JavaScript), and can be run within the Node.js runtime on [OS X](http://en.wikipedia.org/wiki/OS_X), [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), [Linux](http://en.wikipedia.org/wiki/Linux), [FreeBSD](http://en.wikipedia.org/wiki/FreeBSD), and [IBM i](http://en.wikipedia.org/wiki/IBM_i).”

**Nodejs** là một nền tảng được phát triển độc lập được xây dựng trên javascript runtime của chrome’s mà chúng ta có thể xây dựng được  ứng dụng mạng nhanh chóng và dễ dàng mở rộng.

#### Lịch sử phát triển.

Node.js được Ryan Dahl giới thiệu 2009 cho phép lập trình JavaScript phía server. Trước đây JavaScript chỉ được coi là một ngôn ngữ đơn giản tạo hiệu ứng hoạt hình, tương tác phía trình duyệt. Cho đến khi Google Chrome sử dụng V8 engine, một bộ dịch, thực thi mã JavaScript nhanh nhất thế giới tính đến nay. [V8](http://en.wikipedia.org/wiki/V8_%28JavaScript_engine%29) được viết bằng C/C++, mã nguồn mở, cho phép bên thứ 3 toàn quyền sử dụng, triển khai trên hệ điều hành khác nhau: Windows, Linux, Unix, MacOSX. Node.js được xây dựng trên V8 và thư viện [LibUV](https://github.com/libuv/libuv) cho phép lệnh thực thi hiện tại không ngăn các lệnh kế tiếp chạy luôn: cơ chế Non-Blocking I/O.

#### Đặc điểm của NodeJS.

[Cơ chế Non Blocking I/O](http://www.slideshare.net/marcusf/nonblocking-io-event-loops-and-nodejs) giúp máy chủ tạo ít process (hoặc thread) hơn, ít phải luân chuyển giữa các thread để xử lý các tác vụ. Máy chủ web Nginx cũng là một ví dụ tốt Non Blocking I/O chạy tốc độ vượt trội, tốn ít bộ nhớ so với máy chủ Apache sử dụng thread switching và thread pool để hứng, xử lý các yêu cầu.

Khi tiến hành nhiều phép đo đạc tốc độ trên môi trường máy ảo giống hệt nhau về cấu hình, chúng tôi nhận thấy một thực tế không thể chối cãi, web site viết trên Node.js + Express luôn chạy nhanh hơn 1.5 - 2.5 lần web site [PHP + Phalcon](http://phalconphp.com/en/) cùng cấu trúc, cùng hệ cơ sở dữ liệu, cùng cấu hình máy ảo. Trong nhóm các framework PHP, Phalcon luôn là đứng đầu bảng về tốc độ. Tốc độ xử lý của Phalcon trong vài trường hợp còn nhanh hơn cả HHVM. Trước đó tôi đã từng viết bài '[Tại sao chúng tôi chọn Phalcon](http://techmaster.vn/posts/5602/tai-sao-chung-toi-chon-phalcon)'

Có một số ý kiến cho rằng: hiện nay chi phí thuê máy chủ ngày càng rẻ. [Chỉ với 20$/tháng](https://www.digitalocean.com/?refcode=89a8b06156cb), đã có thể thuê một máy chủ ảo hóa 2 core 3.4 GHz, 2 GRAM DDR3, 40 G ổ cứng SSD, rẻ hơn so với năm 2012 khoảng 4 lần. Vậy nếu PHP là một ngôn ngữ phổ biến dễ học, có rất nhiều framework tuyệt vời như WordPress, Joomla, Drupal, Magento, có sẵn các theme giao diện tuyệt đẹp chỉ mất khoảng 40 - 120 USD, người dùng có mọi thứ từ một bộ CMS ổn định đến giao diện đồ họa cấu hình là chạy. Tuy nhiên trang web load chậm không phải lúc nào cũng thiếu RAM hay CPU Core. Nếu xử lý không đúng chỗ thì việc bổ xung RAM, hay CPU Core không cải thiện được triệt để.

#### Những ứng dụng nào có thể viết bằng nodejs?

* Các chương trình như các kênh chat với tốc độ thời gian thực runtime.
* Các chương trình upload file
* Các máy chủ quảng cáo
* Các dịch vụ đám mây
* Và bất kỳ ứng dụng dữ liệu thời gian thực nào.

### Module Socket.IO, Agenda, Express Framework.

#### SocketIO.

Socket là cách bạn tổ chức mô hình client-server để một trong hai bên luôn trong tình trạng sẵn sàng trả lời bên kia và ngược lại. Để đảm bảo việc này, kết nối giữa Client và Server phải ở trạng thái "keep-alive" và phải luôn xảy ra quá trình đồng bộ giữa Client-Server. Socket sẽ mang lại khả năng trả lời tức thì từ một trong hai bên khi bên kia đưa ra một sự kiện, thay vì phải thực thi lại một loạt các thủ tục kết nối phức tạp như trước, và ứng dụng của bạn sẽ trở thành ứng dụng thời gian thực ví dụ: Yahoo Messenger, Skype v.v... đều là các ứng dụng được xây dựng theo mô hình Socket.

#### Agenda.

* Khái niệm.

**Agenda** là một module khá thông dụng trong ứng dụng **NodeJS** cung cấp chức năng đặt lịch hệ thống.

* Nguyên lý hoạt động.

Cách thức hoạt động của **Agenda** là tổ chức các sự kiện vào hàng đợi theo thời gian thực.

* Cài đặt và sử dụng.
* Để cài đặt **Agenda** ta soạn lệnh cài đăt từ Window Command Line.

*Npm install agenda*

* Khai báo Agenda.

*Var Agenda = require(“agenda”);*

* Sử dụng.

*Agenda.define(“ten\_cong\_viec”, function(job, done){*

*Done();*

*})*

#### ExpressJS.

* Khái niệm.
* **Express** là một web application framework for node nó cung cấp cho chúng những rất nhiều tính năng mạnh mẽ trên nền tảng web và các ứng dụng di động.
* Express hỗ trợ các phương thức HTTP và middleware tạo ra 1 API rất mạnh mẽ và sử dụng dễ dàng hơn.Khi mới tiếp cận với Express mình thực sự bị cuốn hút bởi các API của nó,từ cách sử dụng route,template…đều khá dễ tùy chọn và làm việc.
* Cài đặt và sử dụng.
* Để cài đặt **ExpressJS** ta soạn lệnh cài đặt từ Window Command Line

*Npm install express*

* Khai báo.

*Var express = require(“express”);*

*Var app = express();*

*App.listen(PORT);*

* Sử dụng.

*App.route(“/route”, function(request, response){*

*//*

*})*

# Xây dựng chương trình

## Khảo sát

### Khảo sát đề tài.

* Cơ bản về luật giao thông đường bộ.
* Hoạt động giao thông đường bộ phải bảo đảm thông suốt, trật tự, an toàn, hiệu quả; góp phần phát triển kinh tế - xã hội, bảo đảm quốc phòng, an ninh và bảo vệ môi trường.
* Phát triển giao thông đường bộ theo quy hoạch, từng bước hiện đại và đồng bộ; gắn kết phương thức vận tải đường bộ với các phương thức vận tải khác.
* Quản lý hoạt động giao thông đường bộ được thực hiện thống nhất trên cơ sở phân công, phân cấp trách nhiệm, quyền hạn cụ thể, đồng thời có sự phối hợp chặt chẽ giữa các bộ, ngành và chính quyền địa phương các cấp.
* Bảo đảm trật tự, an toàn giao thông đường bộ là trách nhiệm của cơ quan, tổ chức, cá nhân.
* Người tham gia giao thông phải có ý thức tự giác, nghiêm chỉnh chấp hành quy tắc giao thông, giữ gìn an toàn cho mình và cho người khác. Chủ phương tiện và người điều khiển phương tiện phải chịu trách nhiệm trước pháp luật về việc bảo đảm an toàn của phương tiện tham gia giao thông đường bộ.
* Mọi hành vi vi phạm pháp luật giao thông đường bộ phải được phát hiện, ngăn chặn kịp thời, xử lý nghiêm minh, đúng pháp luật.

### Yêu cầu về nghiệp vụ.

* Phân tích thiết kế hệ thống của Ứng dụng.
* Đảm bảo chạy được trên đa nền tảng Web và Mobile sử dụng hệ điều hành Android.
* Hệ thống câu hỏi đầy đủ,đa dạng phù hợp với yêu cầu sát hạch bằng lái xe của cơ quan chức năng.
* Hệ thống chạy ổn định, giao diện thân thiện, dễ sử dụng, phù hợp với nhiều đối tượng trong xã hội.
* Dung lượng ứng dụng nhỏ để thuận tiện cho việc download và cài đặt trên Mobile sử dụng hệ điều hành Android.
* Cập nhật các tin tức cần thiết về luật giao thông và các kỳ sát hạch trên trang chủ.

### Các chức năng

* Chức năng của Quản trị Viên.

|  |  |
| --- | --- |
| *Chức năng* | *Ca sử dụng nguồn* |
| Quản lý đề thi, câu hỏi và đáp án.  Quản lý bài viết  Quản lý thông tin thành viên  Quản lý comment trong bài viết | Đăng nhập |
| Khôi phục mật khẩu | Khôi phục mật khẩu |
| Thêm đề thi  Sửa đề thi  Xóa đề thi  Them câu hỏi  Sửa câu hỏi  Xóa câu hỏi  Them đáp án  Xóa đáp án. | Quản lý đề thi, câu hỏi và đáp án |
| Thêm bài viết  Sửa bài viết  Xóa bài viết | Quản lý bài viết |
| Xem thông tin thành viên  Sửa thông tin thành viên  Xóa thành viên | Quản lý thành viên |
| Duyệt Comment của bài viết | Quản lý báo cáo,ý kiến đóng góp |

Chức năng của quản trị viên.

* Chức năng của Người dùng:

|  |  |
| --- | --- |
| *Chức năng* | *Ca sử dụng nguồn* |
| Đăng ký | Đăng ký |
| Đăng nhập  Dự thi | Đăng nhập |
| Xem danh mục bài viết |  |
| Xem bài viết |
| Xem chi tiết bài viết |
| Xem thông tin thành viên |
| Tìm kiếm bài viết, danh mục, từ trong bài viết |
| Comment bài viết |
| Gửi thư |
|  |

Chức năng người dùng.

## Phân tích thiết kế hệ thống

### Các tác nhân của hệ thống

|  |  |
| --- | --- |
| *Tác nhân* | *Mô tả* |
| Khách | là khách vãng lai chỉ vào hệ thống xem, tìm kiếm. |
| Người dùng | là những người có tài khoản trên hệ thống (là thành viên) có thể bình luận về bài viết và dự thi. |
| Quản trị viên | là người có quyền cao nhất trong hệ thống, quản lý hệ thống, quản lý thành viên,quản lý bài viết,đề thi,câu hỏi. |

Tác nhân của hệ thống.

### Các ca sử dụng của hệ thống

#### Biểu đồ Usecase chính



Biểu đồ ca sử dụng chính

#### Biểu đồ Usecase Khách



Biểu đồ ca sử dụng khách hàng

#### Biểu đồ Usecase Quản trị hệ thống



Biểu đồ ca sử dụng của Quản trị hệ thống website

#### Biểu đồ Usecase thành viên



Biểu đồ Use Case thành viên

### Phân tích từng ca sử dụng

#### Ca sử dụng Đăng nhập

* 1. Đặc tả.

-Mục đích:

+Giúp quản trị viên,thành viên truy cập hệ thống quản lý website.

+Giúp khách có thêm chức năng người dùng.

-Mô tả tóm tắt:

Khách truy cập vào trang chủ của website,click vào nút đăng nhập.

|  |  |
| --- | --- |
| Người dùng | Hệ thống |
| 1.Truy cập trang đăng nhập | 2. Hiển thị trang đăng nhập cho phép khách điền tên đăng nhập và mật khẩu để đăng nhập |
| 3. điền tên đăng nhập và mật khẩu sau đó nhấn đăng nhập. | 4. Kiểm tra tính hợp lệ của tên đăng nhập và mật khẩu.sau đó hiển thị thông báo và chuyển trang. |

Đặc tả ca sử dụng đăng nhập.

* 1. Biểu đồ hoạt động.



Biểu đồ hoạt động ca sử dụng đăng nhập

* 1. Biểu đồ lớp tham gia ca sử dụng



.Biểu đồ lớp tham gia ca sử dụng đăng nhập

* 1. Biểu đồ trình tự



Biểu đồ trình tự ca sử dụng đăng nhập.

#### Ca sử dụng quản lý Thành viên

Mục đích: Giúp quản trị viên quản lý thông tin thành viên trong website.

* 1. Đặc tả
* Luồng sự kiện chính.

|  |  |
| --- | --- |
| Quản trị viên | Hệ thống |
| 1.Chọn danh mục quản lý thành viên | 2.Hiển thị danh sách danh sách thành viên |
| 3.Chọn các chức năng để quản lý thành viên:  -Sửa thông tin thành viên.  -Xóa thành viên. | 4.Thực hiện công việc. |

Đặc tả ca sử dụng quản lý thành viên.

* Xóa thành viên

|  |  |
| --- | --- |
| Quản trị viên | Hệ thống |
| 1.Nhấn vào nút xóa thành viên tương ứng với thành viên cần xóa | 2.Hiển thị hộp thoại xác nhận xóa thành viên. |
| 3.Nhấn xác nhận | 4.Xóa thành viên đã chọn và quay lại danh sách thành viên |

Đặc tả ca sử dụng xóa thành viên.

* Sửa thông tin thành viên.

|  |  |
| --- | --- |
| Quản trị viên | Hệ thống |
| 1.Nhấn nút sửa thông tin thành viên | 2.Hiển thị thông tin của thành viên đã chọn vào form cho phép quản trị viên sửa thông tin của thành viên. |
| 3.Sửa thông tin thành viên và nhấn cập nhật. | 4.Kiểm tra thông tin đã sửa có hợp lệ hay không.Nếu hợp lệ thì thông báo thành công.Ngược lại thông báo lỗi. |

Đặc tả ca sử dụng Quản lý thành viên – hành động sửa thông tin thành viên.

1. Biểu đồ hoạt động

* Xóa thành viên



Biểu đồ hoạt động ca sử dụng Quản lý thành viên – hành động xóa thành viên

* Sửa thông tin thành viên.



Biểu đồ hoạt động ca sử dụng Quản lý thành viên – hành động sửa thông tin thành viên

1. Biểu đồ lớp tham gia ca sử dụng



Biểu đồ lớp tham gia ca sử dụng Quản lý thành viên

1. Biểu đồ trình tự

* Xóa thành viên.



Biểu đồ trình tự ca sử dụng Quản lý thành viên – hành động xóa thành viên

* Sửa thông tin thành viên.



Biểu đồ trình tự ca sử dụng Quản lý thành viên – hành động sửa thành viên

#### Ca sử dụng quản lý danh mục bài viết

* 1. Đặc tả ca sử dụng
* Mục đích

Giúp quản trị viên,nhân viên có thể quản lý danh mục bài viêt.

* Luồng sự kiện chính

|  |  |
| --- | --- |
| Quản trị viên | Hệ thống |
| 1.Chọn quản lý danh mục bài viết. | 2.Hiển thị danh sách mục bài viết. |
| 3.Chọn các chức năng muốn thực hiện:  -Thêm danh mục.  -Sửa danh mục.  -Xóa danh mục. | 4.Thực thi công việc. |

Đặc tả ca sử dụng quản lý danh mục bài viết.

* Thêm danh mục bài viết

|  |  |
| --- | --- |
| Quản trị viên | Hệ thống |
| 1.Chọn them danh mục bài viết | 2.Hiển thị form cho phép điền thông tin của tin vào form như tiêu đề,mô tả,nội dung,ảnh,… |
| 3.Điền thông tin và nhấn thêm | 4.Kiểm tra xem danh mục có bị trùng tên chưa,hoặc các thông tin điền vào có hợp lệ không.Nếu hợp lệ thì tiến hành thêm danh mục bài viết vào csdl,ngược lại thì đưa ra thông báo lỗi. |

Đặc tả ca sử dụng quản lý danh mục bài viết – hành động thêm danh mục.

* Sửa danh mục bài viết.

|  |  |
| --- | --- |
| Quản trị viên | Hệ thống |
| 1.Chọn quản lý danh mục bài viết | 2.Hiển thị danh sách danh mục bài viết. |
| 3.Chọn danh mục bài viết muốn sửa. | 4.Hiển thị thông tin của danh mục bài viết đã chọn vào form và cho phép chỉnh sửa các thông tin của tin tức . |
| 5.Điền thông tin và nhấn cập nhật. | 6.Kiểm tra xem thông tin cập nhật có hợp lệ không.nếu hợp lệ thì thông báo cập nhật thành công,ngược lại thì thông báo lỗi. |

Đặc tả ca sử dụng quản lý danh mục bài viết – hành động sửa danh mục.

* Xóa danh mục bài viết.

|  |  |
| --- | --- |
| Quản trị viên | Hệ thống |
| 1.Chọn danh sách mục bài viết. | 2.Hiển thị danh sách danh mục bài viết gồm các thông tin khái quát về danh mục . |
| 3.Nhấn xóa tương ứng với danh mục bài viết. | 4.Hiển thị hộp thoại yêu cầu xác nhận. |
| 5.Xác nhận | 6.Xóa danh mục bài viết kèm các bài viết trong danh mục. |

Đặc tả ca sử dụng quản lý danh mục bài viết – hành động xóa danh mục.

* 1. Biểu đồ hoạt động
* Thêm danh mục bài viết.



Biểu đồ hoạt động ca sử dụng Quản lý danh mục bài viết – hành động them danh mục.

* Sửa danh mục bài viết



Biểu đồ hoạt động ca sử dụng Quản lý danh mục bài viết – hành động sửa danh mục.

* Xóa danh muc bai viết



Biểu đồ hoạt động ca sử dụng Quản lý danh mục bài viết – hành động xóa danh mục.

* 1. Biểu đồ lớp tham gia ca sử dụng



Biểu đồ lớp tham gia ca sử dụng Quản lý danh mục bài viết.

* 1. Biểu đồ trình tự
* Thêm danh mục bài viết.



Biểu đồ trình tự ca sử dụng Quản lý danh mục bài viết – hành động them danh mục.

* Sửa danh mục bài viết.



Biểu đồ trình tự ca sử dụng Quản lý danh mục bài viết – hành động sửa danh mục.

* Xóa danh mục bài viết.



Biểu đồ trình tự ca sử dụng Quản lý danh mục bài viết – hành động xóa danh mục.

#### Ca sử dụng quản lý bài viết.

1. Đặc tả

* Mục đích :
* Giúp QTV quản lý các bài viết.
* Luồng sự kiện chính

|  |  |
| --- | --- |
| QTV | Hệ thống |
| 1.Chon chức năng quản lý bài viết | 2.Hiển thị danh sách bài viết. |
| 3.Chọn các chức năng cần quản lý:  -Thêm bài viết  -Sửa bài viết.  -Xóa bài viết. | 4.Thực hiện công việc. |

Đặc tả ca sử dụng quản lý bài viết.

* Thêm bài viết.

|  |  |
| --- | --- |
| QTV | Hệ thồng |
| 1.Chọn them danh mục bài viết | 2.Hiển thị trang them danh mục bài viết |
| 3.Điền thông tin của danh mục bài viết. | 4. |

Đặc tả ca sử dụng quản lý bài viết- hành động thêm bài viết.

* Sửa bài viết.

|  |  |
| --- | --- |
| QTV | Hệ thống |
| 1.Chọn bài viết cần sửa | 2.Hiển thị chi tiết bài viết cần sửa |
| 3.Sửa các thông tin và nhấn xác nhận | 4.Kiểm tra hợp lệ của các trường.hợp lệ thì cập nhận bài viết hoặc lỗi thì in ra thông báo. |

Đặc tả ca sử dụng quản lý bài viết- hành động sửa bài viết.

* Xóa bài viết.

|  |  |
| --- | --- |
| QTV | Hệ thống |
| 1.Nhấn nút xóa tương ứng với bài viết cần xóa | 2.Hiển thị hộp thoại cho phép nhận lý do xóa bài và yêu cầu xác nhận |
| 3.Nhập lý do xóa bài và xác nhận. | 4.Gửi mail có nội dung là lý do xóa bài đến người viết bài và xóa bài. |

Đặc tả ca sử dụng quản lý bài viết- hành động xóa bài viết.

1. Biểu đồ hoạt động

* Thêm bài viết.



Biểu đồ hoạt động ca sử dụng Quản lý bài viết – hành động them bài viết.

* Sửa bài viết.



Biểu đồ hoạt động ca sử dụng Quản lý bài viết – hành động sửa bài viết.

* Xóa bài viết.



Biểu đồ hoạt động ca sử dụng Quản lý bài viết – hành động xóa bài viết.

1. Biểu đồ lớp tham gia ca sử dụng



*Hình 3.23.Biểu đồ lớp tham gia ca sử dụng Quản lý bài viết.*

1. Biểu đồ trình tự

* Thêm bài viết.



Biểu đồ trình tự ca sử dụng Quản lý bài viết – hành động them bài viết

* Sửa bài viết.



Biểu đồ trình tự ca sử dụng Quản lý bài viết – hành động sửa bài viết

* Xóa bài viết.



Biểu đồ trình tự ca sử dụng Quản lý bài viết – hành động xóa bài viết

#### Ca sử dụng đăng ký thành viên.

1. Đặc tả

* Mục đích:
* Giúp khách truy cập trở thành thành viên của website.
* Luồng sự kiện chính

|  |  |
| --- | --- |
| Khách | Hệ thống |
| 1.Truy cập trang chủ | 2.Hiển thị trang chủ |
| 3.Chọn chức năng đăng ký thành viên | 4.Hiển thị trang đăng ký thành viên |
| 5.Điền thông tin vào form đăng ký thành viên, cần điều chính xác địa chỉ email để nhận được đường dẫn kích hoạt tài khoản.sau đó nhấn đăng ký. | 6.Gửi đường dẫn kích hoạt tài khoản đến địa chỉ email đã nhập trong form đăng ký.đường dẫn kích hoạt tồn tại trong 24h.Quá 24h mà tài khoản chưa được kích hoạt thì xóa tài khoản. |
| 7.Kiểm tra email và kích vào đường dẫn kích hoạt tài khoản. | 8.Kích hoạt tài khoản. |

Đặc tả ca sử dụng đăng ký thành viên.

1. Biểu đồ hoạt động



Biểu đồ hoạt động ca sử dụng đăng ký thành viên

1. Biểu đồ lớp tham gia ca sử dụng



Biểu đồ lớp tham gia ca sử dụng đăng ký thành viên

1. Biểu đồ trình tự



Biểu đồ trinh tự ca sử dụng đăng ký thành viên

#### Ca sử dụng quản lý đề thi.

* 1. Đặc tả ca sử dụng.
* Mục đích:
* Giúp cho các thành viên tạo cho mình 1 nhóm để cùng thảo luận về 1 vấn đề hay 1 chủ đề.
* Giúp QTV quản lý dễ dàng các thành viên.
* Luồng sự kiện chính.

|  |  |
| --- | --- |
| QTV | Hệ thống |
| 1.Truy cập bảng điều khiển | 2.Hiển thị bảng điều khiển |
| 3.Chọn chức năng quản lý đề thi. | 4.Hiển thị danh sách các đề thi. |
| 5.Chọn các chức năng:  -Thêm đề thi.  -Sửa đề thi  -Xóa đề thi | 6.Thực hiện công việc |

Đặc tả ca sử dụng quản lý đề thi.

* Luồng sự kiện rẽ nhánh.
* Thêm đề thi.

|  |  |
| --- | --- |
| QTV | Hệ thống |
| 1.Chọn chức năng them đề thi | 2.Hiện thị form them đề thi. |
| 3.Điền thông tin đề thi và nhấn them | 4.Kiểm tra hợp lệ.Nếu hợp lệ thì thông báo thành công và tạo nhóm sau đó chuyển đến trang quản lý câu hỏi của đề thi.Ko hợp lệ thì thông báo lỗi. |
| 5.Nhập câu hỏi cho đề thi. | 6.Kiểm tra hợp lệ.Thêm câu hỏi vào đề thi và chuyển đến trang them đáp án cho đề thi. |
| 7.Nhập đáp án cho câu hỏi. | 7.Thêm đáp án cho câu hỏi. |

Đặc tả ca sử dụng quản lý đề thi - hành động thêm đề thi.

* Sửa đề thi.

|  |  |
| --- | --- |
| QTV | Hệ thống |
| 1.Chọn đề thi cần sửa. | 2.Hiển thị form sửa đề thi, danh sách câu hỏi và đáp án của đề thi. |
| 3.Sửa đề thi, câu hỏi và đáp án và nhấn cập nhật. | 4. Kiểm tra hợp lệ.Nếu hợp lệ thì thông báo thành công và cập nhật nhóm.Ko hợp lệ thì thông báo lỗi. |

Đặc tả ca sử dụng quản lý đề thi – hành động sửa đề thi.

* Xóa nhóm.

|  |  |
| --- | --- |
| QTV | Hệ thống |
| 1.Chọn đề cần xóa | 2.Hiển thị hộp thoại yêu cầu xác nhận |
| 3.Nhấn xác nhận | 4.Xóa đề và câu hỏi đáp án của đề. |

Đặc tả ca sử dụng quản lý đề thi – hành động xóa đề thi.

* 1. Biểu đồ hoạt động.
* Thêm đề thi.



Biểu đồ hoạt động ca sử dụng Quản lý đề thi – hành động them đề thi.

* Sửa đề thi.



Biểu đồ hoạt động ca sử dụng Quản lý nhóm – hành động sửa đề thi.

* Xóa đề thi.



Biểu đồ hoạt động ca sử dụng Quản lý đề thi – hành động xóa đề thi.

c.Biểu đồ tham gia ca sử dụng.



Biểu đồ lớp tham gia ca sử dụng Quản lý đề thi.

d.Biểu đồ trình tự

* Thêm đề thi.



Biểu đồ trình tự ca sử dụng Quản lý đề thi – hành động them đề thi.

* Sửa đề thi

Biểu đồ trình tự ca sử dụng Quản lý đề thi – hành động sửa đề thi.

* Xóa nhóm.



Biểu đồ trình tự ca sử dụng Quản lý đề thi – hành động xóa đề thi.

#### Ca sử dụng quản lý comment.

1. Đặc tả ca sử dụng.

* Mục đích:

-Giúp thành viên tham đưa ra ý kiếm về bài viết.

-Giúp QTV quản lý ý thức của cá thành viên trong diễn đàn.

* Luồng sự kiện chính.

Ca sử dụng nguồn: Đăng nhập.

|  |  |
| --- | --- |
| Thành viên | Hệ thống |
| 1.Truy cập bảng điều khiển | 2.Hiển thị bảng điều khiển |
| 3.Truy chức năng quản lý comment | 4.Hiển thị danh sách comment |
| 5.Duyệt comment | 6. Nếu comment hợp lệ không vi phạm thuần phong mỹ tục thì được duyệt.Ngược lại thì xóa comment và gửi mail đển thành viên về lý do xóa comment và cảnh báo vi phạm. |

Đặc tả ca sử dụng quản lý comment.

1. Biểu đồ hoạt động.



Biểu đồ hoạt động ca sử dụng quản lý comment.

1. Biểu đồ lớp tham gia ca sử dụng.



Biểu đồ lớp tham gia ca sử dụng quản lý comment.

1. Biểu đồ trình tự quản lý comment.



Biểu đồ trình tự ca sử dụng quản lý comment.

#### Ca sử dụng làm bài thi.

1. Đặc tả ca sử dụng.

|  |  |
| --- | --- |
| Tác nhân – Thành viên | Hệ thống |
| 1.Truy cập ứng dụng từ trang chủ hoặc trên mobile. | 2.Hiển thị giao diện ứng dụng và danh sách loại thi. |
| 3.Chọn loại bằng lái xe muốn thi | 4.Hiển thị danh sách đề thi của loại bằng lái xe. |
| 5.Chọn đề thi | 6.Hiển thị phiếu thi cho phép nhập thông tin dự thi. |
| 7.Nhập thông tin dự thi và nhấn vào thi. | 8.Hiển thị danh sách câu hỏi. |
| 9.Nhấn bắt đầu thi. | 10.Bắt đầu đếm lùi thời gian 15 phút làm bài,hiển thị câu hỏi thứ nhất. |
| 11.Chọn đáp án cho các câu hỏi.nếu làm xong có thể nhấn nộp bài. | 12.Khi thành viên nộp bài hoặc hết giờ thi.hệ thống tính điểm.Thông báo kết quả cùng bài thi cho thành viên biết. |

Đặc tả ca sử dụng làm bài thi.

1. Biểu đồ hoạt động.



Biểu đồ trình tự ca sử dụng làm bài thi.

1. Biểu đồ lớp tham gia ca sử dụng.



Biểu đồ lớp tham gia ca sử dụng làm bài thi.

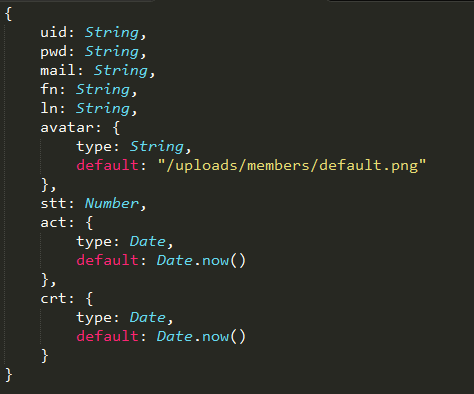
1. Biểu đồ trình tự.



Biểu đồ trình tự ca sử dụng làm bài thi.

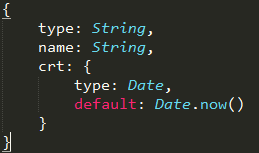
### Thiết kế cơ sở dữ liệu

1. Collection Member – lưu trữ thông tin thành viên.



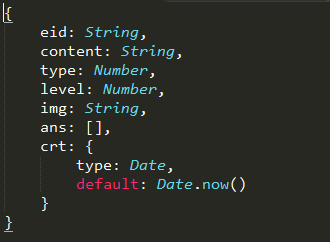
Collection Member.

1. Collection Examp – lưu trữ dữ liệu đề thi.



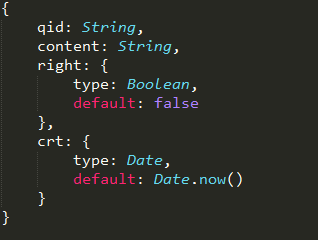
Collection Example.

1. Collection Question – lưu trữ dữ liệu câu hỏi.



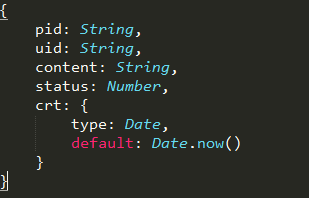
Collection Question.

1. Collecttion Answer – lưu trữ dữ liệu đáp án.



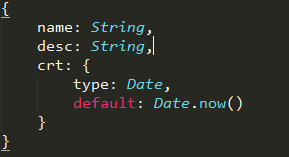
Collection Answer.

1. Collection Comment – lưu trữ dữ liệu comment.



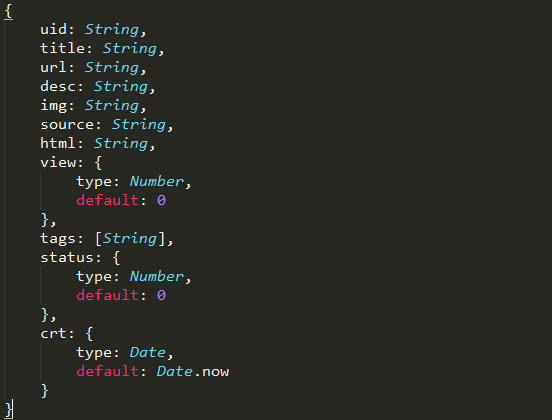
Collection Comment.

1. Collection TypeExamp – lưu trữ dữ liệu loại thi.



Collection TypeExamp.

1. Collection Post – lưu trữ bài viết.



Collection Post.

## Triển khai ứng dụng trên máy chủ CentOS và kết quả.

### Triển khai ứng dụng trên máy chủ CentOS.

B1.Cài đặt NodeJS và MongoDB trên máy chủ CentOS.

B2.Sử dụng GIT hoặc SVN để đẩy source lên server.

B3.Cài đặt package cần thiết của ứng dụng từ file package.json.

B4.Cài đặt PM2 module để duy trì ứng dụng trên server.

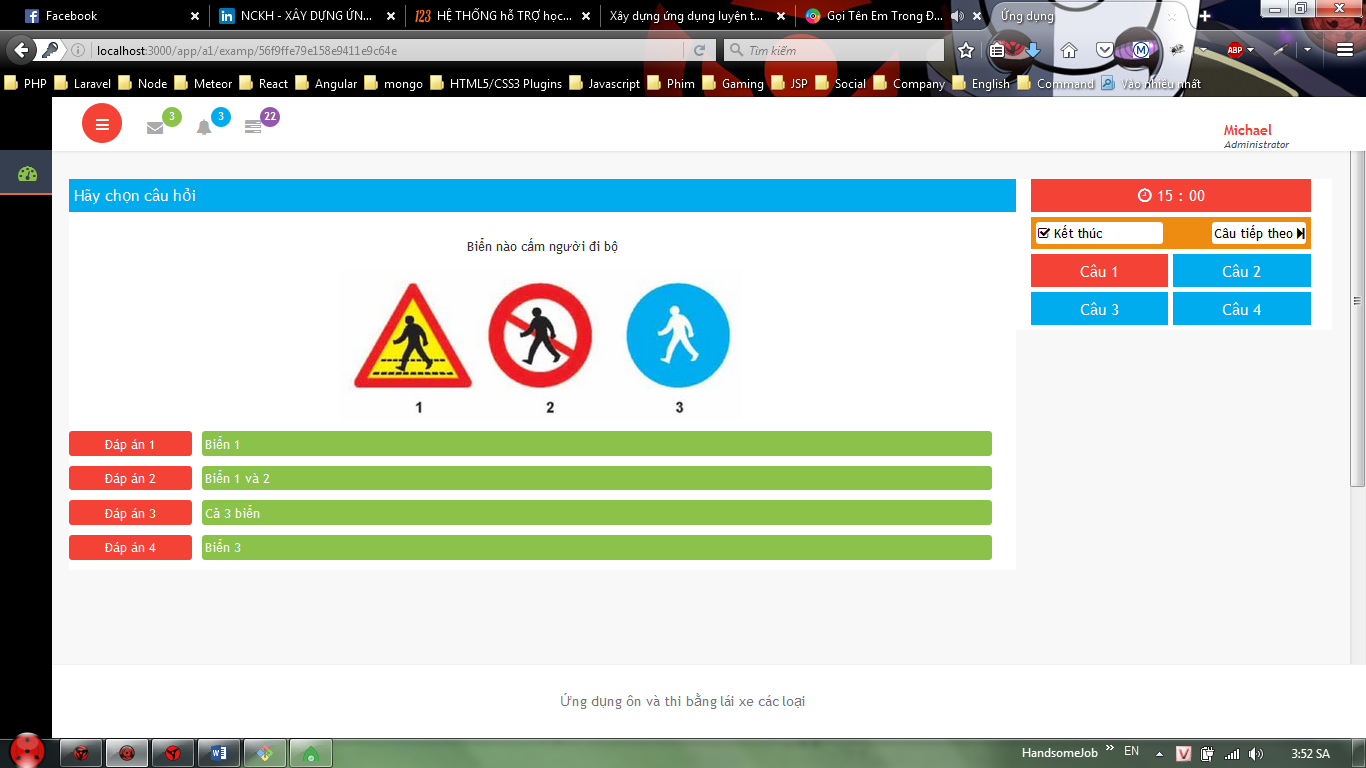
B5.Chạy ứng dụng.

### Kết quả.

* Ứng dụng trên nền web chạy rất nhanh và mượt.
* Tốc độ tải trang khá nhanh, vào khoảng ~0,6s cho trang tin tức.
* Trên lý thuyết, ứng dụng có thể cho phép > 10.000 truy cập 1 lúc và đáp ứng gần như ngay lập tức, độ trễ rất nhỏ không đáng kể.
* Ứng dụng trên Mobile hệ điều hành Android tuy có hiện tượng giật lag nhưng chạy vẫn ổn định, lấy dữ liệu câu hỏi đề thi về khá nhanh.

### Giao diện

* Giao diện chính của ứng dụng trên nền WEB.



Giao diện ứng dụng trên nền WEB.



Giao diện ứng dụng trên Mobile nền tảng Android.

# Tài liệu tham khảo

* Tài liệu Android chính thống của Google.

<http://developer.android.com/intl/vi/reference/org/w3c/dom/Document.html>

* Tài liệu AngularJS chính thống của Google.

<https://docs.angularjs.org/tutorial>

* Tài liệu NodeJS chính thống.

<https://nodejs.org/en/docs/>

* Tài liệu Bootstrap 3 chính thống.

<http://getbootstrap.com/components/>