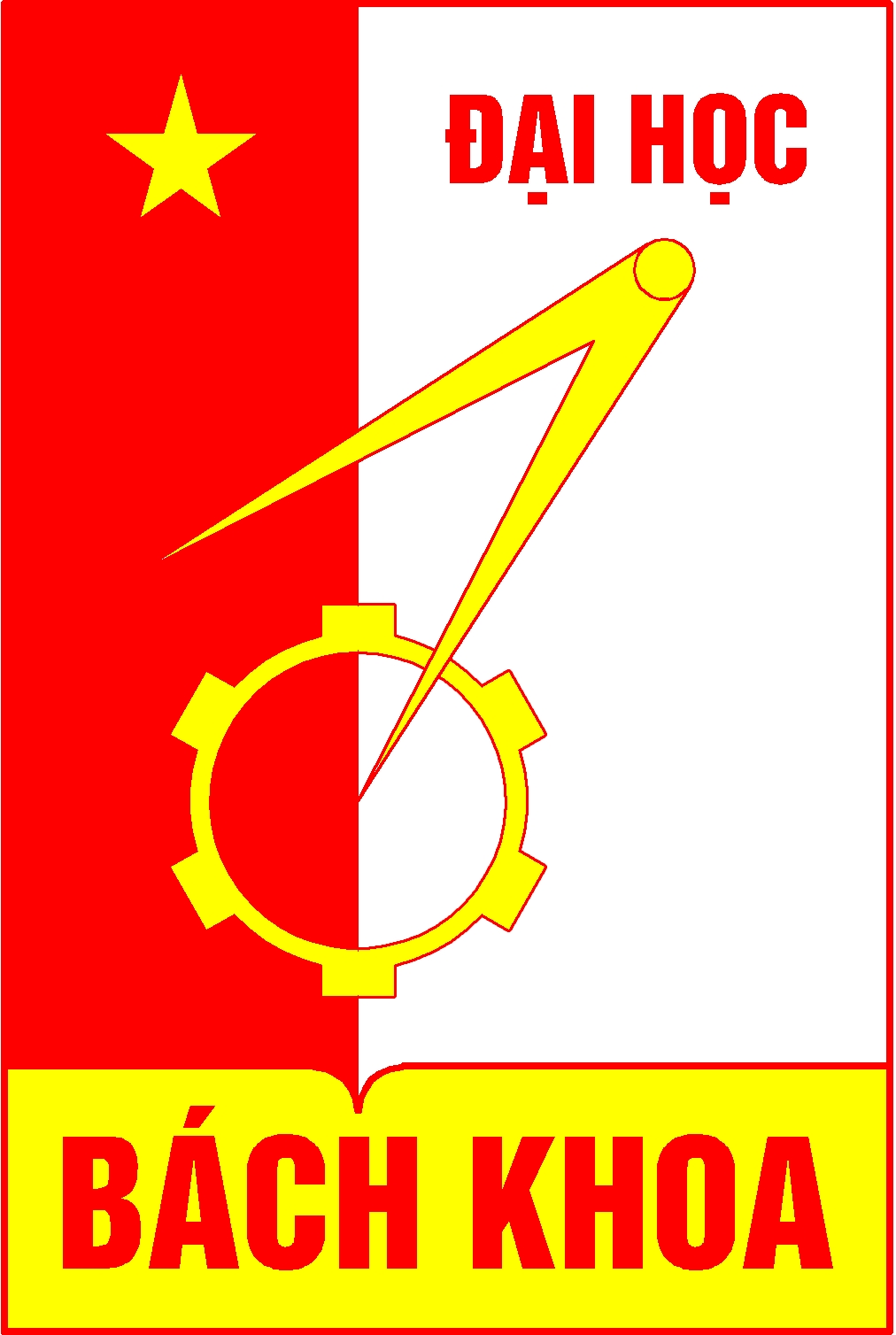
Hà Nội, Tháng Năm 2015

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**BÁO CÁO WEB THẾ HỆ MỚI**

Đề tài: XÂY DỰNG TRANG WEB GIỚI THIỆU VÀ BÁN ĐIỆN THOẠI TRỰC TUYẾN

|  |  |
| --- | --- |
| Giảng viên hướng dẫn | TS. Phạm Huy Hoàng |
| **Sinh viên thực hiện** |  |
|  | **Nguyễn Gia Tuyến 20122726**  **Nguyễn Thành Lợi 20114633**  **Bùi Thái Bình 20114628** |

Mục lục

[**CÁC HÌNH ẢNH SỬ DỤNG** 3](#_Toc420416821)

[**GIỚI THIỆU** 4](#_Toc420416822)

[**CHƯƠNG 1: GIỚI THIỆU VỀ CÁC CÔNG NGHỆ SỬ DỤNG** 5](#_Toc420416823)

[**I.** **Frontend: AngularJS** 5](#_Toc420416824)

[1. Kiến trúc MVC 5](#_Toc420416825)

[2. Two-way binding 6](#_Toc420416826)

[3. Dynamic templates 7](#_Toc420416827)

[4. Expressions 7](#_Toc420416828)

[5. Scopes 7](#_Toc420416829)

[6. Dependency injection 8](#_Toc420416830)

[7. Directives 8](#_Toc420416831)

[8. Routing 8](#_Toc420416832)

[9. Services 8](#_Toc420416833)

[10. Filters 8](#_Toc420416834)

[**II.** **Backend: Play Framework** 8](#_Toc420416835)

[1. Kiến trúc 8](#_Toc420416836)

[2. Full stack framework. 9](#_Toc420416837)

[3. RESTFul 10](#_Toc420416838)

[**CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG** 12](#_Toc420416839)

[**I.** **Các chức năng của website** 12](#_Toc420416840)

[1. Chức năng phía frontend 12](#_Toc420416841)

[2. Chức năng phía backend 13](#_Toc420416842)

[**II.** **Biểu đồ USECASE** 15](#_Toc420416843)

[1. Usecase: Khách hàng 15](#_Toc420416844)

[2. Use case: Hệ thống 15](#_Toc420416845)

[**CHƯƠNG 3: THIẾT KẾ GIAO DIỆN VÀ CÀI ĐẶT CHƯƠNG TRÌNH** 20](#_Toc420416846)

[**I.** **Giao diện cài đặt** 20](#_Toc420416847)

[1. Nền tảng sử dụng. 20](#_Toc420416848)

[2. Hướng dẫn cài đặt. 20](#_Toc420416849)

[**II.** **Sản phẩm** 20](#_Toc420416850)

[**TÀI LIỆU THAM KHẢO** 22](#_Toc420416851)

# **CÁC HÌNH ẢNH SỬ DỤNG**

[Ảnh 1: Kiến trúc MVW 6](#_Toc420417023)

[Ảnh 2: Two-way data binding 7](#_Toc420417024)

[Ảnh 3: Kiến trúc thực hiện của Play Framework 9](#_Toc420417025)

[Ảnh 4: Full stack thư viện trong java 10](#_Toc420417026)

[Ảnh 5: Mô hình MVC trong Play Framework 10](#_Toc420417027)

[Ảnh 6: URL trong router của Play Framework 11](#_Toc420417028)

[Ảnh 7: Chức năng phía front-end. 12](#_Toc420417029)

[Ảnh 8: Chức năng phía back-end. 14](#_Toc420417030)

[Ảnh 9: Usecase khách hàng. 15](#_Toc420417031)

[Ảnh 10: Usecase hệ thống. 16](#_Toc420417032)

[Ảnh 11: Biểu đồ hoạt động tạo đơn hàng. 17](#_Toc420417033)

[Ảnh 12: Biểu đồ hoạt động thanh toán. 18](#_Toc420417034)

[Ảnh 13: Biểu đồ lớp. 19](#_Toc420417035)

[Ảnh 14: Trang khách hàng 20](#_Toc420417036)

[Ảnh 15: Trang quản trị 21](#_Toc420417037)

# **GIỚI THIỆU**

Hiện nay, các công nghệ tiên tiến phát triển ngày càng mạnh mẽ và được ứng dụng ngày càng nhiều vào các lĩnh vực kinh tế, sản xuất cũng như đời sống thường nhật của con người. Một điểm tiêu biểu trong việc phát triển các công nghệ đó phải kể đến việc ứng dụng công nghệ thông tin vào hầu khắp các hoạt động.

Nhờ đó, các công việc được thực hiện nhanh, chính xác và đạt kết quả cao hơn rất nhiều. Khi xã hội ngày càng phát triển, mức sống của người dân được nâng cao thu nhập kinh tế ngày càng được cải thiện thì chiếc điện thoại di động không còn trở nên xa lạ với mọi người nữa mà ngược lại nó là một vật dụng không thể thiếu đối với người dân hiện nay. Hầu hết mỗi người đều trang bị cho mình một chiếc điện thoại phù hợp với nhu cầu và túi tiền của mình. Tuy nhiên, với cuộc sống ngày càng bận rộn như hiện nay thì việc muốn mua một chiếc điện thoại mình ưa thích thì người tiêu dùng phải đến tận cửa hàng để chọn lựa vì thế sẽ mất khá nhiều thời gian và công sức.

Cùng với các lý do nêu trên, qua tìm hiểu nhóm đồ án được biết việc ứng dụng bán điện thoại di động trực tuyến sẽ giúp cho khách hàng giảm bớt được thời gian và công sức phải đến tận cửa hàng để mua. Muốn lựa chọn cho mình một chiếc điện thoại ưng ý phù hợp với túi tiền thì khách hàng chỉ cần ngồi bên chiếc máy tính có nối mạng internet là có thế mua được mặt hàng điện thoại mình cần.

Do đó nhóm em chọn thực hiện đề tài “ Xây dựng Website bán điện thoại di động trực tuyến cho công ty T- Mobile ”

Phần mềm được xây dựng với định hướng giúp việc mua sắm điện thoại của khách hàng dễ dàng và nhanh gọn hơn, không mất thời gian và công sức.

Qua quá trình khảo sát thực tế, tìm hiểu và phân tích thiết kế nhóm đề tài đưa ra bản báo cáo gồm các phần sau để giúp người đọc hiểu rõ về chức năng của Website.

# **CHƯƠNG 1: GIỚI THIỆU VỀ CÁC CÔNG NGHỆ SỬ DỤNG**

1. **Frontend: AngularJS**

Angular là framework javascript mạnh mẽ. Angular tăng cường HTML cho các ứng dụng web. Nó có chức năng để giảm bớt quá trình phát triển ứng dụng web.

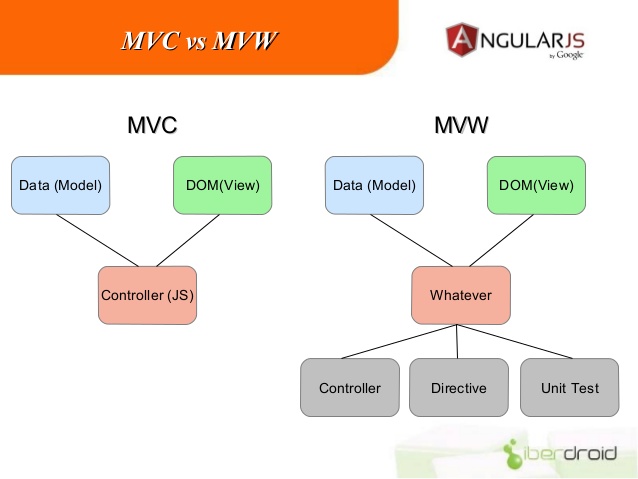
Là một bộ thư viện javascript do Google phát triển để xây dựng một project Single Page Application (SPA). SPA là một web app hay website hiển thị vừa vặn trên một mặt trang web, là ứng dụng bên trong trình duyệt, không yêu cầu phải reload lại toàn bộ trang web mỗi lần sử dụng.

Mục tiêu: xây dựng các ứng dụng nghiệp vụ theo kiểu thêm, đọc, sửa, xóa, và xây dựng với tư tưởng hướng đến việc testing.

Các tính năng chính: Kiến trúc MVC, Two-way binding, Dynamic templates, Expressions, Scopes, Dependency injection, Directives, Routing, Services, Filters, …

1. Kiến trúc MVC

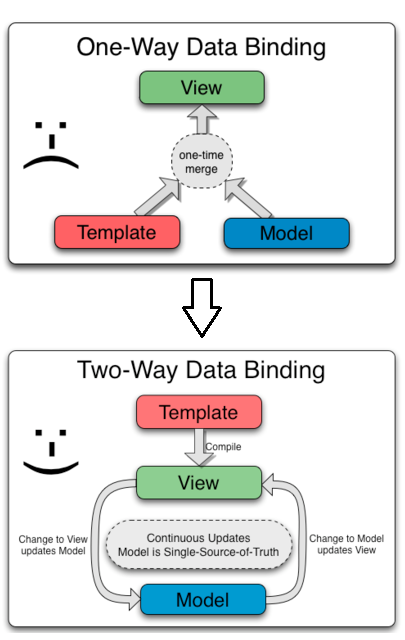
* Phân tách ứng dụng ra các thành phần presentation data, và logic.
* Sau nhiều lần chỉnh sửa và cải tiến API thì kiến trúc của AngularJS đã trở nên gần hơn với MVW (Module View Whatever).



Ảnh 1: Kiến trúc MVW

1. Two-way binding

* Data-binding trong Angular web apps là việc tự động đồng bộ hóa dữ liệu giữa model và các thành phần của views.
* View luôn luôn là thành phần hiển thị ra ngoài của Model. Khi Model thay đổi, View cũng thay đổi theo, và ngược lại.



Ảnh 2: Two-way data binding

1. Dynamic templates

* AngularJs template là 1 đặc tả dạng declarative, cùng với thông tin từ model và controller, trở thành rendered view mà user thấy trên mặt browser. Các thành phần AngularJs và các thuộc tính giúp angular thêm các hành vi và biến đổi template DOM thành dynamic view DOM.

1. Expressions

* Là các snippets giống javascipt được đặt trong bindings. Ví dụ: {{expression}}.
* Được xử lý bởi $parse service.

1. Scopes

* Là Object trong application model, là phần gắn kết giửa view và controller, tự động đồng bộ dữ liệu giữa Model và View.
* Scope cung cấp các APIs để truyền bất kỳ thay đổi nào của Model tới View: $apply.

1. Dependency injection

* AngularJS sử dụng các Dependency injection để tách biệt các modules, các dependency được đưa vào tự động bởi framework.

1. Directives

* Là các tùy biến gắn vào các DOM để HTML biên dịch của AngularJS có thể thêm vào hoặc biến đổi DOM element.

1. Routing

* Dùng để tạo các app theo kiểu SPA.

1. Services

* Là các phần dùng lại được của business logic, được tạo dưới các singleton object.
* Là một phần của module và module đó là nhóm các chức năng.

1. Filters

* Dùng để định dạng dữ liệu hiển thị cho người dùng, hoặc để lọc những dữ liệu cần thiết mà người dùng mong muốn.

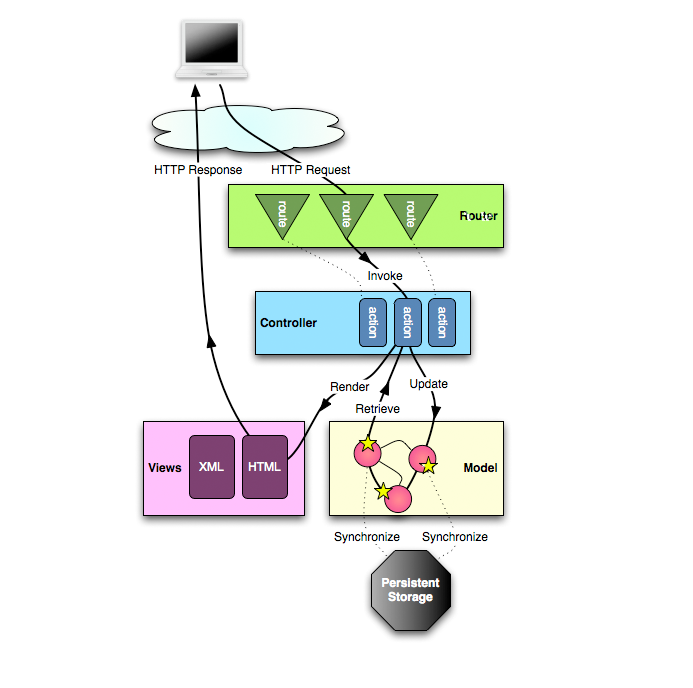
1. **Backend: Play Framework**

Play Framework là framework viết cho 2 ngôn ngữ Scala và Java.

Là Framework đầu tiên của Java xây dựng không dựa trên Servlet. Và có tính mở rộng cao, built on Akka, Real-time, Non-blocking I/O.

Các đặc trưng của Play: A full stack framework, RESTful, Stateless, …

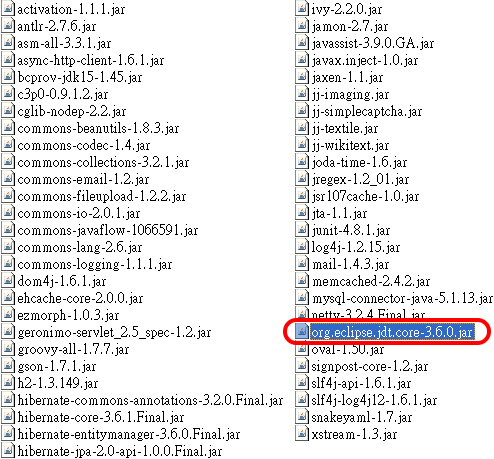
1. Kiến trúc



Ảnh 3: Kiến trúc thực hiện của Play Framework

1. Full stack framework.

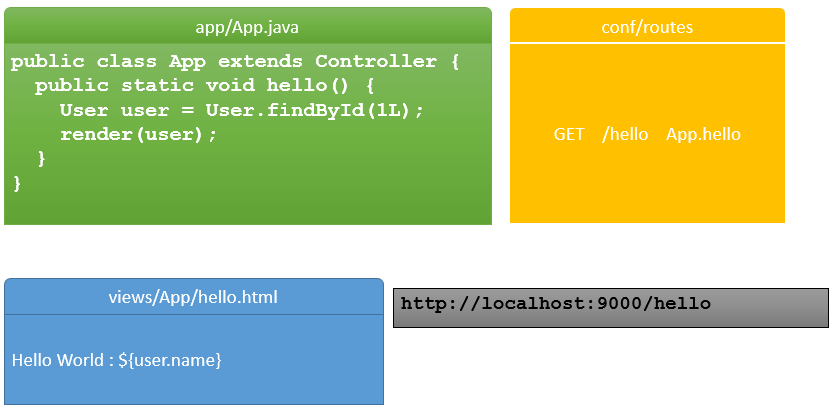
* Nhúng server bên trong các trình duyệt web nên có thể chạy mà không cần Tomcat, GlassFish hay JBoss…
* Hot reload như script (PHP, Ruby, Python).
* Cung cấp nhiều thư viện có sẵn của Java.



Ảnh 4: Full stack thư viện trong java

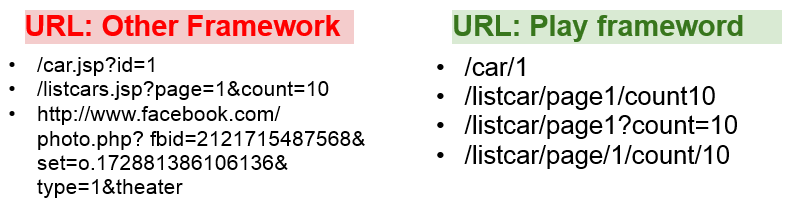
1. RESTFul

* Với thiết kế theo mô hình MVC, Play Framework có một cấu trúc thư mục rõ ràng và thân thiện với người sử dụng.



Ảnh 5: Mô hình MVC trong Play Framework

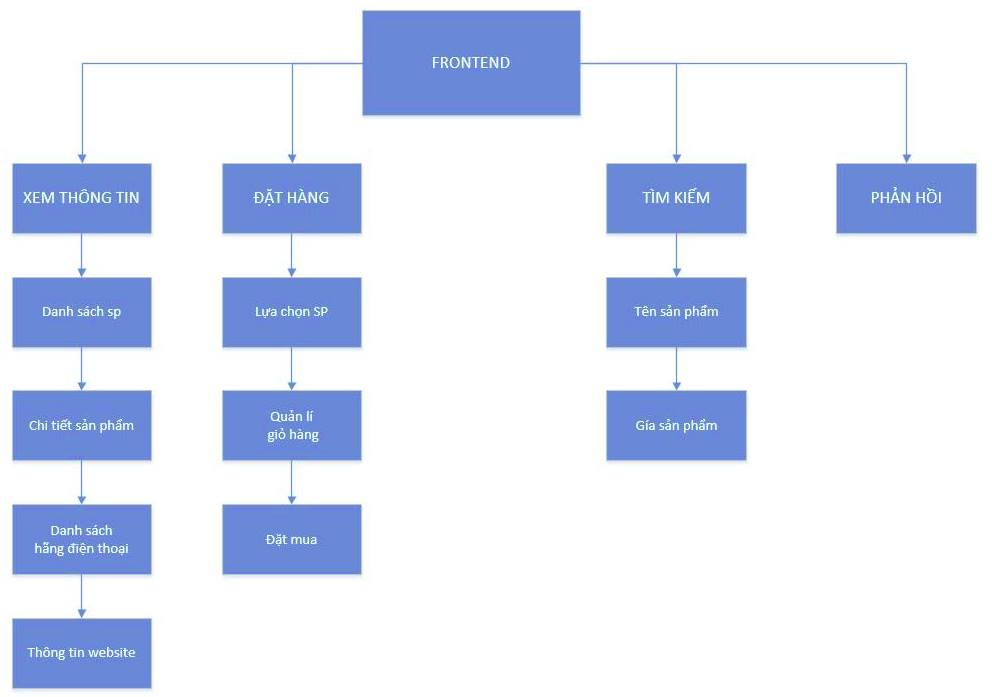
* Kiến trúc url của Play dễ nhìn và dễ quản lý hơn so với các Framework khác của Java.



Ảnh 6: URL trong router của Play Framework

# **CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

1. **Các chức năng của website**
2. Chức năng phía frontend

******

Ảnh 7: Chức năng phía front-end.

1. Xem mặt hàng

Khách hàng có thể truy cập vào website để xem thông tin về sản phẩm, danh sách sản phẩm với các thông tin cơ bản như hình ảnh, giá bán, tên sản phẩm. Danh sách này được chia thành nhiều mục tương ứng với danh mục các sản phẩm của công ty.

* Đầu vào: Thông tin sản phẩm mà khách hàng yêu cầu.
* Xử lý: Khi khách hàng chọn 1 sản phẩm tùy ý, thông tin mà khách hàng yêu cầu sẽ được gửi lên hệ thống, hệ thông sẽ lấy dữ liệu từ cơ sở dữ liệu truy xuất ra cho người dùng.
* Đầu ra: Thông tin chi tiết về điện thoại.

1. Bình luận

Người xem có thể chia sẻ cảm nghĩ, nhận xét của họ về sản phẩm, thông tin bình luận của khách hàng sẽ được lưu vào database bao gồm tên khách hàng, địa chỉ, emal cùng với chủ để và nội dung của bình luận.

* Đầu vào: thông tin về người bình luận và nội dung phản hồi.
* Xử lý: kiểm tra nội dung mà nguwofi dùng nhập vào, nếu hợp lệ thì lưu vài database, nếu không thì thông báo lỗi cho người dùng biết.
* Đầu ra: đưa ra thông báo từ hệ thống, nếu thư gửi thành công thì đưa ra thông báo “bạn đã gửi thành công”, nếu sai thì thông báo: “đã xảy ra lỗi trong hệ thống”.

1. Đặt hàng

Cho phép khách hàng duyệt sản phẩm trên trang web, khi khách hàng gửi thông tin của mặt hàng thì thông tin đó sẽ có trên giỏ hàng.

* Đầu vào: thông tin sản phẩm mà khách hàng muốn mua.
* Xử lý: khi khách hàng chọn 1 mặt hàng, ở phiên làm việc đó, nếu khách hàng chọn lần đầu tiên thì ở giỏ hàng sẽ là 1, còn nếu trong giỏ hàng đã có sản phẩm đó rồi thì sẽ có một thông báo “sản phẩm đã có trong giỏ hàng của bạn”.
* Đầu ra: thông tin về mặt hàng đó được lưu trong giỏ hàng.

1. Giỏ hàng

Cho phép khách hàng xem, điều chỉnh, thêm, xóa các mặt hàng mà khách hàng đã đặt mua, có chức năng tính toán số tiền của đơn hàng đó.

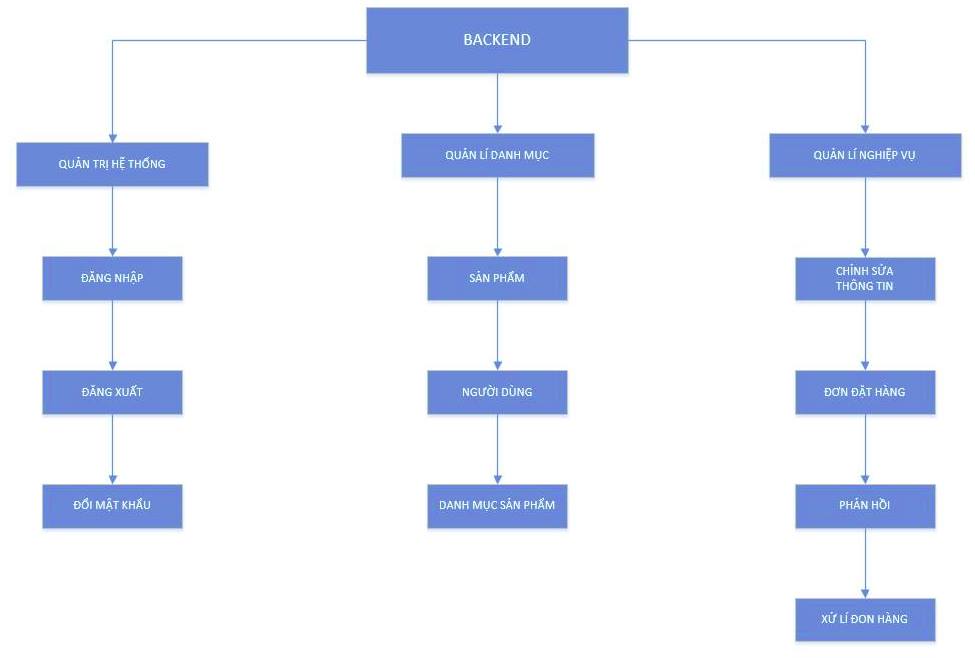
* Đầu vào: thông tin các mặt hàng mà khách hàng đặt mua, số lượng mà khách hàng muốn mua.
* Xử lý: khách hàng có thể thay đổi số lượng mặt hàng muốn mua, khi khách hàng chọn lại số lượng thì cập nhật số lượng mặt hàng đó, tổng giá trị đơn hàng cũng được tính toán lại. Trong trường hợp người dùng thay đổi số lượng sản phẩm không hợp lệ thì sẽ không thể thanh toán hóa đơn được. Khi khách hàng chọn xóa sản phẩm thì sản phẩm đố phải được xóa khỏi giỏ hàng.
* Đầu ra: Thông tin về tình trạng giỏ hàng bao gồm số lượng, giá tiền.

1. Tìm kiếm sản phẩm

Khách hàng có thể biết chi tiết về thông tin về sản phẩm mình muốn mua có thể tìm kiếm trực tiếp ở ô tìm kiếm.

* Đầu vào: tên sản phẩm mà khách hàng muốn tìm kiếm
* Xử lý: khi khách hàng nhập tên sản phẩm muốn tìm kiếm, hệ thống sẽ so sánh tên đó với dữ liệu có trong database. Nếu nhận được kết quả sẽ trả về kết quả tìm kiếm cho người dùng.
* Đầu ra: thông báo kết quả từ hệ thống.

1. Chức năng phía backend



Ảnh 8: Chức năng phía back-end.

* 1. Đăng nhập.

Người quản trị đăng nhập để truy xuất vào hệ thống.

* Đầu vào: tài khoản đăng nhập.
* Xử lý: Xác thực tài khoản, kiểm tra tính hợp lệ của đầu vào.
* Đầu ra: đăng nhập vào hệ thống nếu tài khoản chính xác, nếu tài khoản không đúng, sẽ đưa ra thông báo “tài khoản của bạn không chính xác”.
  1. Đăng xuất.

Người quản trị đăng xuất ra khỏi hệ thống.

* Đầu vào:
* Xử lý: Kết thúc phiên làm việc của người quản trị.
* Đầu ra: Đưa người quản trị trở về trang đăng nhập.
  1. Thay đổi tài khoản

Người quản trị có thể thay đổi tên hoặc mật khẩu của mình.

* Đầu vào: tài khoản hiện tại của người quản trị và thông tin thay đổi.
* Xử lý: xác thực tài khoản, kiểm tra tính hợp lệ của đầu vào và thay đổi tài khoản.
* Đầu ra: đưa ra thông báo thành công nếu phiên làm việc thành công và ngược lại.
  1. Quản lý danh mục.

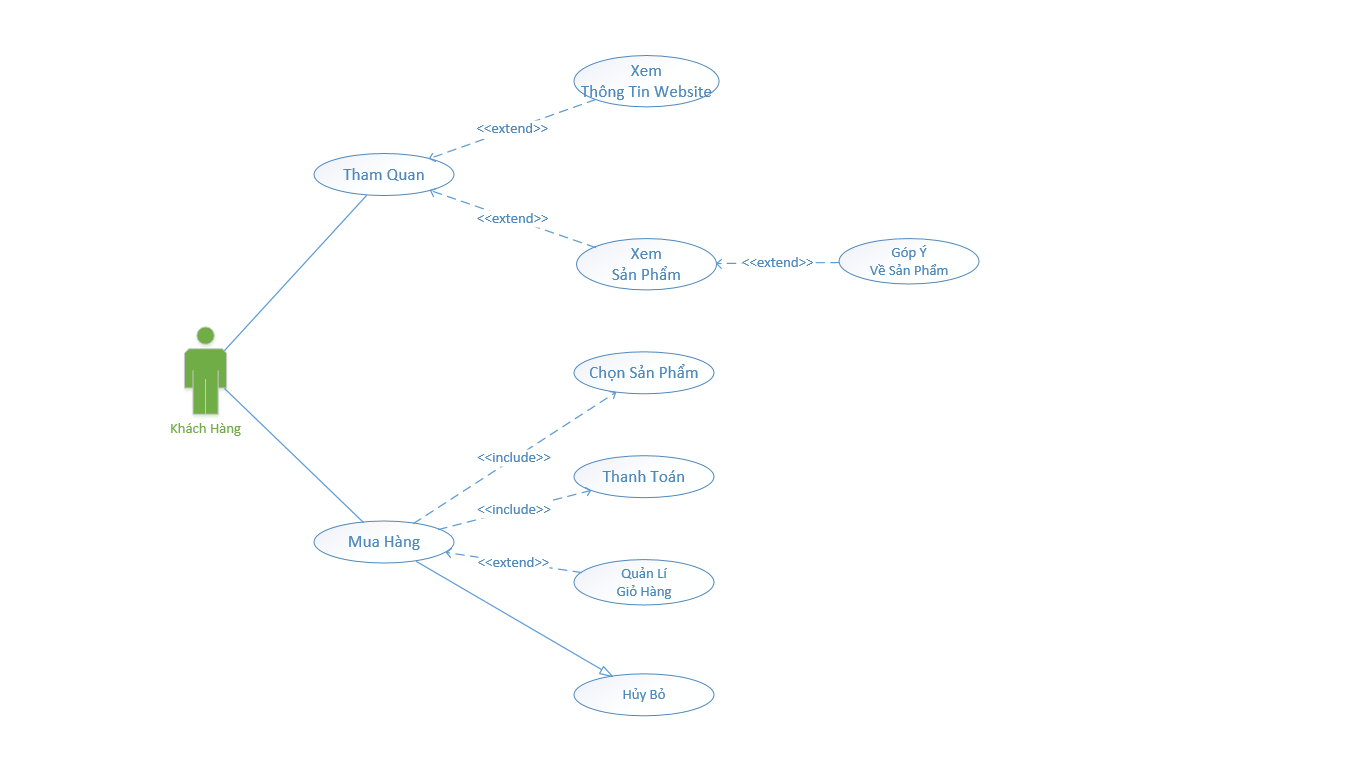
Người quản trị có thể thêm, sửa, xóa các sản phẩm có trong website.

* Đầu vào: sản phẩm bị thay đổi.
* Xử lý: kiểm tra tính hợp lệ của đầu vào và thay đổi sản phẩm.
* Đầu ra: thông báo thành công nếu phiên làm việc diễn ra thành công và ngược lại.
  1. Quản lý nghiệp vụ.

Người quản trị có thể xem các thông tin hóa đơn của khách hàng và các góp ý từ khách hàng.

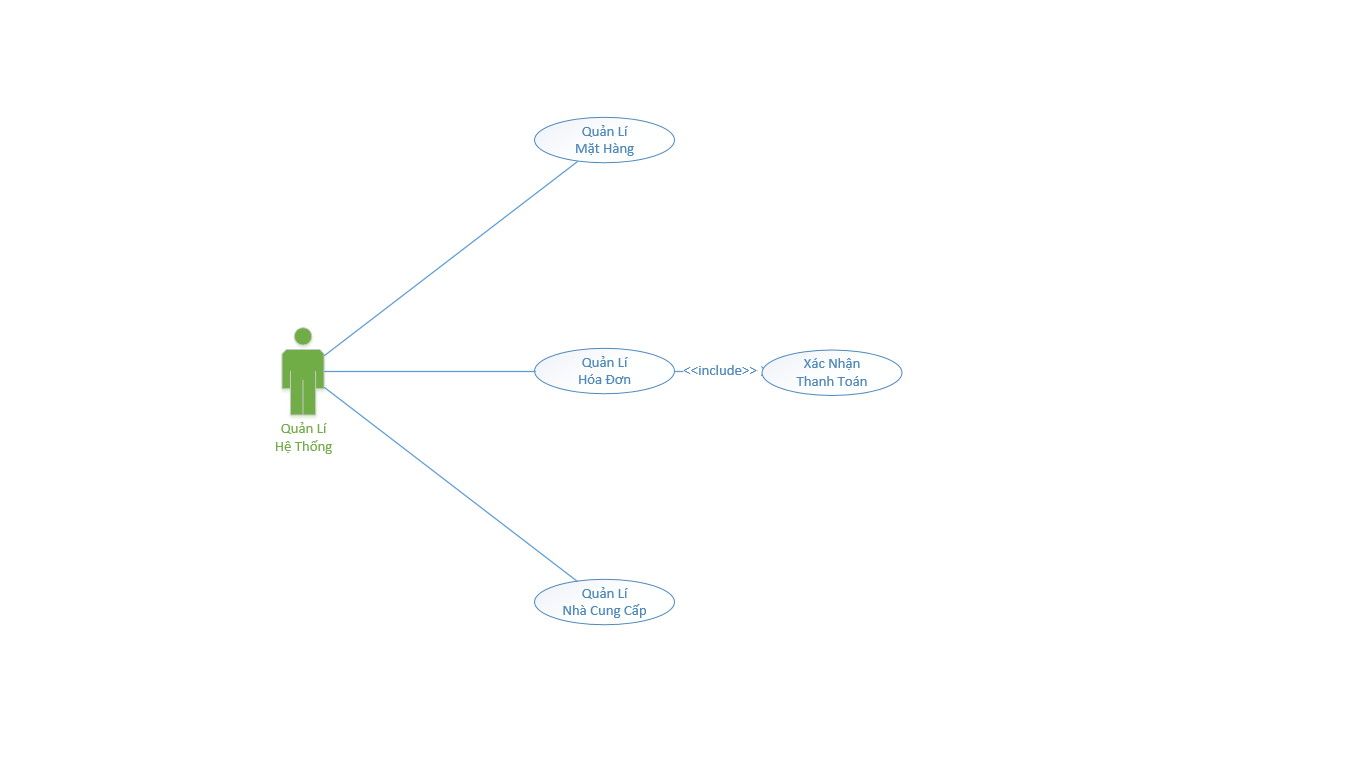
* Đầu vào: Thông tin các hóa đơn và các góp ý từ khách hàng về trang web.
* Xử lý: Đưa ra thông tin các hóa đơn và góp ý của khách hàng trong database lên.
* Đầu ra: Hiển thị các hóa đơn và góp ý.

1. **Biểu đồ USECASE**
2. Usecase: Khách hàng



Ảnh 9: Usecase khách hàng.

1. Use case: Hệ thống



Ảnh 10: Usecase hệ thống.

1. **Biểu đồ hoạt động**

1.Tạo đơn hàng



Ảnh 11: Biểu đồ hoạt động tạo đơn hàng.

1. Thanh Toán



Ảnh 12: Biểu đồ hoạt động thanh toán.

1. **Biểu đồ lớp**

Vẽ biểu đồ lớp cho ca sử dụng: Tạo đơn hàng,Thanh toán.

Phát hiện các lớp thực thể:

+ Khách hàng

+ Đơn hàng

+ Giỏ hàng

+ Sách

+ Thanh Toán

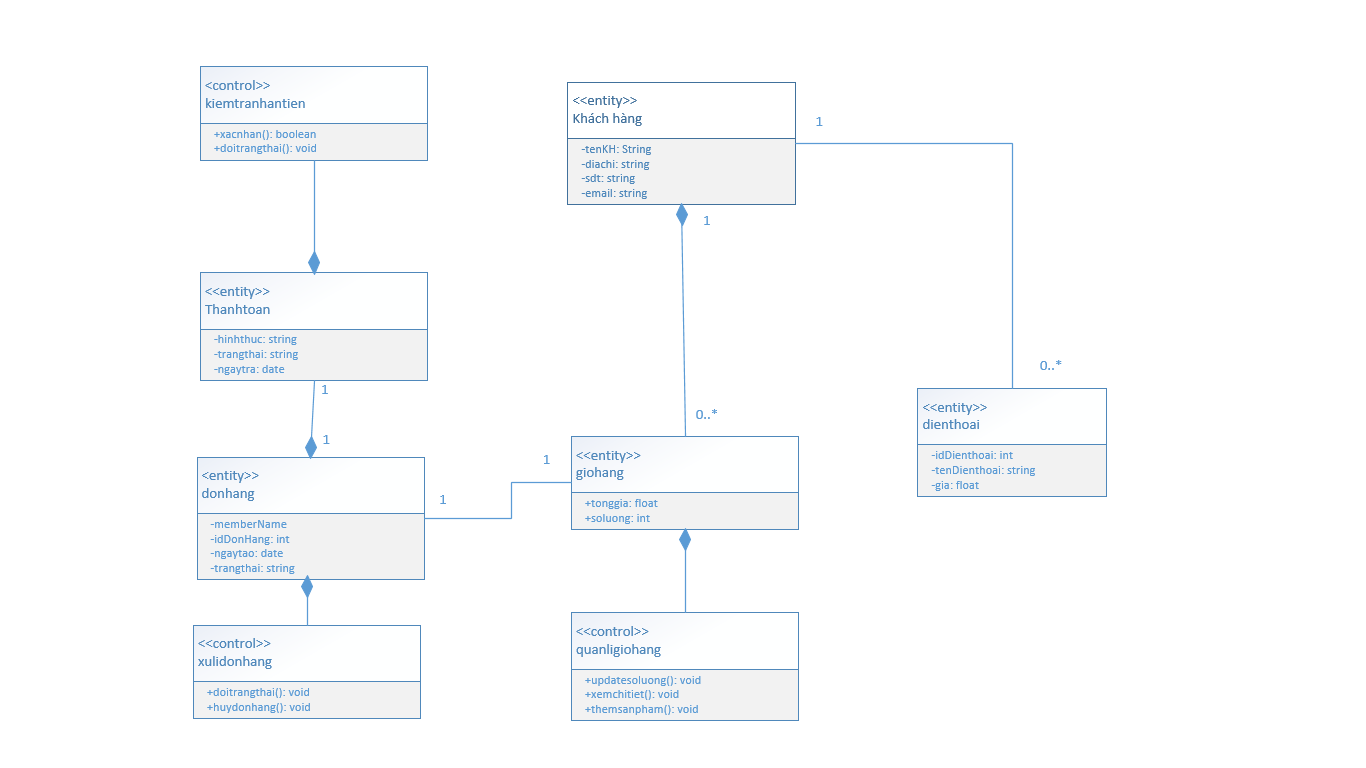
Phát hiện các lớp điều khiển:

+ Quản lý khách hàng

+ Kiểm tra nhận tiền

+ Quản lý giỏ hàng

+ Xử lý đơn hàng

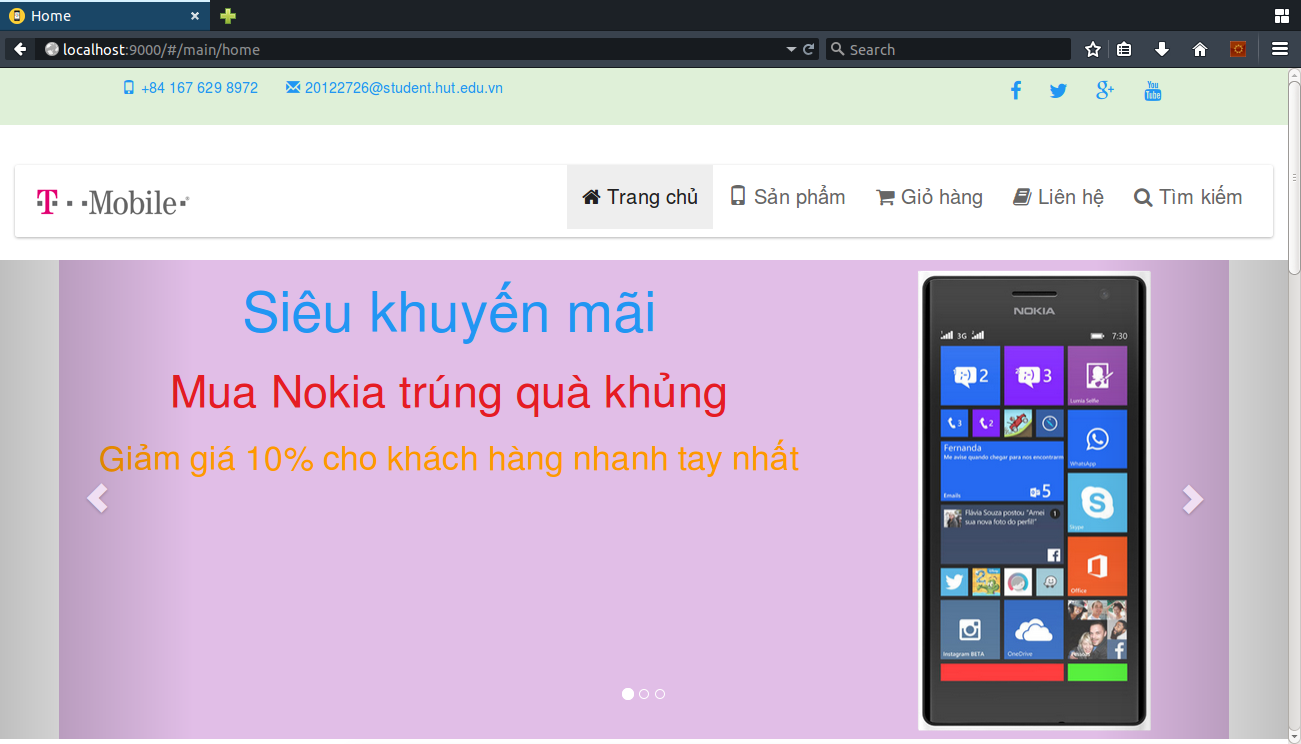


Ảnh 13: Biểu đồ lớp.

# **CHƯƠNG 3: THIẾT KẾ GIAO DIỆN VÀ CÀI ĐẶT CHƯƠNG TRÌNH**

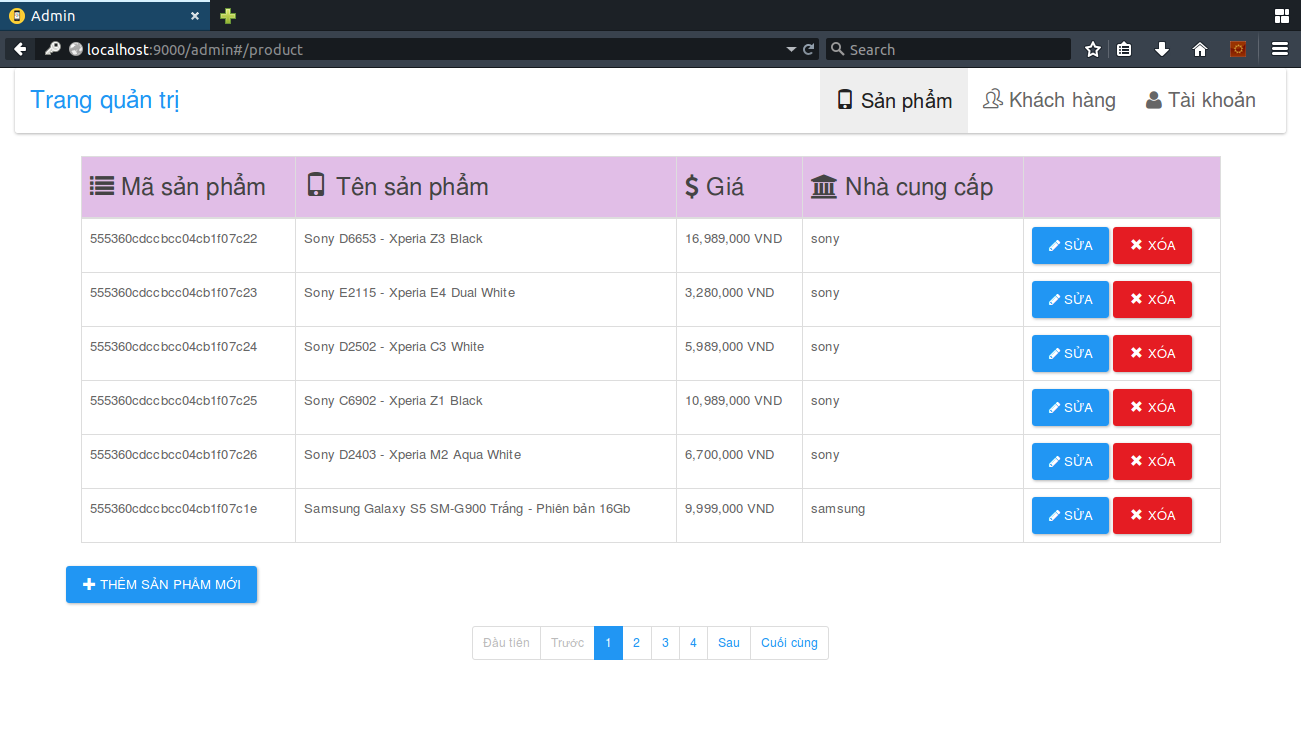
1. **Giao diện cài đặt**
2. Nền tảng sử dụng.
   * Hệ điều hành: Linux Ubuntu.
   * Ngôn ngữ: Java, JavaScript, NoSQL
   * Server: PlayFramework.
   * IDE: Eclipse, Sublime text.
   * Kiến trúc: client-server
3. Hướng dẫn cài đặt.
   * Cài đặt database: cài đặt mongodb như trang chủ của MongoDB <http://mongodb.org/>
   * Di chuyển tới thư mục project và gõ lệnh ***./activator run***sau đó mở trình duyệt và gõ ***localhost:9000.***
4. **Sản phẩm**

* Trang khách hàng



Ảnh 14: Trang khách hàng

* Trang quản trị



Ảnh 15: Trang quản trị

# **TÀI LIỆU THAM KHẢO**

* Play Framework
  + [**https://www.playframework.com/documentation/2.3.x/Home**](https://www.playframework.com/documentation/2.3.x/Home)
  + Tài liệu của Ts. Trần Thị Minh Châu ĐHQGHN [**http://uet.vnu.edu.vn/~chauttm/e-books/play/Play.htm**](http://uet.vnu.edu.vn/~chauttm/e-books/play/Play.htm)
* AngularJS
  + <https://docs.angularjs.org/tutorial>
  + <https://github.com/jmcunningham/AngularJS-Learning>
  + Slide của Vũ Nam Khánh Qsoft Vietnam <http://www.slideshare.net/qsoftvietnam/angularjs-yeoman-overviewqsoftvietnamcom>
  + Mastering Web Application Development with AngularJS của Pawel Kozlowski và Peter Bacon Darwin.
  + Ng-book của Ari Lerner.