

TẬP ĐOÀN BƯU CHÍNH VIỄN THÔNG VIỆT NAM  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

-----



# BÁO CÁO THỰC TẬP TỐT NGHIỆP ĐẠI HỌC

*Đề tài: “Ứng dụng web đọc truyện tranh trực tuyến”*

Người hướng dẫn	:	TS. HỒ MẠNH TÀI
Sinh viên thực hiện	:	TRẦN VƯƠNG QUANG
Mã số sinh viên	:	N14DCCN107
Lớp	:	D14CQCP01
Khoá	:	2014
Hệ	:	Đại học chính quy

TP.HCM, tháng 08/2018



TẬP ĐOÀN BƯU CHÍNH VIỄN THÔNG VIỆT NAM  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

-----



# BÁO CÁO THỰC TẬP TỐT NGHIỆP ĐẠI HỌC

*Đề tài: “Ứng dụng web đọc truyện tranh trực tuyến”*

**Người hướng dẫn :** TS. HỒ MẠNH TÀI  
**Sinh viên thực hiện :** TRẦN VƯƠNG QUANG  
**Mã số sinh viên :** N14DCCN107  
**Lớp :** D14CQCP01  
**Khoá :** 2014  
**Hệ :** Đại học chính quy

TP.HCM, tháng 08/2018

## LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các thầy cô của trường Học viện Công nghệ Bưu chính Viễn thông, đặc biệt là các thầy cô khoa Công nghệ Thông tin của trường đã tạo điều kiện cho em thực tập. Và em cũng xin chân thành cảm ơn thầy thầy Hồ Mạnh Tài, giảng viên hướng dẫn của em đã nhiệt tình hướng dẫn hướng dẫn để em hoàn thành tốt khóa thực tập.

Trong quá trình thực tập, cũng như là trong quá trình làm bài báo cáo thực tập, khó tránh khỏi sai sót, rất mong các thầy, cô bỏ qua. Đồng thời do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp thầy, cô để em học thêm được nhiều kinh nghiệm và chuẩn bị tốt cho công việc sau này

*Em xin chân thành cảm ơn!*

Đánh giá của cơ sở thực tập

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập- Tự do- Hạnh phúc**

TP. Hồ Chí Minh, ngày      tháng      năm 20.....

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN  
THỰC TẬP TỐT NGHIỆP ĐẠI HỌC**

- 1. Tên đề tài: Ứng dụng web đọc truyện tranh trực tuyến**
- 2. Sinh viên: TRẦN VƯƠNG QUANG      Lớp: D14CQCP01**
- 3. Giáo viên hướng dẫn: TS. HỒ MẠNH TÀI**
- 4. Nơi công tác:**

**NỘI DUNG NHẬN XÉT**

1. Đánh giá chung:

.....  
.....

2. Đánh giá chi tiết:

.....  
.....  
.....  
.....

3. Nhận xét về tinh thần, thái độ làm việc:

.....  
.....

4. Kết luận:

.....  
.....

5. Điểm hướng dẫn ():

**GIẢNG VIÊN HƯỚNG DẪN**

(Ký, ghi rõ họ tên)

## Mục lục

MỞ ĐẦU .....	1
1 CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI .....	2
1.1 Tổng quan .....	2
1.2 Tìm hiểu các công nghệ hỗ trợ thiết kế hệ thống website: .....	2
1.3 Công nghệ ứng dụng:.....	2
1.4 Công cụ sử dụng: .....	2
1.5 Mục tiêu đề tài: .....	2
2 CHƯƠNG 2: GIỚI THIỆU CÁC CÔNG NGHỆ VÀ CÔNG CỤ SỬ DỤNG .....	3
2.1 Công nghệ sử dụng: .....	3
2.1.1 Frontend: ANGULAR.....	3
2.1.2 Backend: SPRING .....	5
2.2 Công cụ sử dụng: .....	14
2.2.1 Eclipse Photon.....	14
2.2.2 Visual studio code .....	14
2.2.3 MySQL workbench.....	15
3 CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG: .....	16
3.1 Tổng quan ứng dụng: .....	16
3.1.1 Đối tượng sử dụng:.....	16
3.1.2 Các chức năng của ứng dụng: .....	16
3.2 Xây dựng các lược đồ: .....	17
3.2.1 Usecase Diagram.....	17
3.2.2 Class Diagram .....	19
4 CHƯƠNG 4: Xây dựng ứng dụng:.....	21
4.1 Quá trình thực hiện: .....	21
4.2 Giao diện các màn hình chính: .....	22
KẾT LUẬN .....	31
□ Kết quả đạt được: .....	31
□ Hạn chế:.....	31
□ Hướng phát triển: .....	31
TÀI LIỆU THAM KHẢO:.....	32

## DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH

Hình 1: Angular logo .....	3
Hình 2: Angular architecture .....	4
Hình 3: Spring framework.....	6
Hình 4: Spring layer .....	9
Hình 5: Spring Batch .....	11
Hình 6. Usecase diagram (Guest, Member, System) .....	17
Hình 7: Class Diagram .....	19
Hình 8: Giao diện trang chủ .....	22
Hình 9: Giao diện trang đăng nhập.....	23
Hình 10: Giao diện trang đăng kí .....	24
Hình 11: Giao diện tìm kiếm .....	25
Hình 12: Giao diện truyện tranh chi tiết.....	26
Hình 13: Giao diện màn hình đọc truyện .....	27
Hình 14: Giao diện màn hình truyện đang theo dõi .....	28



## MỞ ĐẦU

Truyện tranh, là những câu chuyện xảy ra trong cuộc sống hay những chuyện được tưởng tượng ra được thể hiện qua những bức tranh có hoặc không kèm lời thoại hay các từ ngữ, câu văn kể chuyện.

Hai dòng truyện tranh ảnh hưởng lớn tới mọi dòng truyện tranh hiện nay là truyện tranh Nhật Bản (Manga) và truyện tranh Âu - Mỹ (Comic). Hiện nay xuất hiện một dòng truyện tranh mới chịu ảnh hưởng của Manga Nhật Bản và đang dần nổi tiếng đó là Truyện tranh Hàn Quốc (Manhwa). Ngoài ra Manhwa (mạn họa) của Trung Quốc cũng là một dòng truyện tranh đang chiếm thị phần ngày càng lớn. Và truyện tranh đã được xem là nghệ thuật thứ chín của nhân loại. Nhiều liên hoan cùng các buổi lễ trao giải dành cho truyện tranh đã được tổ chức ở nhiều nơi trên thế giới.

Sylvain Lemay, giảng viên truyện tranh của Canada cho rằng trước đây mọi người cho truyện tranh chỉ thích hợp cho trẻ em, nhưng nay quan niệm này đã thay đổi, đối tượng của truyện tranh đã mở rộng và hướng đến người lớn nhiều hơn. Ở Việt Nam, dòng truyện tranh này cũng từng được khai thác, qua loạt Danh tác Việt Nam của công ty Phan Thị, chuyển thể các tác phẩm văn học nổi tiếng ở Việt Nam sang thể loại truyện tranh.

Nhưng với thời đại công nghệ thông tin phát triển vũ bão như hiện nay, đặc biệt là công nghệ web. Người đọc cần có thêm nhiều hơn nữa những lựa chọn để tiếp cận với thứ nghệ thuật đương đại này thay cho việc mua các ấn phẩm giấy in mực truyền thống. Chính vì vậy trang web chia sẻ truyện tranh đã ra đời, số hóa những tác phẩm nghệ thuật và giúp mọi người kết nối gần nhau hơn.

# 1 CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

## 1.1 Tổng quan

Phát triển một ứng dụng web phục vụ mục đích đọc truyện tranh trực tuyến cơ bản của người dùng. Gồm các chức năng:

- Đọc truyện
- Tìm kiếm truyện theo tên
- Tìm kiếm truyện theo thể loại
- Theo dõi truyện
- Để lại nhận xét
- Bầu chọn truyện

## 1.2 Tìm hiểu các công nghệ hỗ trợ thiết kế hệ thống website:

- **Lý thuyết:**
  - Tìm hiểu Spring framework.
  - Tìm hiểu Angular.
  - Tìm hiểu RESTful API.
- **Thực hành:**
  - Phân tích và thiết kế hệ thống
  - Xây dựng website đọc truyện có các tính năng cơ bản như sau:
    - Đăng kí/Đăng nhập
    - Xem truyện
    - Tìm truyện theo tên
    - Lọc truyện theo thể loại
    - Đăng kí, theo dõi truyện

## 1.3 Công nghệ ứng dụng:

- Spring framework
- Angular 6
- MySQL
- RESTful API

## 1.4 Công cụ sử dụng:

- Eclipse Photon
- Visual studio code
- MySQL workbench

## 1.5 Mục tiêu đề tài:

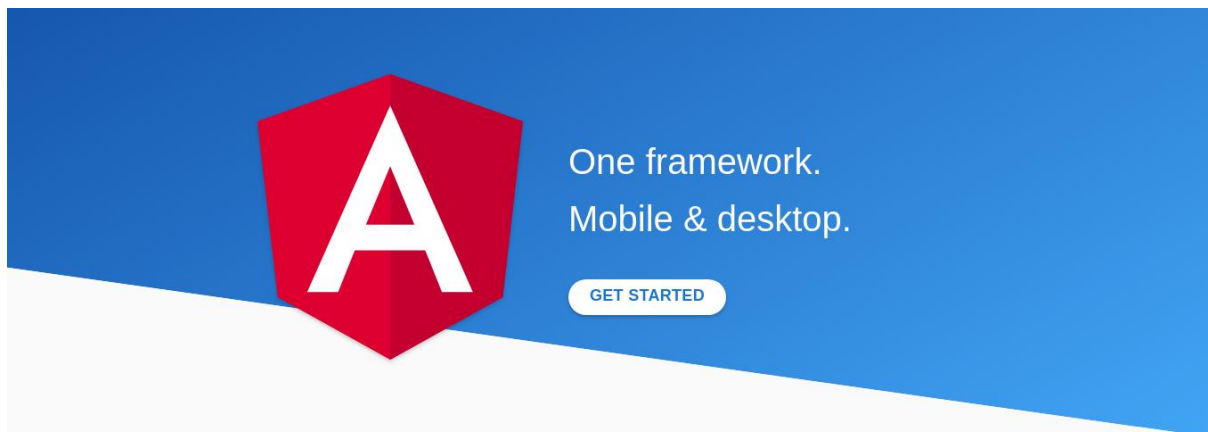
- Hệ thống phải dễ sử dụng, có tính khả thi
- Cung cấp chức năng đã nêu ở trên cũng như bổ sung thêm những tính năng mới trong quá trình phát triển.

## 2 CHƯƠNG 2: GIỚI THIỆU CÁC CÔNG NGHỆ VÀ CÔNG CỤ SỬ DỤNG

### 2.1 Công nghệ sử dụng:

#### 2.1.1 Frontend: ANGULAR

##### 2.1.1.1 Angular là gì?



Hình 1: Angular logo

Đối với một lập trình viên web, làm việc với javascript là một kỹ năng không thể thiếu, để làm trang web sinh động và mang tính tương tác cao hơn. Thư viện JQuery đã làm rất tốt điều đó. Nhưng công nghệ thì luôn thay đổi và phát triển, Single Page Applications (SPA) đã trở nên phổ biến và phát triển cực kì mạnh mẽ và Angular (ngoài ra còn có React , Vue.Js ...) sinh ra để làm điều đó. Vậy Angular là gì?

Angular là một nền tảng giúp xây dựng ứng dụng web dễ dàng. Ngoài ra có thể sử dụng Angular để viết native app cho di động hoặc desktop. Angular được google phát triển nên chúng ta có thể yên tâm về vấn đề phát triển lâu dài và cộng đồng hỗ trợ. Angular sử dụng Typescript để phát triển ứng dụng. TypeScript vừa có tính chặt chẽ của một ngôn ngữ có kiểu (kiểm tra kiểu dữ liệu ngay khi compile), vừa giữ được sự linh hoạt và gọn nhẹ của javascript, giúp cho angular phù hợp với tất cả các project ở mọi quy mô.

##### 2.1.1.2 Tại sao chọn Angular?

Angular là một trong những framework giành được rất nhiều sự ưa chuộng gần đây. Vì sao Angular được phổ biến đến vậy? Sự kết nối của cộng đồng, mã nguồn mở, sự quan tâm của Google team, v.v. Angular là một framework tốt và giải quyết rất nhiều vấn đề. Một trong những lợi thế của Angular là khả năng tạo ra nhiều ứng dụng trên một trang duy nhất.

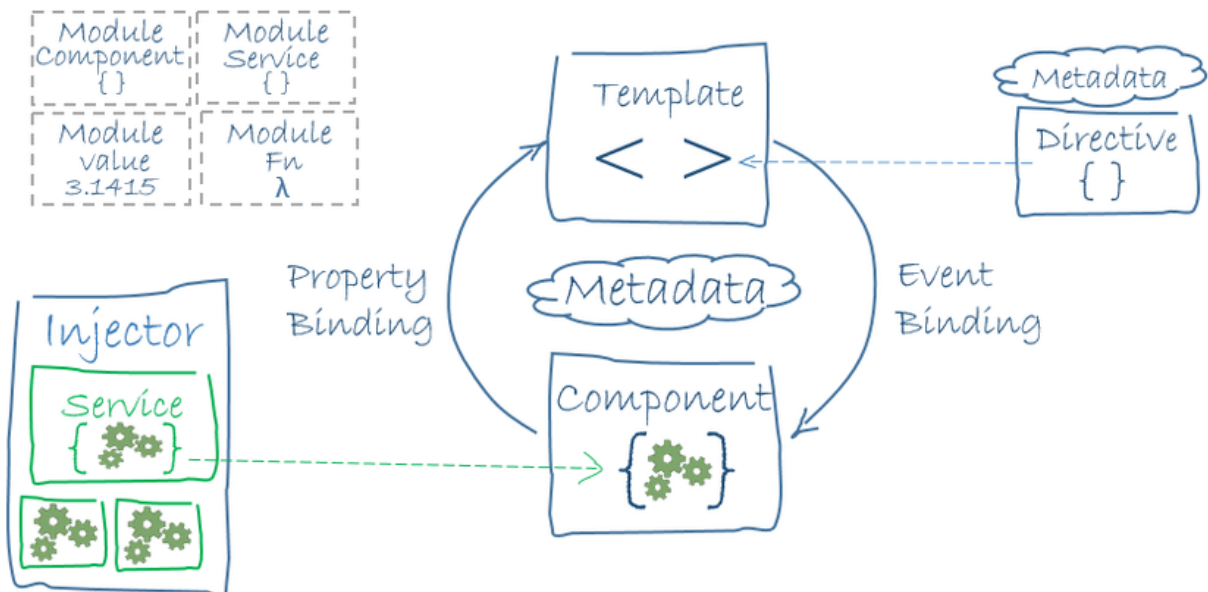
Việc chuyển đổi AngularJS 1 (AngularJS) sang AngularJS 2 (Angular) đã gây ra một chút phiền toái cho hầu hết các developer bởi vì không giống với các framework khác, quá trình thay đổi sẽ được áp dụng từ từ cho ứng dụng, nhưng đối với AngularJS, chúng ta giống như là viết lại từ đầu. Việc hỗ trợ cho AngularJS vẫn sẽ được duy trì trong một thời gian dài như thể AngularJS và Angular là hai framework thật sự khác nhau.

Nếu đã từng xây dựng một vài ứng dụng Web, chúng ta đều nghe đến cái tên Angular, một Framework Javascript giúp xây dựng ứng dụng Web đầy đủ tính năng từ phía Client. Angular lần đầu được phát hành bởi gã khổng lồ Google vào năm 2010 với phiên bản AngularJS, sau đó đã có chỗ đứng khá vững chắc trong một thời gian dài, phiên bản Angular 2 phát hành năm 2016 mang đến một bước chuyển mình vượt bậc, một công cụ thực sự mạnh mẽ cho việc phát triển ứng dụng Web trên cả nền tảng Mobile và Desktop. Hiện nay chúng ta đang có phiên bản Angular 6.

Hiện giữa vô vàn các thư viện và frameworks Javascript, Angular có gì nổi bật và tại sao chúng ta nên chọn Angular cho ứng dụng tiếp theo của mình ?

- **Angular giúp nâng cao năng suất của các lập trình viên:**
- **Cấu trúc phát triển rõ ràng:**
- **Extensive binding:**
- **Hỗ trợ đầy đủ tính năng điều hướng (routing):**
- **Angular giúp giảm tối đa kích thước và tăng tối đa hiệu suất của ứng dụng.**
- **Document và cộng đồng (community)**

### 2.1.1.3 Angular Architecture



Hình 2: Angular architecture

Dựa vào sơ đồ kiến trúc trên ta có thể liệt kê ra 8 thành phần chính tạo ra một ứng dụng Angular:

- **Modules:**
- **Angular modules vs. JavaScript modules**

- **Angular libraries**
- **Components**
- **Templates**
- **Metadata**

## **2.1.2 Backend: SPRING**

### **2.1.2.1 Spring là gì?**

Spring là một framework giúp các nhà phát triển xây dựng những hệ thống và ứng dụng chạy trên JVM một cách đơn giản, tiện gọn, nhanh chóng và mềm dẻo.

- **EJB và Spring**

Có thể nói rằng Spring Framework ra đời là nhờ sự thành công của EJB. EJB (Enterprise JavaBeans) là một kiến trúc phát triển dành cho các ứng dụng doanh nghiệp đòi hỏi tính mạnh mẽ và khả năng mở rộng cao. EJB là một thành phần quan trọng của nền tảng Java EE. EJB được tạo ra bởi hai gã khổng lồ công nghệ là IBM và Sun vào những năm cuối thập niên 90. Dưới sự phát triển của hai tập đoàn này, EJB đã trở nên nặng nề hơn. Tiêu chuẩn EJB cũng nhanh chóng được thông qua. Sau tất cả, EJB được ca ngợi là giải pháp tốt nhất để xây dựng các ứng dụng phân tán dành cho doanh nghiệp.

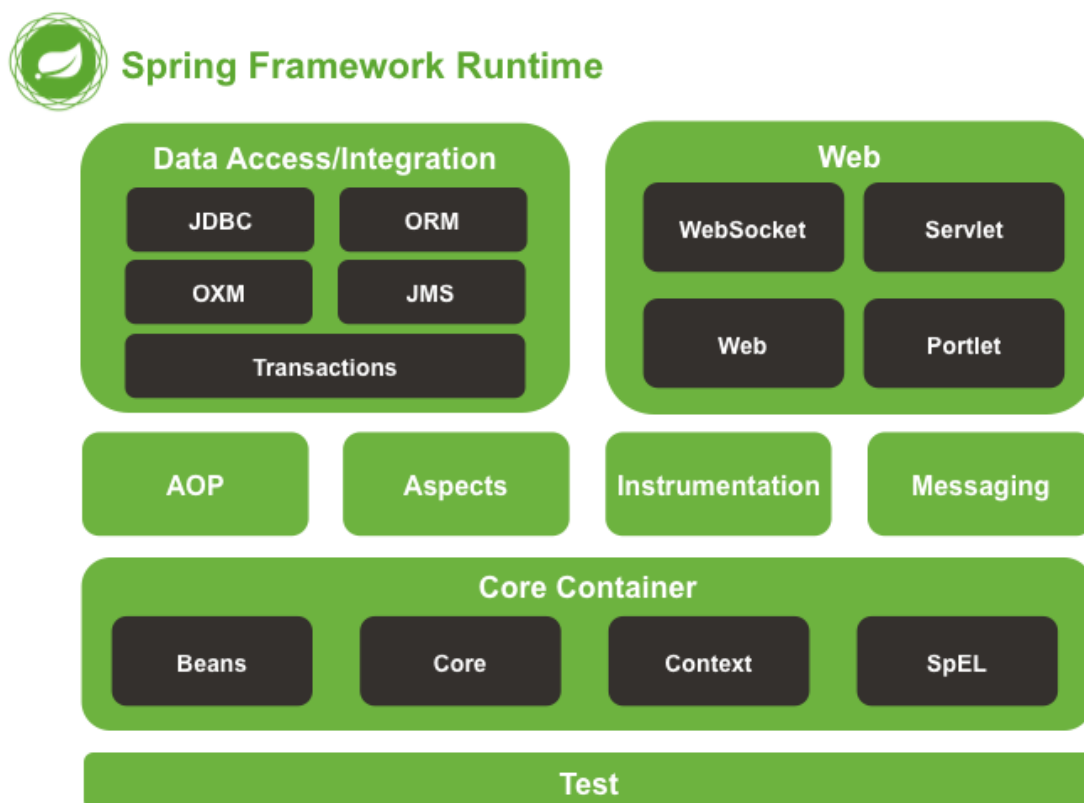
Nhưng rồi những chỉ trích về chi tiết kỹ thuật của EJB đã tăng nhanh còn hơn sự nổi tiếng của nó. EJB là một framework lớn với độ phức tạp cao. Các nhà phát triển khi làm việc với EJB đã nhận ra rằng nó không hoạt động như những gì nó đã đề ra. EJB chỉ tuyệt vời trên bàn giấy, nhưng ra thực tế đó lại là một chuyện khác.

EJB vẫn được tiếp tục phát triển, dễ sử dụng hơn nhiều so với buổi ban đầu. Nhưng những chỉ trích vẫn còn đó, nỗi thất vọng trong cộng đồng Java ngày càng tăng lên, đòi hỏi cần có một giải pháp thay thế.

- **Spring Core**

Spring Core chính là thành phần trung tâm, cốt lõi của Spring Framework. Đây chính là nền tảng để xây dựng nên các thành phần khác.

Lược đồ dưới đây mô tả mối quan hệ giữa Spring Core với các thành phần khác trong Spring Framework:



Hình 3: Spring framework

- **Spring Bean**

Spring Bean là trung tâm của Spring Core và là trái tim của một ứng dụng Spring. Trái ngược với EJB, Spring Framework được thiết kế từ lỗi bằng cách sử dụng các POJO hay các Spring Bean. Spring Bean có thể được hiểu là các đối tượng Java đơn giản. Điều này khá tương đồng với nguyên lý thiết kế "Đơn trách nhiệm" (single responsibility) của Robert C Martin trong lập trình hướng đối tượng.

- **Dependency Injection (DI)**

Dependency Injection (có thể dịch tiêm các thành phần phụ thuộc) là một sức mạnh nổi bật của Spring Framework. Trong các phiên bản gốc của EJB người ta không thấy sự xuất hiện của Dependency Injection, nhưng có lẽ do nhận thấy sự thành công từ Spring khi sử dụng Dependency Injection, các phiên bản EJB 3.x đã cung cấp các annotation để thực hiện inject các dependency. Dependency Injection là một mẫu thiết kế phần mềm mà các đối tượng phụ thuộc sẽ được inject vào một lớp nào đó. Dependency Injection là một implementation cụ thể của khái niệm Inversion of Control (đảo ngược điều khiển).

Ta sẽ có một ví dụ đơn giản về Dependency Injection như sau: Bạn có một web controller có nhiệm vụ lưu thông tin gửi từ form người dùng. Theo nguyên lý Đơn trách nhiệm, bạn không muốn lớp controller tương tác với cơ sở dữ liệu. Thay vào đó, bạn sẽ sử dụng một lớp service để làm công việc này. Như vậy, controller của bạn sẽ chỉ phải xử lý dữ liệu của form (get form data, validate data, ...) rồi gọi một phương thức của lớp service được inject để lưu dữ liệu. Controller không cần phải quan tâm hay lo lắng về kết nối cơ sở dữ liệu, pooling hay bảng nào sẽ được update. Cũng như service không cần phải biết request có những thông tin gì.

- **Spring Context**

Spring Context mang mọi thứ lại với nhau. Spring Context kế thừa các tính năng của Spring Bean và bổ sung các hỗ trợ cho internationalization (ví dụ như các resource bundle), event propagation, resource loading ... Ngoài ra, Spring Context cũng hỗ trợ các tính năng của Java EE như EJB, JMX và truy cập từ xa cơ bản. Interface ApplicationContext là tiêu điểm của Spring Context.

- **Spring Expression Language (SpEL)**

Spring Expression Language là một ngôn ngữ ngắn gọn giúp cho việc cấu hình Spring Framework trở nên linh hoạt hơn.

- **Các dự án trong Spring Framework**

Trên thực tế Spring Framework là một tập hợp của nhiều dự án con. Spring Core như chúng ta đã tìm hiểu là nền tảng của các dự án trong Spring Framework. Một dự án sẽ đảm nhận một chức năng riêng trong việc xây dựng các ứng dụng doanh nghiệp. Chúng ta sẽ cùng đi qua các dự án lớn trong Spring.

- **Spring MVC**

Spring MVC được thiết kế dành cho việc xây dựng các ứng dụng nền tảng web. Đây là một dự án chúng ta không thể bỏ qua khi xây dựng các ứng dụng Java web.

- **Spring Data**

Cung cấp một cách tiếp cận đúng đắn để truy cập dữ liệu từ cơ sở dữ liệu quan hệ, phi quan hệ, map-reduce và thậm chí còn hơn thế nữa.

- **Spring Security**

Dự án này cung cấp các cơ chế xác thực (authentication) và phân quyền (authorization) cho ứng dụng của bạn.

- **Spring Boot**

Spring Boot là một framework giúp chúng ta phát triển cũng như chạy ứng dụng một cách nhanh chóng.

- **Spring Batch**

Dự án này giúp chúng ta dễ dàng tạo các lịch trình (scheduling) và tiến trình (processing) cho các công việc xử lý theo mẻ (batch job).

- **Spring Integration**

Spring Integration là một implementation của Enterprise Integration Patterns (EIP). Dự án này thiết kế một kiến trúc hướng thông điệp hỗ trợ việc tích hợp các hệ thống bên ngoài.

- **Spring XD**

Bạn đang băn khoăn về Big Data, tại sao không thử tìm hiểu Spring XD - mục tiêu của dự án này là đơn giản hóa công việc phát triển các ứng dụng Big Data.

- **Spring Social**

Dự án này sẽ kết nối ứng dụng của bạn với các API bên thứ ba của Facebook, Twitter, LinkedIn ...

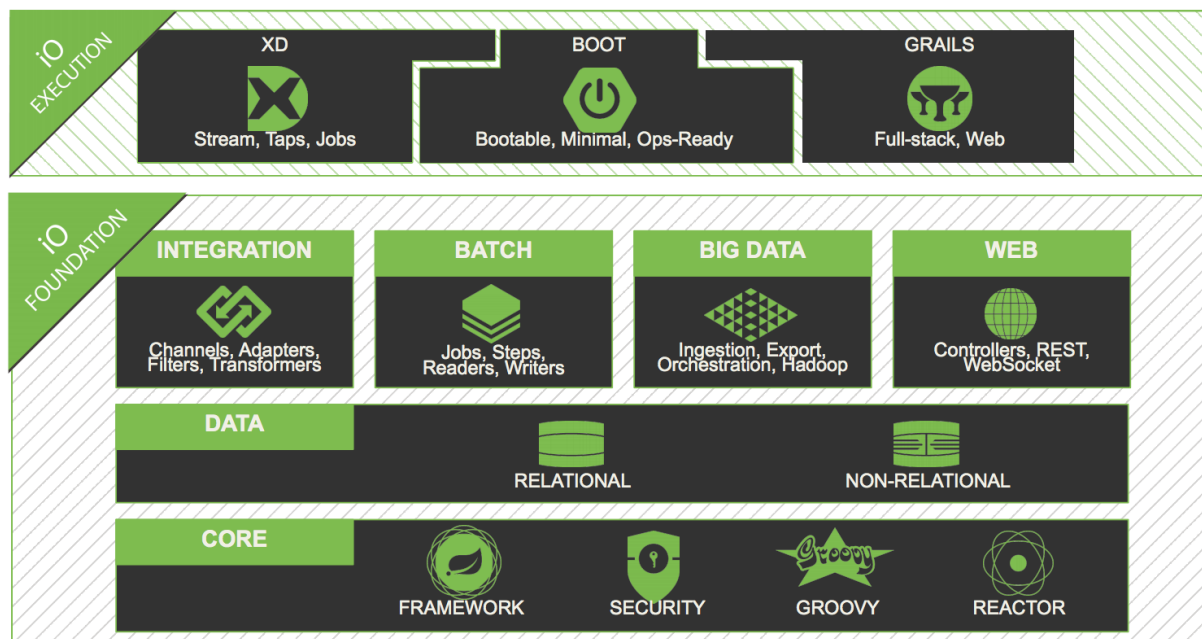
### 2.1.2.2 Tại sao chọn Spring?

Trải qua hơn 20 năm phát triển, Java đã để lại nhiều dấu ấn, tốt có, xấu có. Nếu coi Java như nền tảng quan trọng để xây dựng Enterprise Software thì Spring chiếm một phần không hề nhỏ trong nền tảng ấy.

Khi Spring ra đời, nhiệm vụ của nó là thay thế các công nghệ Java dùng cho doanh nghiệp, vốn khá cồng kềnh và nặng nề như EJB. Spring đề xuất một giải pháp nhẹ nhàng, tinh gọn hơn so với EJB bằng cách bơm cho POJO sức mạnh vốn chỉ có ở EJB và các anh em của nó. Dần dần EJB và J2EE cũng cải tiến theo hướng của Spring: EJB cũng sử dụng POJO, triển khai ý tưởng DI(Dependency Injection) và AOP(Aspect-Oriented Programming).

Dù J2EE(hay JEE) có thể đuổi kịp được Spring, Spring vẫn không ngừng phát triển và vươn xa đến những vùng đất J2EE mới chỉ chấp chững hoặc chưa từng đặt chân vào như: phát triển trên nền mobile, tích hợp social API, NoSQL database, cloud computing, big data... Tương lai của Spring thực sự rất tươi sáng.





Hình 4: Spring layer

### Lí do nên chọn Spring:

- **Xây dựng Web Application với Spring:**  
 Với nhiều Java developer, phát triển web application là mục tiêu chính của họ. 3 thử thách cần vượt qua : state management, workflow, validation. Một điều không vui nữa, HTTP lại là stateless design, vì vậy, vượt qua 3 thử thách kể trên chưa bao giờ đơn giản.  
 Spring MVC là một framework được xây dựng trên nền của Spring. Nó vẫn bảo tồn được các tinh hoa của Spring và là một công cụ đắc lực hỗ trợ trong vấn đề này. Dựa trên mô hình MVC kinh điển, Spring MVC sẽ giúp bạn xây dựng các web application linh hoạt và mạnh mẽ.  
**Spring Web Flow:** Nếu cần một web app kiểu stateful, hãy thử Spring Web Flow. Spring Web Flow giúp xây dựng một wizard - một tập hợp các cửa sổ mà người dùng phải duyệt qua tuần tự trước khi lấy được dữ liệu. Spring Web Flow giúp bạn tránh được việc lưu trữ quá nhiều dữ liệu tạm. Ví dụ các website đặt vé máy bay, vé tàu hỏa...
- **Tiếp cận với RESTful Service:**  
 Một web service developer sẽ cung cấp một "hợp đồng" với consumer của service đó trước khi consumer gọi đến service.

Trong thế giới của SOAP, điều này gần giống với việc tạo một interface, nhờ interface đó mà client biết được method nào có thể sử dụng được.

Trong thế giới của REST, chúng ta xây dựng các URL kèm theo dữ liệu phù hợp, chúng ta vẫn cần biết server đang "mong đợi" thông tin gì.

*Jersey cũng là một open source web service xây dựng trên nền tảng Java. Việc xây dựng web service của Jersey tuân theo đặc tả của JSR. Việc convert các API từ Spring sang Jersey và ngược lại tương đối dễ dàng.*

Jersey chỉ tập trung vào REST, Jersey không thể hỗ trợ một JSP webpage trong khi Spring MVC mang lại cho bạn cả 2 thứ một cách xuất sắc. Nếu client muốn dùng jQuery trong JSP webpage, Spring MVC hoàn toàn đáp ứng được. Kể cả khi chúng ta muốn sử dụng một framework JavaScript MVC bất kỳ, Spring MVC vẫn hỗ trợ. Khi phân vân giữa Spring MVC và các giải pháp khác, hãy theo Spring MVC.

Spring MVC mang đến sự thanh thoát cũng như đảm bảo cấu trúc mạch lạc cho mô hình servlet programming. Nó tách riêng model với view (Thực ra vấn đề này được Struts MVC giải quyết từ cách đây rất lâu. Spring MVC chỉ theo sau một chút)

Các framework Javascript MVC có vài trò gần giống với Spring MVC nhưng chúng được áp dụng cho tầng Javascript. Với các framework Javascript MVC, bạn có thể dựng các trang web độc lập với logic của server.

Hai tầng được dựng từ Java và Javascript sẽ liên lạc và trao đổi dữ liệu với server thông qua các đường RESTful (RESTful URL).

- **Spring Security và vấn đề bảo mật:**

Với tất cả project có dính dáng đến phát triển web, đều cần đến Spring Security. Sau đây là 2 phút giới thiệu về Spring Security:

Spring Security sẽ intercept các URL theo một mẫu (pattern) nào đó, chẳng hạn, các URL có dạng "/user" thì mọi người dùng đều truy cập được, còn dạng "admin" thì chỉ khả dụng với một vài cá nhân nào đó.

Nhờ Spring Security, chúng ta có thể dán một annotation lên các method để phục vụ công tác bảo mật.

Spring Security cung cấp cả 2 cơ chế authentication(xác thực) và authorization(phân quyền):

Xác thực - authentication: xác định bạn là ai

Phân quyền - authorization: giới hạn những hành động của bạn

Khi bắt đầu một project mới để sử dụng 2 tính năng này, ta tích hợp Spring Security và sử dụng cơ chế in-memory authentication.

Với cơ chế này, chúng ta sẽ tạo ra sẵn một đồng username, password và các phân quyền, nhét chúng vào một file xml hoặc java... Spring Security sẽ cho

phép chúng ta truy cập với một username và password tương ứng, sau khi truy cập, chúng ta sẽ được phân quyền phù hợp. Sau này, chúng ta có thể thay đổi cơ chế xác thực, phân quyền mà không làm ảnh hưởng tới business logic của hệ thống.

- **Thao tác trong CSDL với Spring:**

Để làm việc với cơ sở dữ liệu, ta thường dùng Spring ORM và hibernate.

ORM = Object Relational Mapping - Một cơ chế cho phép biểu diễn các bảng dữ liệu dưới dạng Java object và ngược lại. Hibernate và myBatis là hai ORM framework phổ biến nhất. Một sự thật thú vị, trước khi iPhone và iPad trở nên phổ biến, myBatis từng được biết đến dưới cái tên iBatis.

Nếu bạn phải xử lý một lượng lớn các đoạn code JDBC cồng kềnh, hãy chuyển qua Spring JDBC. Spring JDBC sẽ giúp thu gọn đáng kể khối code JDBC truyền thống thông qua các Template

Trường hợp dùng hibernate cho ORM nhưng phải gọi một SQL procedure từ hệ quản trị CSDL Oracle. Vì hibernate chỉ chấp nhận các dạng procedure "phù hợp" nên sẽ dùng Spring JDBC để gọi procedure này nếu hibernate không chấp nhận nó.

- **Spring Batch:**



Hình 5: Spring Batch

Spring Batch cung cấp một tập hợp các reusable function (các hàm có thể tái sử dụng). Các hàm này giữ vai trò trọng yếu khi xử lý một lượng lớn các bản ghi trong CSDL.

Có những trường hợp những câu truy vấn CSDL mất tới 1 tiếng đồng hồ để hoàn thành. Và tất nhiên chúng ta không thể bắt người dùng đợi từng đó thời gian để lấy được kết quả mà họ mong muốn.

Ví dụ: Một project về kiểm toán, mỗi phép toán trên CSDL đều tác động lên hàng triệu bản ghi, với trường hợp này, cần dùng Spring Batch. Khi đó, mỗi request phát sinh sẽ kích hoạt một Spring Batch Job và ngay lập tức một Job ID sẽ được trả về. Ta có thể dùng Job ID đó để kiểm soát trạng thái của tiến trình xử lý và thông báo cho người dùng biết tiến độ công việc. Các thông báo cho người dùng có thể xuất hiện dưới dạng email notification hoặc bạn có thể dùng web sockets để bắn trực tiếp notification tới người dùng khi họ đang online.

Spring Batch cũng cung cấp cho một giải pháp để tự động sinh các thông báo khi trạng thái của hệ thống thay đổi. ...

Spring Batch sẽ đảm bảo tính toàn vẹn và nhất quán dữ liệu.

- **Spring Integration và các hệ thống ngoại vi:**

Ví dụ: Trong một project, đối tác sẽ gửi các report qua web service với định dạng XML. Trung bình, chúng ta nhận 100.000 file XML trong 6 tháng. Các file XML này không quá lớn nhưng vấn đề ở đây là không thể xác định được thời điểm mà các file sẽ được gửi đến. Vì vậy, mục tiêu là phải có một bộ phận tự động "hứng" được các file này ngay khi chúng được gửi đến, sau đó tiến hành tiền xử lý rồi chuyển tiếp vào hệ thống.

Đó là một kịch bản phổ biến khi xây dựng một service phụ thuộc vào hệ thống bên ngoài. Với trường hợp này, chúng ta chọn Spring Integration. Ta cài đặt một file poller - lấy nơi đó để lưu trữ các file mà bên khách hàng gửi đến. Khi có bất cứ file nào "bay" đến, một event sẽ được tự động kích hoạt, thực hiện các nghiệp vụ xử lý rồi chuyển vào hệ thống.

Tất nhiên phần code xử lý các file này độc lập hoàn toàn so với phần còn lại của hệ thống. Nhờ vậy, hệ thống không phụ thuộc hoàn toàn vào hệ thống của đối tác. Bên cạnh đó, việc test các mô-đun nhận file này cũng dễ dàng hơn rất nhiều.

- **Spring Test:**

Spring không hề đứng ngoài xu thế TDD. Nó cung cấp một module phục vụ cho test các application - Spring Test

Trong module này, với Unit Test, sẽ có một tập các mock object implementation để viết unit test tương thích với JNDI, servlet, portlets. Còn với Integration Test, module này hỗ trợ việc load các bean theo nhiều context khác nhau.

- **Áp dụng Spring cho các Java project đơn lẻ:**

Ví dụ: một sản phẩm có giao diện dạng CLI - Command Line Interface - giao diện command line. Sản phẩm này có các business rules engine dựa trên nền Java và rất nặng về XML parsing. Khi hệ thống có vấn đề, rất khó để test xem rắc rối đang xảy ra ở đâu.

Giải pháp đề xuất lúc đó là áp dụng Spring Framework vào quy trình test. Dùng các mô-đun Spring Context, Spring Test và JUnit, sau đó tạo một test package trong project đó. Tester sẽ có một file XML và các unit test để chuẩn bị dữ liệu trước mỗi lượt test.

- **Chuyển đổi application thành dạng executable với Spring boot:**

Spring Boot là một project mới dựa trên ý tưởng sử dụng chính Spring để đơn giản hóa Spring.

Spring Boot chủ yếu sử dụng automatic configuration để loại bỏ Spring configuration truyền thống. Dù ta sử dụng Gradle hay Maven, Spring Boot đều cung cấp một số starter project được tối ưu sẵn để giảm kích thước build files.

- **Spring Social và quá trình tích hợp các phương tiện truyền thông:**

Dùng Spring Social để cho phép người dùng sign in vào application. Application cho phép đăng nhập từ Facebook, Google, LinkedIn, Twitter..... Spring Social sử dụng cơ chế OAuth để việc login từ các trang mạng xã hội trở nên suôn sẻ.

cũng có thể áp dụng Spring Mobile cho application. Spring Mobile cho phép bạn tinh chỉnh application sao cho phù hợp với các thiết bị di động. Dù vậy, thực tế cho thấy Spring Mobile không phải là lựa chọn đúng đắn nhất cho các thiết bị di động. Thay vào đó, hãy cân nhắc Bootstrap và AngularJS.

- **Bắt đầu một project với Spring:**

Spring sẽ giúp bạn bắt đầu một project bất kì phù hợp với nhu cầu của chính bạn.

Trong vòng vài phút, Spring Boot sẽ giúp bạn tạo một web app hoặc standalone app, rất đơn giản nhưng không kém phần mạnh mẽ.

Spring Cloud là nền tảng để xây dựng micro service một cách nhanh chóng. Micro service xây dựng bởi Spring Cloud đã được tự động config phục vụ cho registration và discovery.

Spring XD sẽ là trợ thủ đắc lực cho các hệ thống phân tán và big data.

## **2.2 Công cụ sử dụng:**

### **2.2.1 Eclipse Photon**

Eclipse là 1 công cụ hỗ trợ lập trình mã nguồn mở được phát triển bởi IBM.

Eclipse như một môi trường phát triển Java tích hợp (IDE), với Eclipse chúng ta có thể mở rộng hơn mã nguồn bằng cách chèn thêm các plugins cho project (PDE- Plug-in Development Environment). Mặc dù Eclipse được viết bằng ngôn ngữ lập trình Java, nhưng việc sử dụng nó không hạn chế chỉ cho ngôn ngữ Java. Ví dụ, Eclipse hỗ trợ sẵn hoặc có thể cài thêm các plugins để hỗ trợ cho các ngôn ngữ lập trình như C/C++ và COBOL. Ngoài ra, còn rất nhiều ngôn ngữ khác như PHP, Groovy, ...

Eclipse còn hỗ trợ cho lập trình viên code theo các mô hình phát triển như MVC, tạo thêm các lib hỗ trợ phát triển phần mềm.

#### **Ưu điểm**

- Tạo thuận lợi cho tích hợp liên mạch các công cụ bên trong mỗi một và xuyên qua nhiều kiểu nội dung và các nhà cung cấp công cụ khác nhau.
- Hỗ trợ việc xây dựng nhiều công cụ.
- Hỗ trợ một tập hợp không hạn chế các nhà cung cấp công cụ, bao gồm cả các nhà cung cấp phần mềm độc lập (ISV).
- Hỗ trợ các công cụ thao tác các kiểu nội dung bất kỳ (bao gồm cả HTML, Java, C, JSP, EJB, XML, và GIF).
- Hỗ trợ cả môi trường phát triển ứng dụng GUI lẫn không dựa trên GUI.
- Chạy trên nhiều hệ điều hành, bao gồm cả Windows và Linux.
- Lợi dụng tính phổ biến của ngôn ngữ Java để viết các công cụ.
- Do sử dụng SWT/JFace nên có lẽ load nhanh hơn

#### **Nhược điểm**

- Cài đặt phức tạp
- Tốn phần cứng máy Eclipse cũng khá tốn bộ nhớ máy để cài đặt nên hiện nay eclipse đã khắc phục bằng phiên bản Eclipse Luna, không cần cài đặt
- Tốn bộ nhớ máy: chạy eclipse cũng khá tốn bộ nhớ máy tính nên để sử dụng nó máy của bạn cũng cần có cấu hình tương đối cao
- Thiếu tính nhất quán vì có nhiều plugins quá
- 

### **2.2.2 Visual studio code**

Visual Studio Code là một trình biên tập mã được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nó cũng cho phép tùy chỉnh, do đó, người dùng có thể thay đổi theme, phím tắt, và cá tùy chọn khác. Nó miễn phí và là phần mềm mã nguồn mở, mặc dù gói tải xuống chính thì là có giấy phép.

Visual Studio Code được dựa trên Electron, một nền tảng được sử dụng để triển khai các ứng dụng Node.js máy tính cá nhân chạy trên động cơ bố trí Blink. Mặc dù nó sử

dụng nền tảng Electron nhưng phần mềm này không phải là một bản khác của Atom, nó thực ra được dựa trên trình biên tập của Visual Studio Online (tên mã là "Monaco").

Trong cuộc khảo sát vào năm 2018 trên Stack Overflow, Visual Studio Code được xếp hạng là trình biên tập mã phổ biến nhất, với 34.9% của 75398 người trả lời tuyên bố sử dụng nó.

Visual Studio Code là một trình biên tập mã. Nó hỗ trợ nhiều ngôn ngữ và chức năng tùy vào ngôn ngữ sử dụng theo như trong bảng sau. Nhiều chức năng của Visual Studio Code không hiển thị ra trong các menu tùy chọn hay giao diện người dùng. Thay vào đó, chúng được gọi thông qua khung nhập lệnh hoặc qua một tập tin .json (ví dụ như tập tin tùy chỉnh của người dùng). Khung nhập lệnh là một giao diện theo dòng lệnh. Tuy nhiên, nó biến mất khi người dùng nhấp bất cứ nơi nào khác, hoặc nhấn tổ hợp phím để tương tác với một cái gì đó ở bên ngoài đó. Tương tự như vậy với những dòng lệnh tồn nhiều thời gian để xử lý. Khi thực hiện những điều trên thì quá trình xử lý dòng lệnh đó sẽ bị hủy.

Visual Studio Code có thể được mở rộng qua plugin. Điều này giúp bổ sung thêm chức năng cho trình biên tập] và hỗ trợ thêm ngôn ngữ. Một tính năng đáng chú ý là khả năng tạo phần mở rộng để phân tích mã, như là các linter và công cụ phân tích, sử dụng Language Server Protocol.

### **2.2.3 MySQL workbench**

MySQL Workbench là trình trực quan để giao tiếp với MySQL, nó thực hiện hiển thị đồ họa cho người dùng như các bảng, các database, mô hình... Nếu không có nó thì ta sẽ phải giao tiếp với MySQL bằng giao diện dòng lệnh khá phức tạp và mất thời gian.

### 3 CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG:

#### 3.1 Tổng quan ứng dụng:

Trang web được viết theo kiến trúc REST, nên sẽ chia ứng dụng thành 2 phần để lập trình: Server và Client.

- ❖ Phần Server: Dùng Spring để lập trình các chức năng, cung cấp API cho phía Client.
- ❖ Phần Client: Dùng Angular để lập trình các chức năng, gọi API từ Server cung cấp để lấy dữ liệu và thay đổi một phần trang thay vì load lại toàn bộ

##### 3.1.1 Đối tượng sử dụng:

- Khách: người dùng truy cập trang web chưa login vào hệ thống
- Thành viên: người dùng có tài khoản và đã truy cập vào hệ thống qua chức năng login

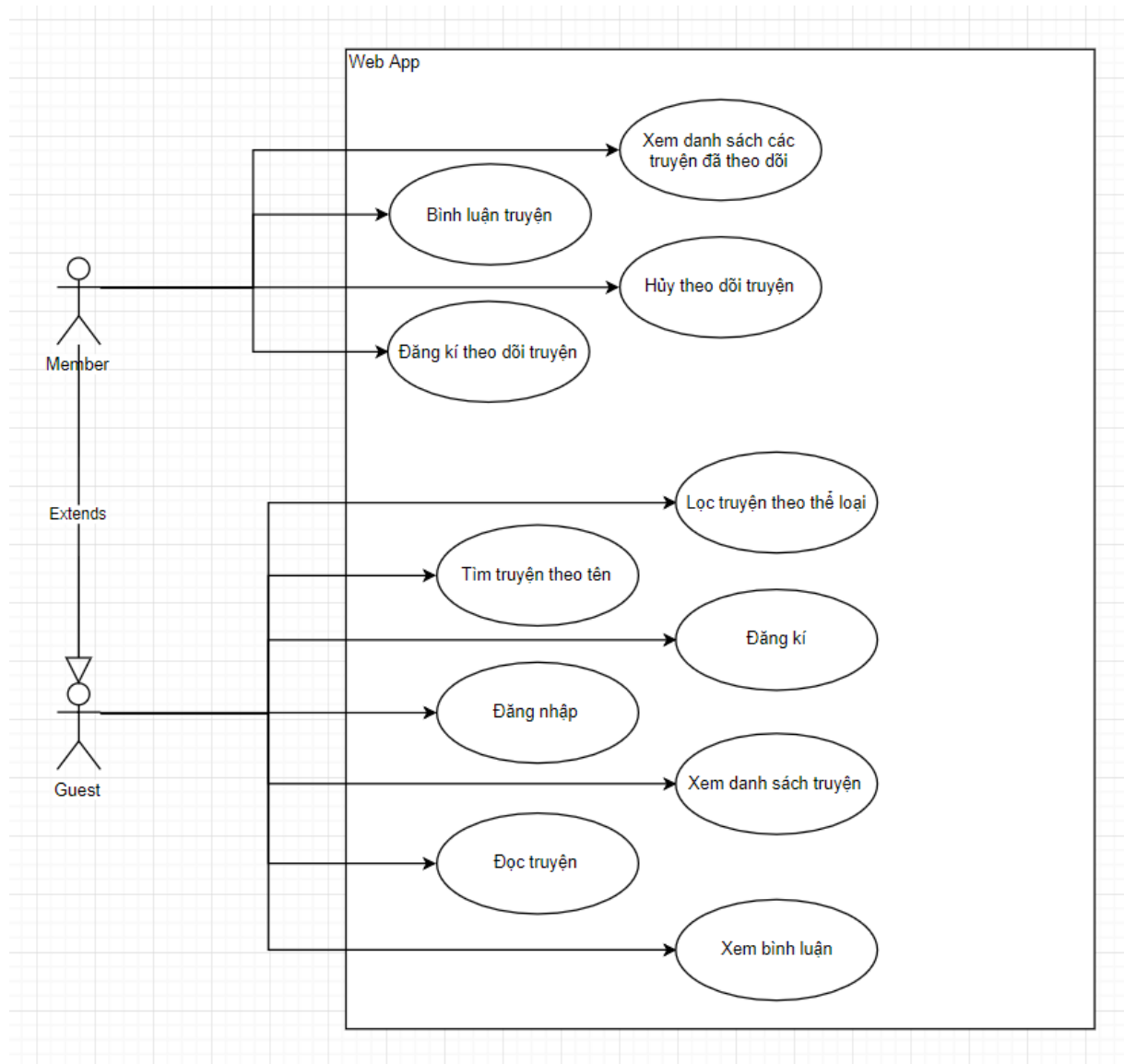
##### 3.1.2 Các chức năng của ứng dụng:

- Đối với khách:
  - Đăng kí tài khoản mới
  - Đăng nhập hệ thống
  - Xem danh sách các truyện
  - Đọc truyện
  - Tìm kiếm truyện theo tên
  - Lọc danh sách truyện theo thể loại
  - Xem danh sách bình luận của truyện
- Đối với thành viên đã đăng nhập:
  - Các chức năng tương tự như khách
  - Bình luận truyện
  - Lưu các truyện vào tủ truyện
  - Hủy các truyện đã theo dõi
  - Xem danh sách các truyện đã theo dõi



## 3.2 Xây dựng các lược đồ:

### 3.2.1 Usecase Diagram



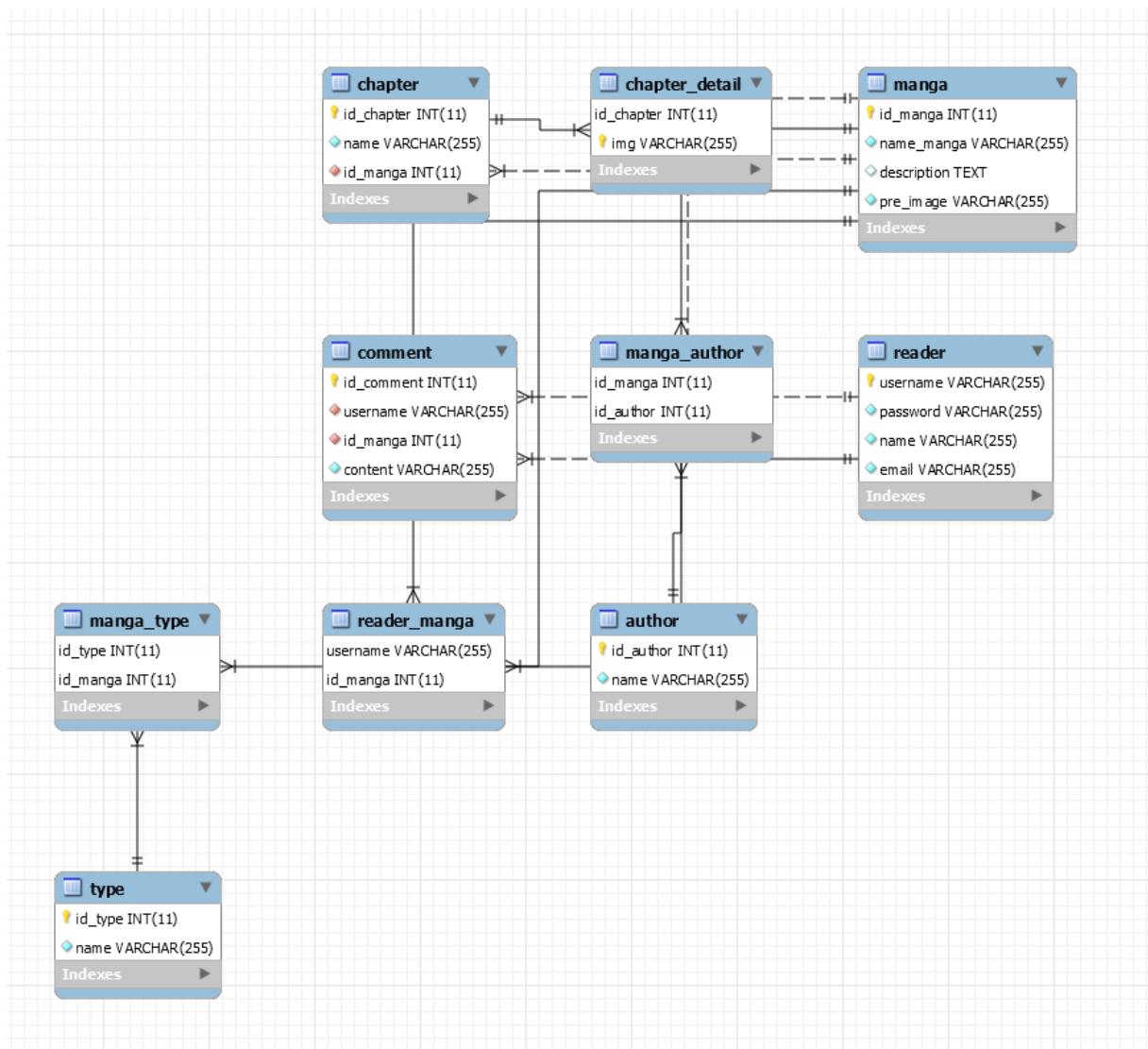
Hình 6. Usecase diagram (Guest, Member, System)

- Actor:
  - o Khách: người truy cập trang web nhưng chưa đăng nhập
  - o Thành viên: người truy cập đã đăng nhập
- Usecase::

Tên use case	Actor	Mô tả
Đăng kí	Khách	Khách nhập thông tin để hệ thống lưu trữ, tạo tài khoản mới

Đăng nhập	Khách	Khách nhập thông tin đã đăng kí để đăng nhập vào hệ thống
Xem danh sách truyện	Khách/Thành viên	Xem danh sách các truyện có trong hệ thống
Đọc truyện	Khách/Thành viên	Đọc nội dung truyện
Xem bình luận	Khách/Thành viên	Xem bình luận của truyện
Tìm truyện theo tên	Khách/Thành viên	Tìm kiếm truyện theo tên truyện bằng cách nhập trên truyện vào ô tìm kiếm
Lọc truyện theo thể loại	Khách/Thành viên	Lọc danh sách truyện theo thể loại hiện có trong hệ thống
Bình luận truyện	Thành viên	Đề lại bình luận về một truyện khi đã đăng nhập
Đăng kí theo dõi truyện	Thành viên	Thêm một truyện vào tủ truyện của thành viên đó
Hủy theo dõi truyện	Thành viên	Xóa một truyện khỏi tủ truyện của thành viên đó
Xem danh sách các truyện đã theo dõi	Thành viên	Xem danh sách truyện hiện có trong tủ truyện của thành viên đó

### 3.2.2 Class Diagram



Hình 7: Class Diagram

Số TT	Table	Trường	Kiểu dữ liệu	Mô tả
1	Type	Id_type	INT(11)	Khóa chính của bảng thể loại
		Name	VARCHAR(255)	Tên của thể loại
2	Manga_type ( <u>id_type, id_manga</u> )	Id_type	INT(11)	Xác định thể loại
		Id_manga	INT(11)	Xác định truyện

3	Manga	Id_manga	INT(11)	Khóa chính, xác định truyện
		Name_manga	VARCHAR(255)	Tên truyện
		Description	TEXT	Mô tả truyện
		Pre_image	VARCHAR(255)	Đường dẫn tới ảnh mô tả của truyện
4	Chapter	Id_chapter	INT(11)	Khóa chính xác định chương
		Name	VARCHAR(255)	Tên của chương truyện
		Id_manga	INT(11)	Xác định truyện mà chương này thuộc về
5	Chapter_detail (id_chapter,img)	Id_chapter	INT(11)	Xác định chương truyện mà thông tin này thuộc về
		Img	VARCHAR(255)	Đường dẫn tới nội dung hình ảnh
6	Author	Id_author	INT(11)	Xác định tác giả
		Name	VARCHAR(255)	Tên tác giả
7	Manga_author	id_manga	INT(11)	Xác định truyện
		Id_author	INT(11)	Xác định tác giả
8	Reader	Username	VARCHAR(255)	Tên truy cập, xác định thành viên
		Password	VARCHAR(255)	Mật khẩu thành viên
		Name	VARCHAR(255)	Tên thành viên
		Email	VARCHAR(255)	Địa chỉ email thành viên
9	Reader_manga	Username	VARCHAR(255)	Xác định thành viên
		Id_manga	INT(11)	Xác định truyện mà thành viên trên theo dõi
10	Comment	Id_comment	INT(11)	Xác định bình luận
		Username	VARCHAR(255)	Xác định người bình luận
		Id_manga	INT(11)	Xác định truyện mà bình luận thuộc về
		Content	VARCHAR(255)	Nội dung bình luận

## **4 CHƯƠNG 4: Xây dựng ứng dụng:**

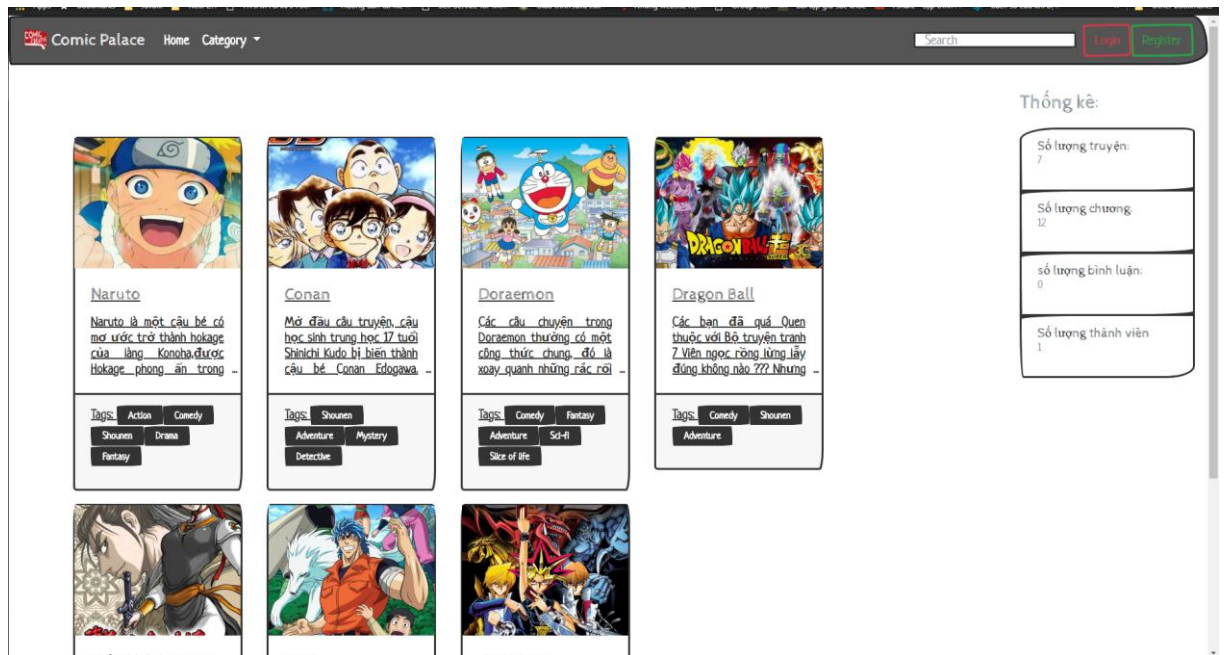
Dựa vào các thiết kế bên trên, bắt đầu quá trình lập trình để hiện thực các yêu cầu hệ thống.

### **4.1 Quá trình thực hiện:**

- Bước 1: Phân tích, thiết kế, mô tả các chức năng cần thiết của website.
- Bước 2: Tạo Project và lập trình ở phần Backend.
- Bước 3: Viết API cho trang web.
- Bước 4: Thêm dữ liệu vào cơ sở dữ liệu và kiểm tra tính đúng đắn của dữ liệu thông qua việc gọi API.
- Bước 5: Thiết kế giao diện cho trang web.
- Bước 6: Sử dụng Angular, Bootstrap, HTML, CSS,...để tiến hành lập trình frontend theo thiết kế.
- Bước 7: Lập trình xử lý dữ liệu từ Backend gửi về thông qua API, hiển thị thông tin cho người dùng.
- Bước 8: Chạy thử Website và sửa lỗi.

## 4.2 Giao diện các màn hình chính:

- Trang chủ:



Hình 8: Giao diện trang chủ

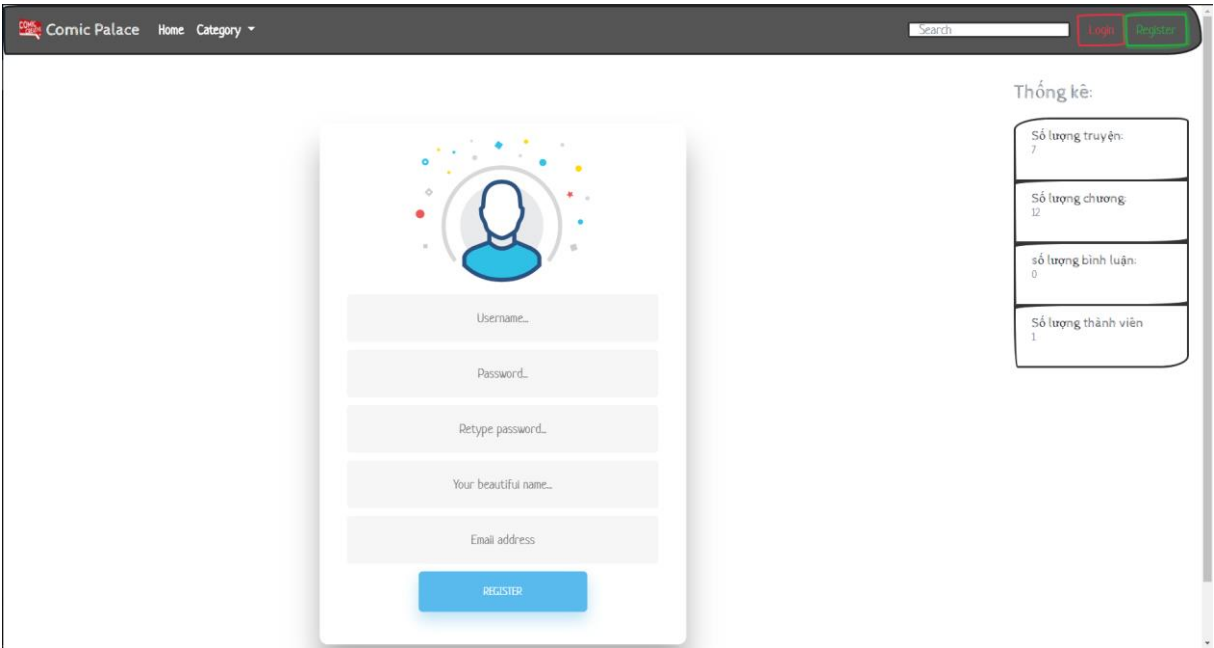
Khu vực	Thành phần	Loại	Mô tả
Thanh điều hướng trên cùng	Logo + Brand	Link	Hiển thị logo và tên ứng dụng
	Home	Link	Điều hướng về trang chủ
	Category	Dropdown	Hiển thị toàn bộ thể loại truyện và lọc truyện theo thể loại
	Search	Text Input	Nhập tên truyện cần tìm kiếm
	Login	Button	Đăng nhập
	Register	Button	Đăng kí
Nội dung chính của trang	ListTruyen	Card List	Danh sách các truyện gồm thông tin: Hình đại diện, tên truyện, mô tả, thể loại
Khung thống kê	Sidebar	List	Hiển thị thống kê một số thông tin của ứng dụng

- Giao diện trang đăng nhập:

Hình 9: Giao diện trang đăng nhập

Thành phần	Loại	Mô tả
Khung nhập username	Text Input	Nhập username
Khung nhập password	Text Input	Nhập password
Nút Xác nhận	Button	Đăng nhập

- Giao diện trang đăng kí:



Hình 10: Giao diện trang đăng kí

Thành phần	Loại	Mô tả
Khung nhập username	Text Input	Nhập username
Khung nhập password	Text Input	Nhập password
Khung xác nhận password	Text Input	Xác nhận password
Khung nhập tên	Text Input	Nhập thông tin tên
Khung nhập địa chỉ email	Text Input	Nhập thông tin email
Nút Register	Button	Đăng nhập



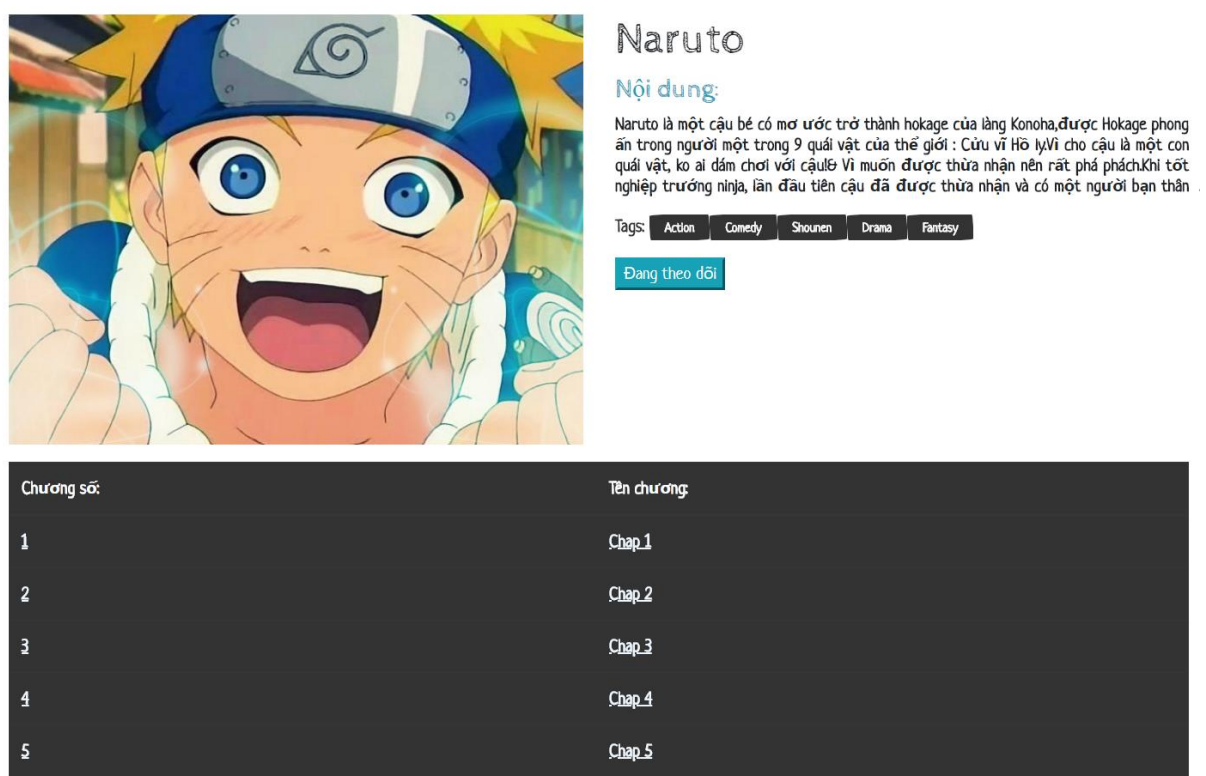
- Giao diện tìm kiếm:



Hình 11: Giao diện tìm kiếm

- Người dùng nhập tên truyện, ứng dụng hiển thị danh sách các truyện gần đúng nhất

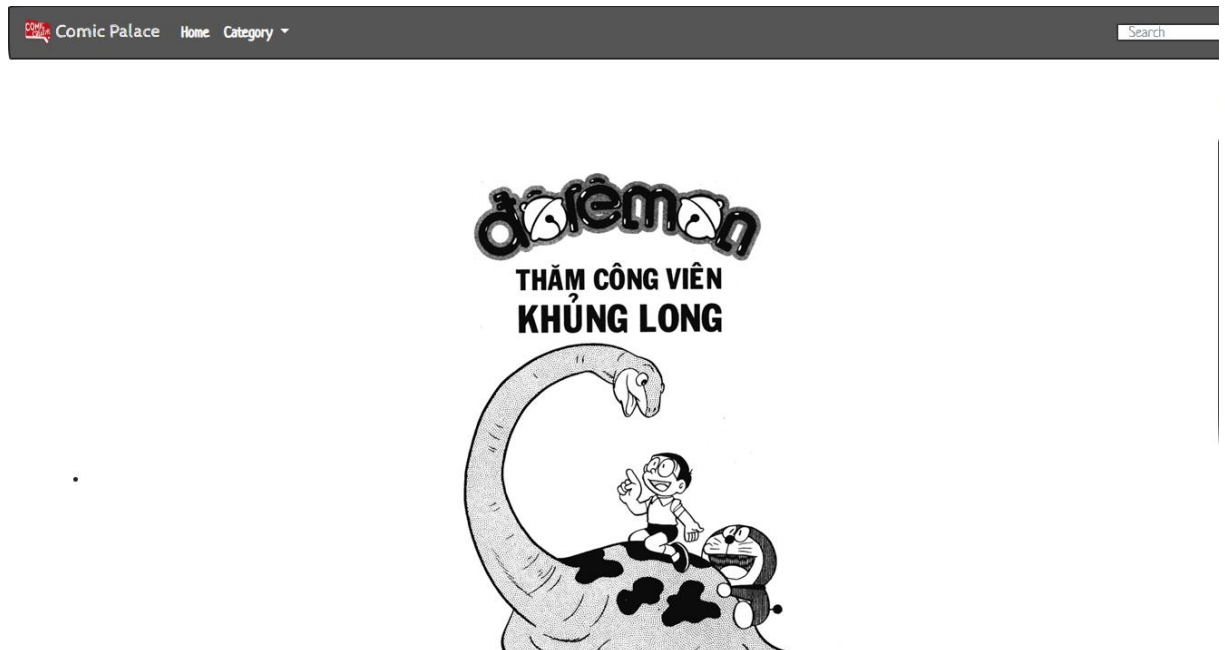
- Giao diện thông tin đầy đủ của truyện tranh:



Hình 12: Giao diện truyện tranh chi tiết

Thành phần	Loại	Mô tả
Hình đại diện	img	Hình đại diện của truyện
Naruto	Text	Tên truyện
Nội dung	Text	Nội dung của truyện
Tags	Badge	Các thể loại của truyện
Đang theo dõi	Button	Theo dõi hoặc bỏ theo dõi truyện khi đã đăng nhập
Table chương	Table	Thông tin các chương của truyện

- Màn hình đọc truyện:



Hình 13: Giao diện màn hình đọc truyện

Hiện thị nội dung hình ảnh của chương truyện khi được chọn

- Giao diện màn hình truyện đang theo dõi:

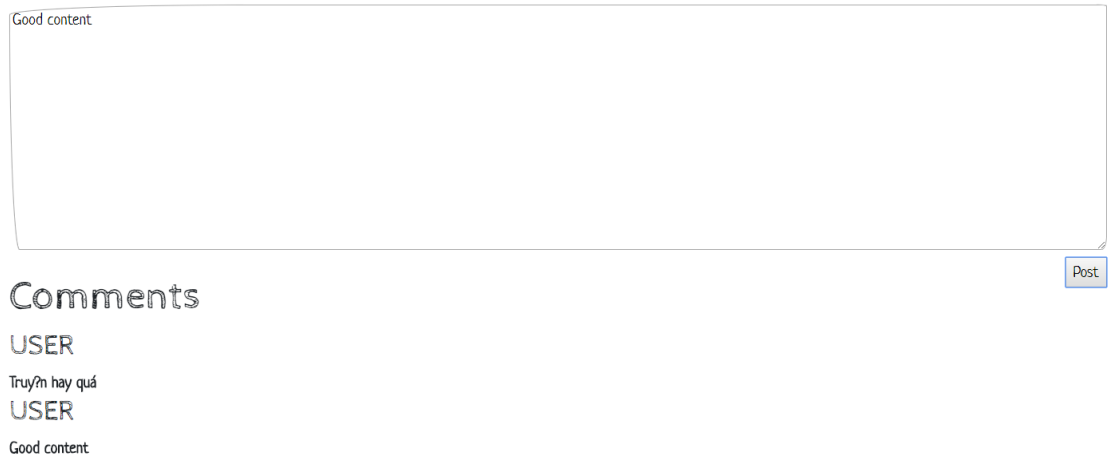


Hình 14: Giao diện màn hình truyện đang theo dõi

- Gồm 1 table hiển thị các truyện đang theo dõi khi người dùng đã đăng nhập, gồm các thông tin:
  - Tên truyện: Hiển thị tên truyện
  - Chương mới nhất: Hiển thị chương mới nhất của truyện giúp người dùng đọc chương mới nhất



## - Giao diện bình luận:



Good content

Post

### Comments

USER

Truyện hay quá

USER

Good content

Hình 15: Giao diện bình luận

Thành phần	Loại	Mô tả
Khung nhập nội dung	TextArea	Nhập nội dung muốn bình luận, hiển thị khi người dùng đã đăng nhập
Nút post	Button	Đăng bình luận, hiển thị khi người dùng đã đăng nhập
Danh sách bình luận	List	Hiển thị các bình luận gồm: username người bình luận, nội dung bình luận

## KẾT LUẬN

### ❖ Kết quả đạt được:

Website về cơ bản đã thỏa mãn được các yêu cầu của một website giúp người đọc tiếp cận với truyện tranh, bao gồm Tạo tài khoản, đăng nhập, xem truyện, tìm truyện, đăng kí theo dõi truyện.

### ❖ Hạn chế:

Do thời gian ngắn và chương trình đào tạo thực tập khá tổng quát, cùng với việc tìm hiểu khá nhiều công nghệ cho nên việc xảy ra nhiều lỗi làm cho quá trình hoàn thành bị chậm. Bên cạnh đó một số các chức năng vẫn cần bổ sung để trở thành một website hoàn chỉnh hơn

### ❖ Hướng phát triển:

- Database truyện cứng, cần thêm tính năng update truyện
- Cần tạo ra hai vai trò mới: Admin và contributor với mục đích:
  - o Admin: Quản lý toàn bộ website(truyện tranh, người đọc, contributor)
  - o Contributor: Dịch và đăng truyện mới, quản lý truyện đã đăng
- Tối ưu giao diện và logic để cải thiện trải nghiệm người dùng

**TÀI LIỆU THAM KHẢO:**

1. Hướng dẫn cài đặt công cụ, môi trường: <https://o7planning.org>
2. Tài liệu Angular: <https://angular.io/docs>
3. Tài liệu Spring Framework: <https://spring.io/guides>
4. Diễn đàn trao đổi, thảo luận: <https://stackoverflow.com>