**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**<NGUYỄN CHU PHƯỚC ÂN>**

**<HỆ THỐNG …>**

**ĐỒ ÁN NGÀNH**

**NGÀNH <CÔNG NGHỆ THÔNG TIN>**

**TP. HỒ CHÍ MINH, <2023>BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**<NGUYỄN CHU PHƯỚC ÂN>**

**< HỆ THỐNG … >**

**Mã số sinh viên: <2051052006>**

**ĐỒ ÁN NGÀNH**

**NGÀNH <CÔNG NGHỆ THÔNG TIN>**

**Giảng viên hướng dẫn: <DƯƠNG HỮU THÀNH>**

**TP. HỒ CHÍ MINH, <2023>**

LỜI CẢM ƠN

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

TÓM TẮT ĐỐ ÁN NGÀNH

ABSTRACT

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc146402542)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 2](#_Toc146402543)

[TÓM TẮT ĐỐ ÁN NGÀNH 3](#_Toc146402544)

[ABSTRACT 4](#_Toc146402545)

[DANH MỤC TỪ VIẾT TẮT 7](#_Toc146402546)

[DANH MỤC HÌNH VẼ 8](#_Toc146402547)

[DANH MỤC BẢNG 9](#_Toc146402548)

[MỞ ĐẦU 10](#_Toc146402549)

[Chương 1. Giới thiệu 11](#_Toc146402550)

[1.1. Giới thiệu đề tài 11](#_Toc146402551)

[1.2. Mục tiêu đề tài 11](#_Toc146402552)

[1.3. Đối tượng đề tài 11](#_Toc146402553)

[1.4. Bố cục báo cáo 12](#_Toc146402554)

[Chương 2. CƠ SỞ LÝ THUYẾT 13](#_Toc146402555)

[2.1. Spring Boot 13](#_Toc146402556)

[2.1.1. Kiến trúc 14](#_Toc146402557)

[2.1.2. Các thành phần công nghệ được sử dụng 16](#_Toc146402558)

[2.2. ReactJs 20](#_Toc146402559)

[2.2.1. Kiến trúc 21](#_Toc146402560)

[2.2.2. Các thành phần công nghệ được sử dụng 23](#_Toc146402561)

[2.3. MySQL 28](#_Toc146402562)

[2.3.1. Kiến trúc 29](#_Toc146402563)

[2.3.2. Các thành phần công nghệ được sử dụng 31](#_Toc146402564)

[2.4. Amazon Web Service (AWS) 31](#_Toc146402565)

[2.4.1. Kiến trúc 32](#_Toc146402566)

[2.4.2. Các thành phần công nghệ được sử dụng 32](#_Toc146402567)

[Chương 3. HỆ THỐNG…. 32](#_Toc146402568)

[3.1. Giới thiệu 32](#_Toc146402569)

[3.2. Kiến trúc của hệ thống 32](#_Toc146402570)

[3.3. Phân tích và thiết kế hệ thống 32](#_Toc146402571)

[3.3.1. Phân tích use case 32](#_Toc146402572)

[3.3.2. Thiết kế giao diện [optional] 33](#_Toc146402573)

[3.3.3. Thiết kế CSDL 33](#_Toc146402574)

[3.3.4. Các nghiệp vụ của đề tài 33](#_Toc146402575)

[Chương 4. Kết luận và hướng phát triển 34](#_Toc146402576)

[4.1. Kết luận 34](#_Toc146402577)

[4.2. Hướng phát triển 34](#_Toc146402578)

[TÀI LIỆU THAM KHẢO 36](#_Toc146402579)

[PHỤ LỤC 37](#_Toc146402580)

DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| STT | Từ viết tắt | Từ đầy đủ |
| 1 | CSDL | Cơ sở dữ liệu |
| 2 | RDBMS | Relational database management system |
| 3 | ORM | Object Relational Mapping |
| 4 | JDBC | Java Database Connectivity |
| 5 | AWS | Amazon Web Service |
| 6 | Amazon S3 | Amazon Simple Storage Service |

DANH MỤC HÌNH VẼ

[Hình 2.1: Spring Framework modules 13](#_Toc146402524)

[Hình 2.2: Giao diện trang web tạo ứng dụng Spring Boot đơn giản 14](#_Toc146402525)

[Hình 2.3: Các lớp trong kiến trúc Spring Boot 15](#_Toc146402526)

[Hình 2.4: Workflow của Spring Boot 15](#_Toc146402527)

[Hình 2.5: Kiến trúc của JDBC 17](#_Toc146402528)

[Hình 2.6: Kiến trúc Hibernate 18](#_Toc146402529)

[Hình 2.7: Spring Security workflow 19](#_Toc146402530)

[Hình 2.8: Giao diện tải Node.js 21](#_Toc146402531)

[Hình 2.9: Stateful và stateless component 22](#_Toc146402532)

[Hình 2.10: Minh họa quá trình so sánh giữa Virtual DOM và DOM thật 22](#_Toc146402533)

[Hình 2.11: Luồng dữ liệu của React 23](#_Toc146402534)

[Hình 2.12: Luồng chạy của useEffect 25](#_Toc146402535)

[Hình 2.13: Quá trình hoạt động của Redux 26](#_Toc146402536)

[Hình 2.14: Quá trình hoạt động của Redux khi có Redux-Saga 27](#_Toc146402537)

[Hình 2.15: Giao diện cài đặt MySQL 29](#_Toc146402538)

[Hình 2.16: Kiến trúc của MySQL 30](#_Toc146402539)

DANH MỤC BẢNG

[Bảng 1.1: Tên bảng 1 10](#_Toc367742567)

[Bảng 2.1: Tên bảng 1 11](#_Toc367742568)

MỞ ĐẦU

# Giới thiệu

## Giới thiệu đề tài

Trong xã hội nơi mà công nghệ đang ngày càng phát triển, số lượng file (hình ảnh, video, ...) tăng càng ngày càng nhiều và mang rất nhiều giá trị từ giải trí đến chính trị. Thế nhưng có rất nhiều file giá trị như vậy mà chỉ được lưu trữ lại ở đâu đó mà không thể chuyển đổi lại thành giá trị kinh tế. Và đây cũng là lý do hệ thống này được sinh ra. Với hệ thống này, mọi người ở khắp mọi nơi chỉ đơn giản là upload file này lên hệ thống cùng với giá tiền mong muốn và file này sẽ được giao bán trên hệ thống giống như Tiki, Shopee, … nhưng sản phẩm ở đây là file. Hệ thống không những có đầy đủ chức năng mà một hệ thống thương mại điện tử có mà còn cung cấp nhiều tính năng để làm việc với file (hiện tại chỉ hỗ trợ file có đuôi .jpg và .png) như giải nén, kiểm tra chất lượng ảnh, …

## Mục tiêu đề tài

Theo mục đích như được trình bày ở trên, trên thị trường đã có rất nhiều nền tảng được ra mắt như Pixabay, ShuttleSock, … được tạo ra để tạo ra một môi trường cho phép người dùng có thể “bán” file của mình và tạo ra lợi nhuận từ nó. Thể nhưng những nền tảng này lại không dễ dàng cho người Việt Nam do để thanh toán hay đăng bán file của mình trên những nền tảng này thì phải sở hữu thẻ thanh toán quốc tế và những vấn đề pháp lý, quả trình giao dịch, … ở nước ngoài đôi khi không phù hợp với người dân nước ta. Đó cũng là lý do tôi phát triển hệ thống này, với ý tưởng được dựa trên những nền tảng đã có sẵn trên thị trường nhưng nó chủ yếu nhắm đến thị trường Việt Nam và cung cấp những tính năng phù hợp với người Việt như ngôn ngữ, phương thức thành toán quen thuộc như Momo, ngân hàng trong nước, … mà không cần sử dụng đến Visa hay Paypal.

## Đối tượng đề tài

Về đối tượng của hệ thống này, tất cả mọi người có file cần chuyển đổi về mặt kinh tế đều là đối tượng của hệ thống này, đặc biệt là ở quốc gia Việt Nam. Như đã trình bày ở trên, người dùng chỉ đơn giản upload file lên cùng với giá tiền mong muốn, việc còn lại sẽ là nhiệm vụ của hệ thống. Nhưng người dùng cần phải đảm bảo phải tôn trọng quyền sở hữu của bất kì loại file nào và không mang tính tiêu cực, cực đoan. Trong tương lai hệ thống sẽ phát triển tính năng kiểm tra bản quyền cùng những hành vi vi phạm và xử lý sai phạm.

## Bố cục báo cáo

Ở chương 1, tôi đã giới thiệu cũng như trình bày mục tiêu và đối tượng được nhắm đến của hệ thống này. Chúng ta đã có một cái nhìn tổng quan về hệ thống và những lợi ích mà nó sẽ mang lại.

Chương 2, tôi sẽ trình bày về các công nghệ được tôi áp dụng vào hệ thống này. Cụ thể, tôi sẽ trình bày về sự sử dụng Reactjs để tạo ra giao diện cho người dùng, Spring Boot đóng vai trò là Server tiếp nhận request của người dùng, MySQL là CSDL để lưu trữ dữ liệu và cuối cùng là Amazon S3 là dịch vụ của Amazon dùng để lưu trữ các file của khách hàng.

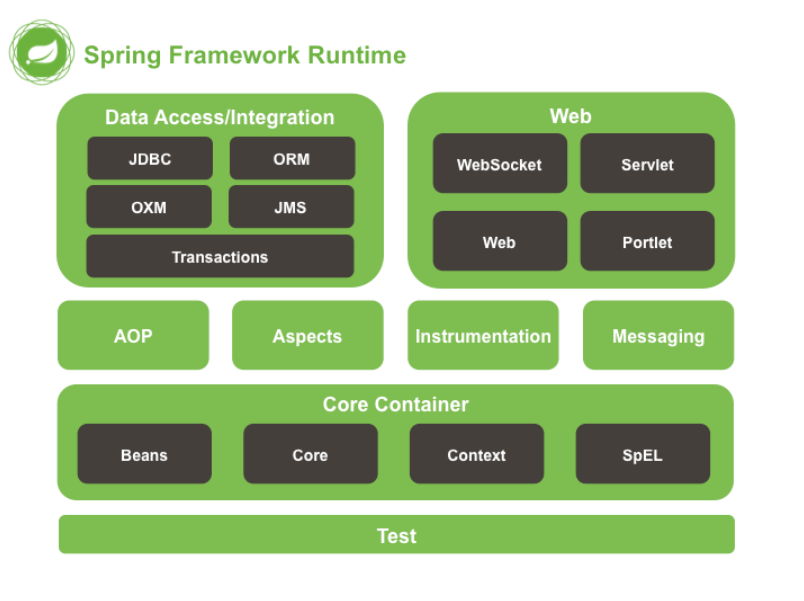
Chương 3, tôi sẽ trình bày chi tiết về hệ thống cũng như cách hệ thống hoạt động. Các sơ đồ usecase diagram, sequence diagram, class diagram cũng sẽ được trình bày ở đây.

Chương 4, tôi sẽ đánh giá ưu và nhược điểm của hệ thống cũng như của bản thân tôi. Đồng thời, tôi sẽ trình bày những tính năng dự kiến sẽ tích hợp trong tương lai.

# CƠ SỞ LÝ THUYẾT

## Spring Boot

Để hiểu được bản chất của Spring Boot và vì sao Spring Boot trở nên phổ biến, tôi sẽ trình bày về Spring Framework đầu tiên. Trong Java, thuật ngữ “Spring Framework” không quá xa lạ với các lập trình viên, nó là một framework mã nguồn mở được thiết kế theo hai design pattern chính là Dependency Injection (tiêm phụ thuộc) và Aspect Oriented Programming (lập trình hướng khía cạnh) giúp giảm sự phụ thuộc giữa các thành phần và giảm sự phức tạp trong việc cấu hình. Cùng với tính gọn nhẹ và trong suốt của mình, Spring Framework đã trở thành sự lựa chọn tuyệt vời để phát triển những ứng dụng cần hiệu suất cao. Spring Framework bao gồm khoảng 20 modules khác nhau và được phân ra thành 8 nhóm riêng biệt, chi tiết được thể hiện ở hình ảnh dưới đây:

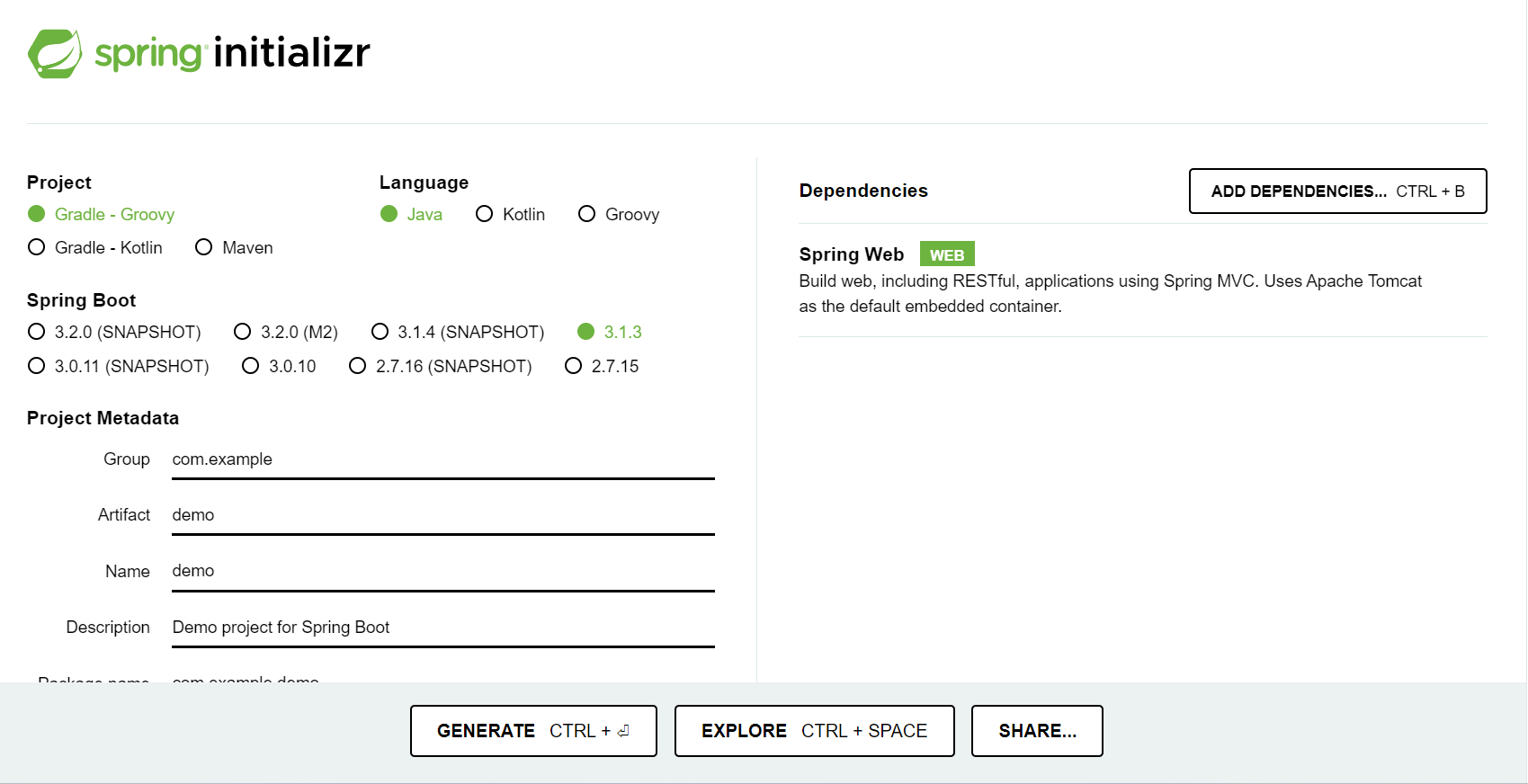


Hình 2.1: Spring Framework modules

Nhờ đó, Spring MVC, được xây dựng trên nền tảng của Spring Framework và được thiết kế theo mô hình MVC, đã trở nên hữu ích và được sử dụng rất nhiều để tạo ra các ứng dụng web chạy bằng Java. Nhưng nhược điểm của nó là sự phức tạp trong việc cấu hình nên từ đó Spring Boot đã ra đời.

Spring Boot là một trong những dự án trong hệ sinh thái Spring và càng ngày càng trở nên phổ biến hơn không chỉ nhờ vào việc được xây dựng dựa trên Spring Framework mà nó còn có những tính năng hữu ích như tự động cấu hình, tạo ứng dụng độc lập có thể chạy ngay mà không cần cấu hình nhiều. Nhờ đó, việc tạo và chạy một ứng dụng Spring Boot có thể chỉ diễn ra trong vài phút.

Để tạo một ứng dụng web Spring Boot một cách đơn giản, chúng ta chỉ cần truy cập trang web Spring initializr như hình ở bên dưới:

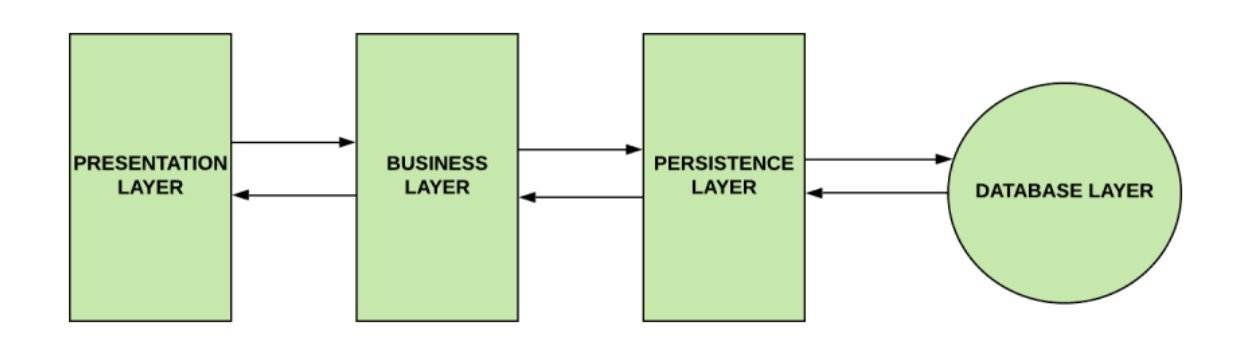


Hình 2.2: Giao diện trang web tạo ứng dụng Spring Boot đơn giản

Sau đó hãy chọn những tùy chọn và dependencies phù hợp và trong trường hợp này tôi sẽ chọn “Spring web” dependency. Tiếp theo chỉ cần bấm nút “Generate” là chúng ta đã có một ứng dụng web Spring Boot có thể chạy ngay.

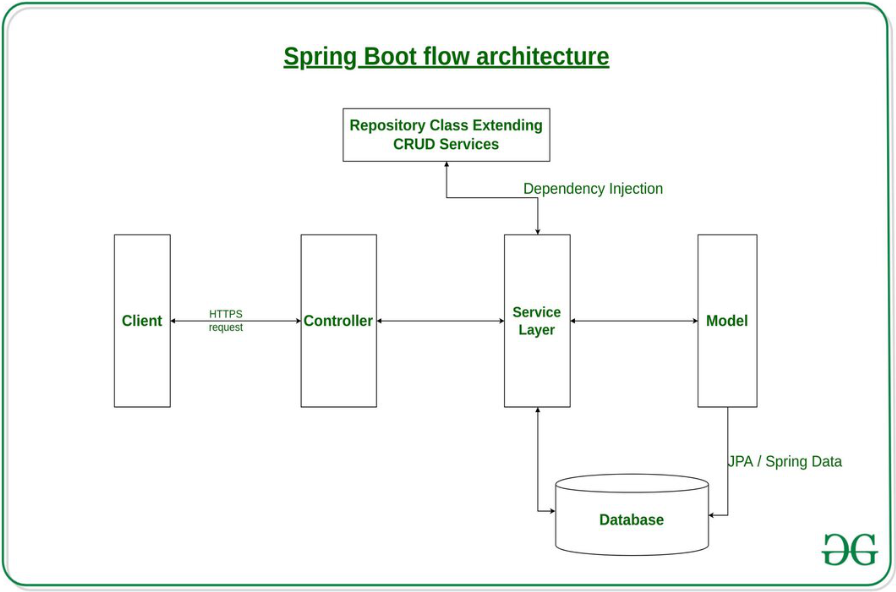
### Kiến trúc

Kiến trúc Spring Boot được thiết kế theo mô hình phân lớp và thường có 4 lớp như sau:



Hình 2.3: Các lớp trong kiến trúc Spring Boot

Lớp Presentaion là lớp đầu tiên và là lớp đảm nhiệm xử lý các HTTP request từ phía người dùng, quá trình xác thực người dùng (authentication) và chuyển đổi dữ liệu từ json sang java object và ngược lại. Lớp Business là lớp thứ hai, lớp này đảm nhiệm việc kiểm tra tính hợp lệ của dữ liệu (validation), quá trình ủy quyền người dùng (authorization) và thực hiện các logic nghiệp vụ. Lớp Persistence là lớp thứ 3 chứa các logic truy vấn, chuyển java object thành hàng dữ liệu trong database và ngược lại. Lớp Database là lớp cuối cùng và thực hiện các tương tác với database. Và đây là workflow của Spring Boot:



Hình 2.4: Workflow của Spring Boot

Đầu tiên, người dùng tương tác với giao diện của hệ thống như click vào một cái nút, … Sau đó, một Http request sẽ được gửi tới Controller và nó sẽ phân tích request đó và sẽ gọi Service tương ứng để xử lý request theo logic nghiệp vụ. Trong trường hợp Service cần dữ liệu hay thay đổi dữ liệu thì nó sẽ lấy dữ liệu từ database thông qua các Repository và dữ liệu được thể hiện thông qua các Model. Sau khi xử lý xong, Service trả kết quả về cho Controller và Controller trả về Http response cho người dùng.

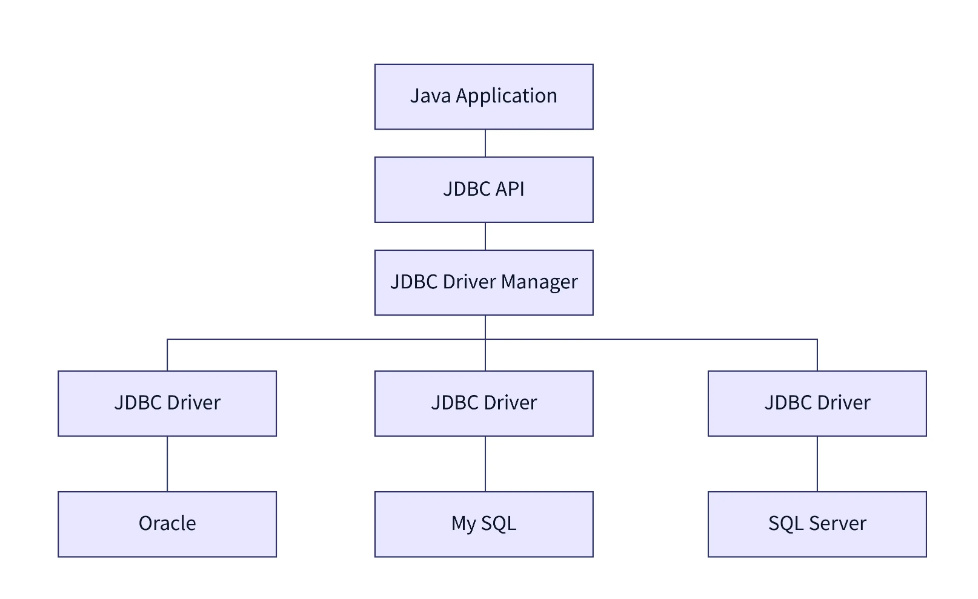
### Các thành phần công nghệ được sử dụng

Hệ sinh thái Spring rất rộng lớn nên ở dự án này tôi sẽ chỉ sử dụng các công nghệ phục vụ cho mục đích của mình và sau đây, tôi sẽ trình bày những công nghệ được tôi sử dụng:

#### JDBC

Trong bất kì hệ thống nào, việc kết nối, thực hiện các câu lệnh truy vấn SQL và xử lý kết quả trả về của CSDL luôn là một trong những vấn đề quan trọng cần được quản lý cẩn thận. Và JDBC (Java Database Connectivity) là API của Java đảm nhiệm nhiệm vụ đó. Nó bao gồm các interface và class được viết bởi ngôn ngữ Java cho phép ứng dụng gửi yêu cầu tới CSDL.

Kiến trúc của JDBC gồm hai phần chính là JDBC API và JDBC Driver Manager như hình ảnh dưới đây:



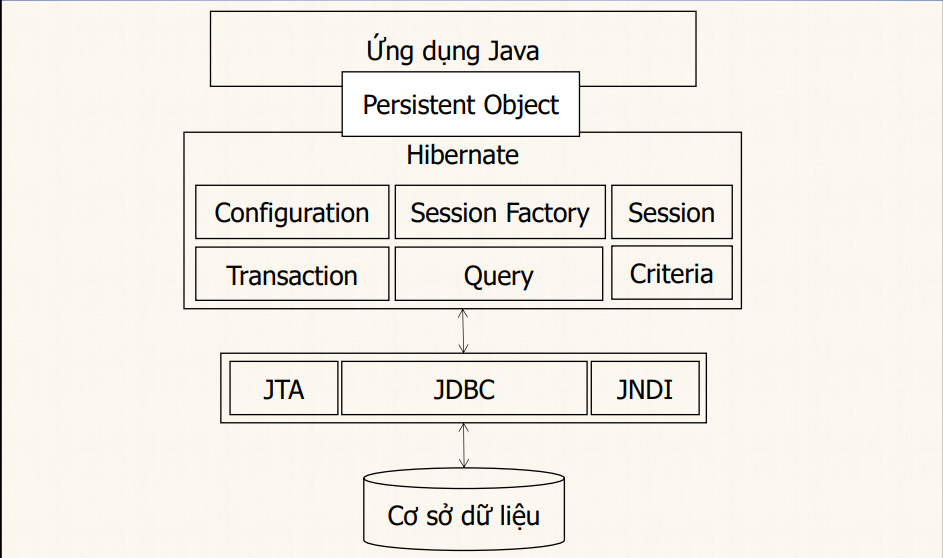
Hình 2.5: Kiến trúc của JDBC

JDBC API cung cấp các interface và class đảm nhiệm việc thực thi các câu lệnh truy vấn SQL và xử lý dữ liệu được trả về.

JDBC Driver Manager chịu trách nhiệm kết nối chương trình với CSDL bằng cách tải và quản lý trình điểu khiển của chính CSDL đó.

#### Hibernate

Trong quá trình tạo ra sản phẩm, việc truy vấn với CSDL là việc vô cùng quan trọng và tốn rất nhiều công sức bởi mỗi hệ quản trị cơ sở dữ liệu quan hệ SQL lại có những cú pháp truy vấn khác nhau buộc những lập trình viên phải viết câu truy vấn cho từng loại, ẩn chứa nhiều vấn đề về bảo mật và khiến chương trình trở nên bừa bộn. Do đó, ORM (Object Relational Mapping) sinh ra để giải quyết các vấn đề trên. ORM là một kỹ thuật dùng để tạo ra “sự ràng buộc” hay là “cầu nối” giữa đối tượng trong ngôn ngữ lập trình và CSDL quan hệ, áp dụng nó vào ngôn ngữ Java gọi là JPA (Java Persistence API). JPA là đặc tả của ngôn ngữ Java chứa các interface hoạt động theo ORM. Hibernate chính là một ORM Framework hay nói cách khác là “JPA implementation”. Hibernate tạo ra câu truy vấn SQL và thực thi nó bằng JDBC, cung cấp nhiều tính năng trong việc tương tác với database một cách dễ dàng hơn, các lập trình viên có thể tập trung vào logic hơn là những câu lệnh truy vấn. Hình ảnh dưới đây là kiến trúc của Hibernate:



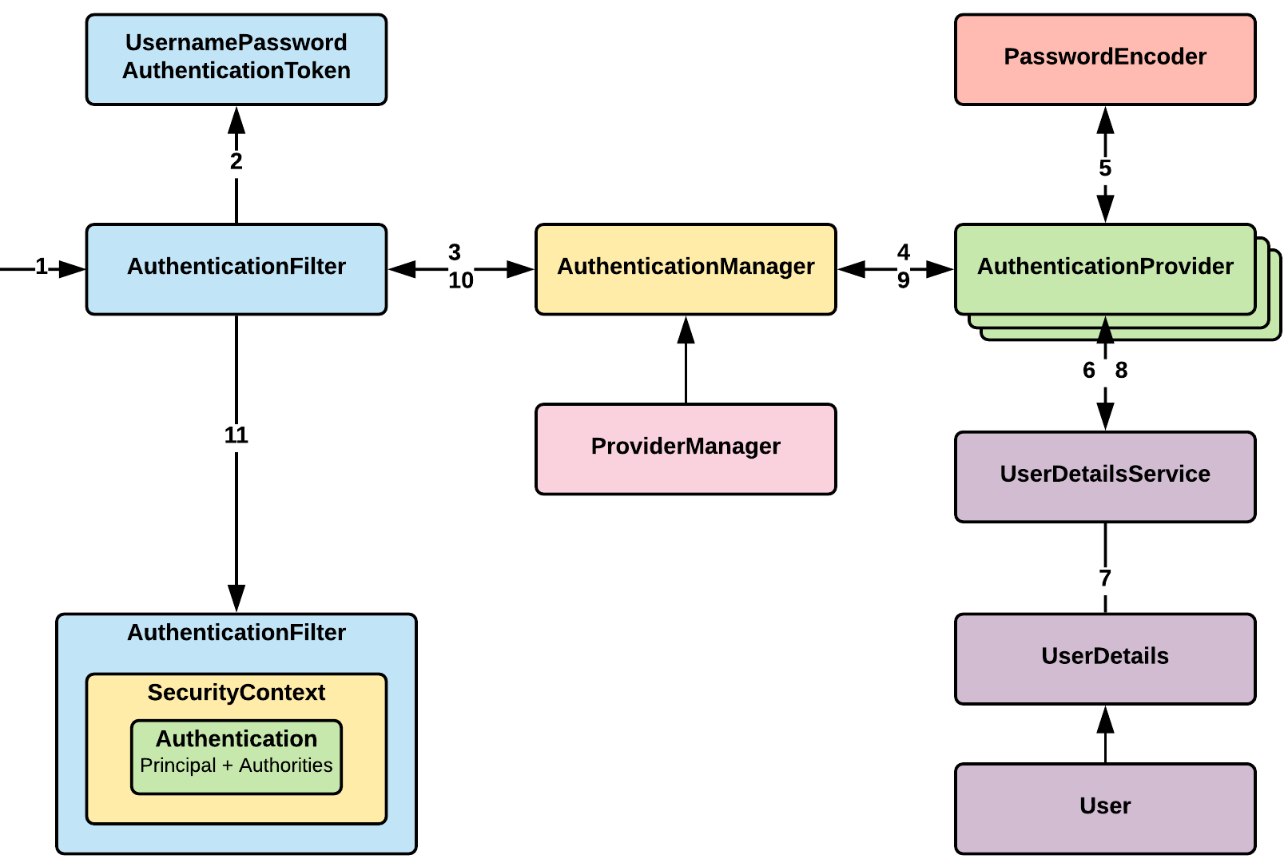
Hình 2.6: Kiến trúc Hibernate

Để cài đặt Hibernate vào dự án, chúng ta cần tải gói Maven Dependency là Spring Boot Starter Data JPA như:

 Ngoài ra, gói này còn cung cấp thêm Spring Data JPA là một dự án con của Spring Data. Nó đã tích hợp sẵn cấu hình mặc định Repository và các hàm truy vấn cơ bản, giúp bạn giảm sự lặp lại và dài dòng trong khi lập trình.

#### Spring Security

Spring Security là một framework đồng thời cũng là một trong những module của Spring Framework đảm nhiệm vai trò xác thực và ủy quyền người dùng khi họ truy cập đến tài nguyên của hệ thống.



Hình 2.7: Spring Security workflow

Khi người dùng gửi request tới hệ thống, request sẽ phải đi qua AuthenticationFilter, nó sẽ chặn hoặc cho request đi tới DispatcherServlet. Trong trường hợp request chứa thông tin về người dùng, nó sẽ lưu thông tin đó vào UsernamePasswordAuthenticationToken chuyển tới AuthenticationManage. Ở đây, AuthenticationManager đảm nhiệm quản lý các AuthenticationProvider như DaoAutenticationProvider, JwtAuthenticationProvider, … và sẽ lựa chọn provider phù hợp để xử lý đối tượng UsernamePasswordAuthenticationToken, cuối cùng AuthenticationProvider sẽ là lớp thật sự thực hiện xác thực người dùng. Nó sử dụng UserDetailsService để tìm kiếm thông tin người dùng trong CSDL và trả về UserDetail chứa thông tin của người dùng như username, password, role, … PasswordEncoder đảm nhiệm việc mã hóa mật khẩu người dùng.

Để tích hợp Spring Security vào dự án, chúng ta cần thêm gói thuộc Maven Dependency là spring-boot-starter-security là:



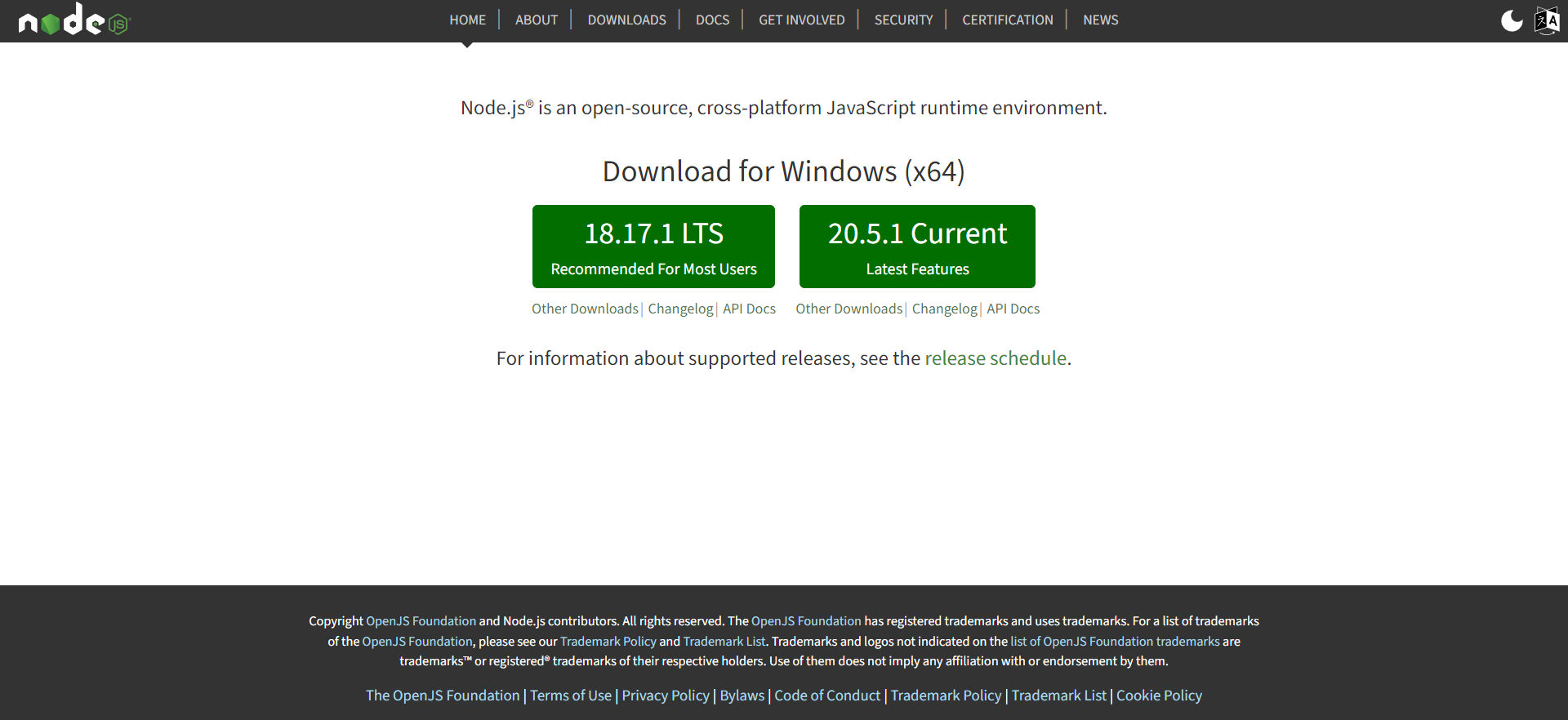
## ReactJs

Trong bất kì một hệ thống nào, việc tương tác với người dùng là một việc vô cùng quan trọng. Việc giao tiếp với hệ thống thông qua các dòng lệnh là điều rất phức tạp và khó hiểu với người sử dụng. Chính vì vậy, một giao diện thân thiện luôn là một giải pháp được đầu tư và thư viện Reactjs sinh ra để thực hiện nó.

ReactJs là một thư viện Javascript mã nguồn mở miễn phí dùng để xây dựng giao diện người dùng dựa trên component hay còn được gọi là Component-based software engineering (CBSE). Ban đầu, Reactjs được tạo ra vào năm 2011 bởi Facebook và chỉ được sử dụng nội bộ. Cho đến năm 2013, nó mới trở thành mã nguồn mở như bây giờ và vẫn được duy trì bởi Meta (hay còn được gọi là Facebook) cùng với cộng đồng lập trình viên.

Nhờ vào những ưu điểm tuyệt vời của ReactJs mà đã và đang phổ biến cho đến thời điểm hiện tại (năm 2023). Một trong những ưu điểm của ReactJs chính là hệ sinh đa dạng của nó, kết hợp và hoạt động tốt với các ngôn ngữ backend khác đồng thời cũng có nhiều thư viện ngoài hỗ trợ, có tính tái sử dụng cao, hiệu suất tốt do có sự tương tác chủ yếu trên Virtual DOM thay vì DOM thật, cộng đồng lớn, …

Để cài đặt một dự án Reactjs, đầu tiên ta cần phải cài đặt [Node.js](https://nodejs.org/en) phù hợp với hệ điều hành của máy tính. Dưới đây là trang web tải Node.js:



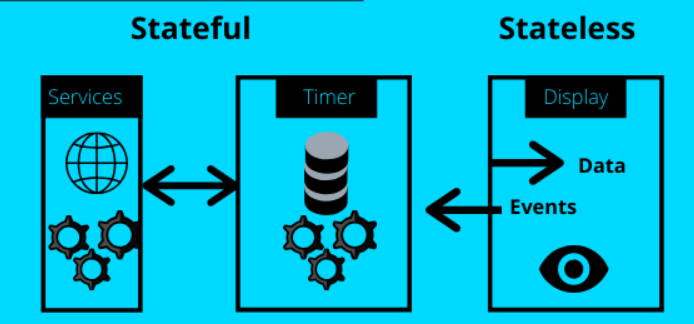
Hình 2.8: Giao diện tải Node.js

Sau đó, thực hiện lệnh sau trong Terminal của bạn: “npx create-next-app appname” và sau đó một dự án reactjs sẽ được tạo ra trên máy của bạn.

### Kiến trúc

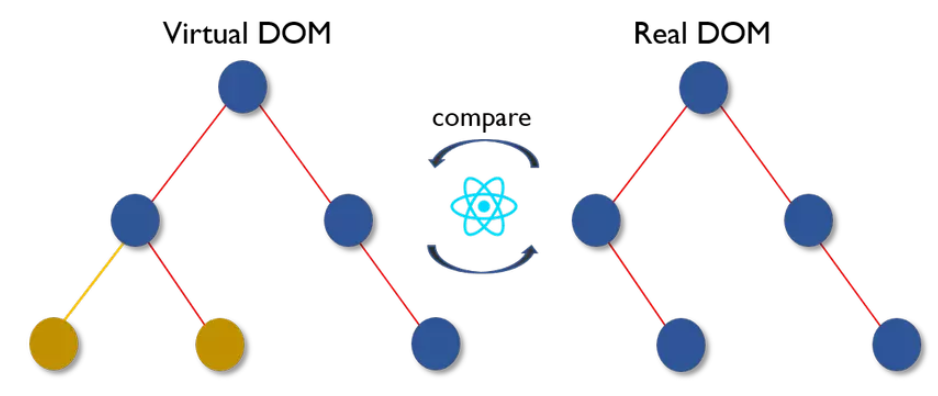
Sau khi tham khảo nhiều tài liệu liên quan, tôi nhận thấy rằng mỗi một tài liệu đều có sự trình bày khác nhau về kiến trúc của ReactJs. Nhưng về cơ bản, kiến trúc của ReactJs chủ yếu dựa trên component, Virtual DOM và one-way-data-flow

Đầu tiên là component, thành phần cơ bản của những hệ thống website được tạo ra từ Reactjs. Chúng giống như những viên gạch khi chúng ta xếp chúng theo một bản thiết kế thì sẽ tạo nên một ngôi nhà và trong trường hợp này chính là giao diện người dùng. Mỗi component đều có nhiệm vụ riêng của chúng và được phân ra thành 2 loại đó là Stateful component và Stateless component. Stateful component hay còn có thể hiểu là parent component hay “thùng chứa các component khác”. Chúng thường có nhiệm vụ lưu trữ những thông tin của component, thường được tải bất đồng bộ từ các service (lưu trữ trong state) và truyền dữ liệu cho các component con bên trong nó đồng thời trả lời cho câu hỏi cách mà component này hoạt động. Stateless component ngược lại với Stateful component, chúng thường được gọi là child component. Chúng nhận dữ liệu được truyền từ component cha (thông qua props) và thể hiện nó lên giao diện. Như vậy, Stateless component sẽ thể hiện component sẽ trông như thế nào.



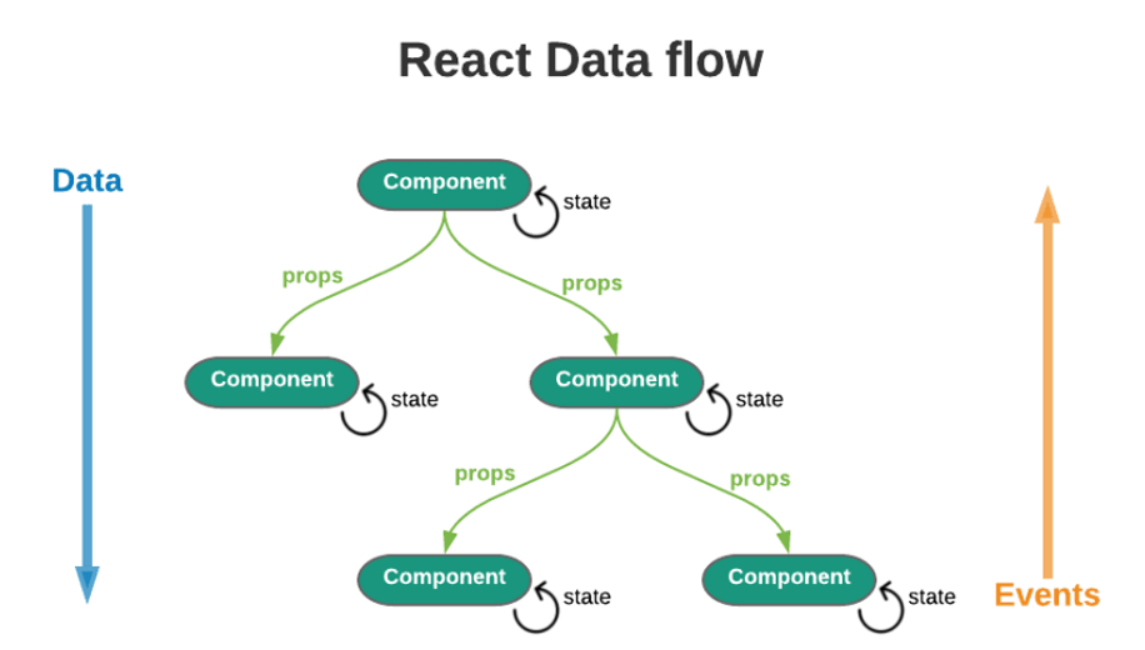
Hình 2.9: Stateful và stateless component

Tiếp theo là Virtual DOM, hay còn được gọi là “DOM ảo”, nó được coi là một sự trừu tượng của DOM thật. Khi một thành phần trong website thay đổi, một bản copy Virtial DOM sẽ được tạo ra và được so sánh với DOM hiện tại để tìm ra thành phần nào bị thay đổi. Sau đó, nó sẽ chỉ cập nhập duy nhất thành phần bị thay đổi trong DOM thật thay vì cập nhập toàn bộ DOM. Đây cũng là lý do vì sao Virtual DOM cung cấp cho Reactjs khả năng tải trang cực kỳ nhanh, tăng tính trải nghiệm sản phẩm của người dùng.



Hình 2.10: Minh họa quá trình so sánh giữa Virtual DOM và DOM thật

Cuối cùng là one-way-data-flow, hay còn được gọi là luồng dữ liệu một chiều. Trong ReactJs luồng dữ liệu là một chiều, cụ thể là từ component cha xuống component con như hình ảnh sau:



Hình 2.11: Luồng dữ liệu của React

Như ta có thế thấy trên hình 2.11, luồng dữ liệu chỉ đi theo một chiều từ trên xuống dưới thông qua các props và các dữ liệu này sẽ không bị thay đổi. Nhờ vậy, việc kiểm soát dữ liệu trong ReactJs trở nên đơn giản và khi xảy ra lỗi thì có thể dễ dàng đoán được nguyên nhân ở đâu. Khi component con có nhu cầu thay đổi dữ liệu, chúng sẽ phải truyền “sự kiện” lên component cha và việc cập nhật lại state cũng là do component cha đảm nhiệm, sau đó lại tiếp tục truyền dữ liệu mới xuống lại cho component con thông qua props.

### Các thành phần công nghệ được sử dụng

#### React hooks

Hooks trong ReactJs cho phép bạn sử dụng nhiều tính năng khác trong component của bạn. Sau đây là những hook đã được áp dụng trong dự án này:

UseState giúp lưu lại trạng thái của một component. Nó sẽ trả về một mảng gồm 2 phần tử là giá trị của state và hàm setter giá trị, mỗi khi state thay đổi giá trị thì component sẽ được tải lại (re-render). Khi sử dụng state, chúng ta cần truyền giá trị mặc định cho nó. Nếu không, giá trị mặc định sẽ là null. Dưới đây là cú pháp khai báo:

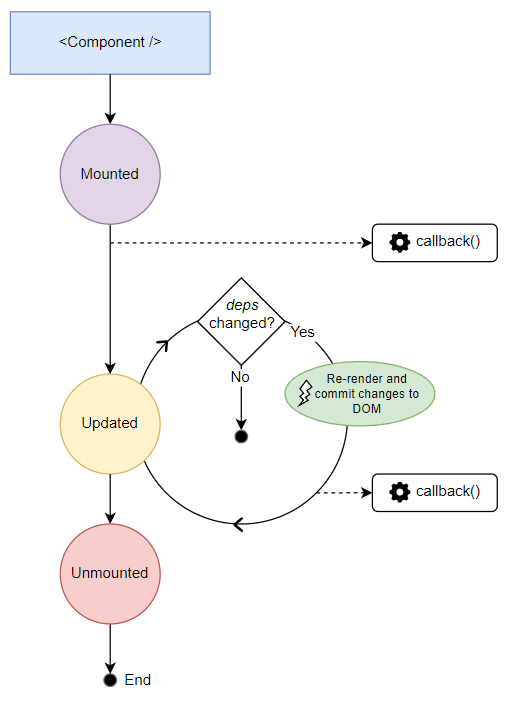


UseRef giúp chúng ta có thể tương tác với thành phần của DOM một cách dễ dàng hơn. Đồng thời, trong một số trường hợp nó cũng có thể thay thế useState mà không cần phải tải lại component. Giống như state, useRef nhận một tham số làm giá trị hiện tại và trả về một object có thuộc tính current là thành phần của DOM. Sau đây là cú pháp:



UseEffect thường được sử dụng cho việc tải dữ liệu bất đồng bộ, nó nhận vào hai tham số là một hàm callback và một mảng dependencies. Hàm callback là hàm sẽ được gọi sau khi component return xong và mảng dependencies chứa những biến (thường là state) để kiểm soát việc khi nào hàm callback được thực hiện. Hàm callback chỉ được thực hiện khi giá trị trong mảng denpendencies thay đổi. Đặc biệt, hàm callback có thể trả về một function khác, thường được gọi là “clean up function”, hàm này sẽ được gọi trước khi component được tải lại (re-render). Cách sử dụng useEffect:

Luồng chạy của useEffect khác phức tạp cho những người lần đầu tiếp xúc nên tôi sẽ minh họa bằng một hình ảnh:



Hình 2.12: Luồng chạy của useEffect

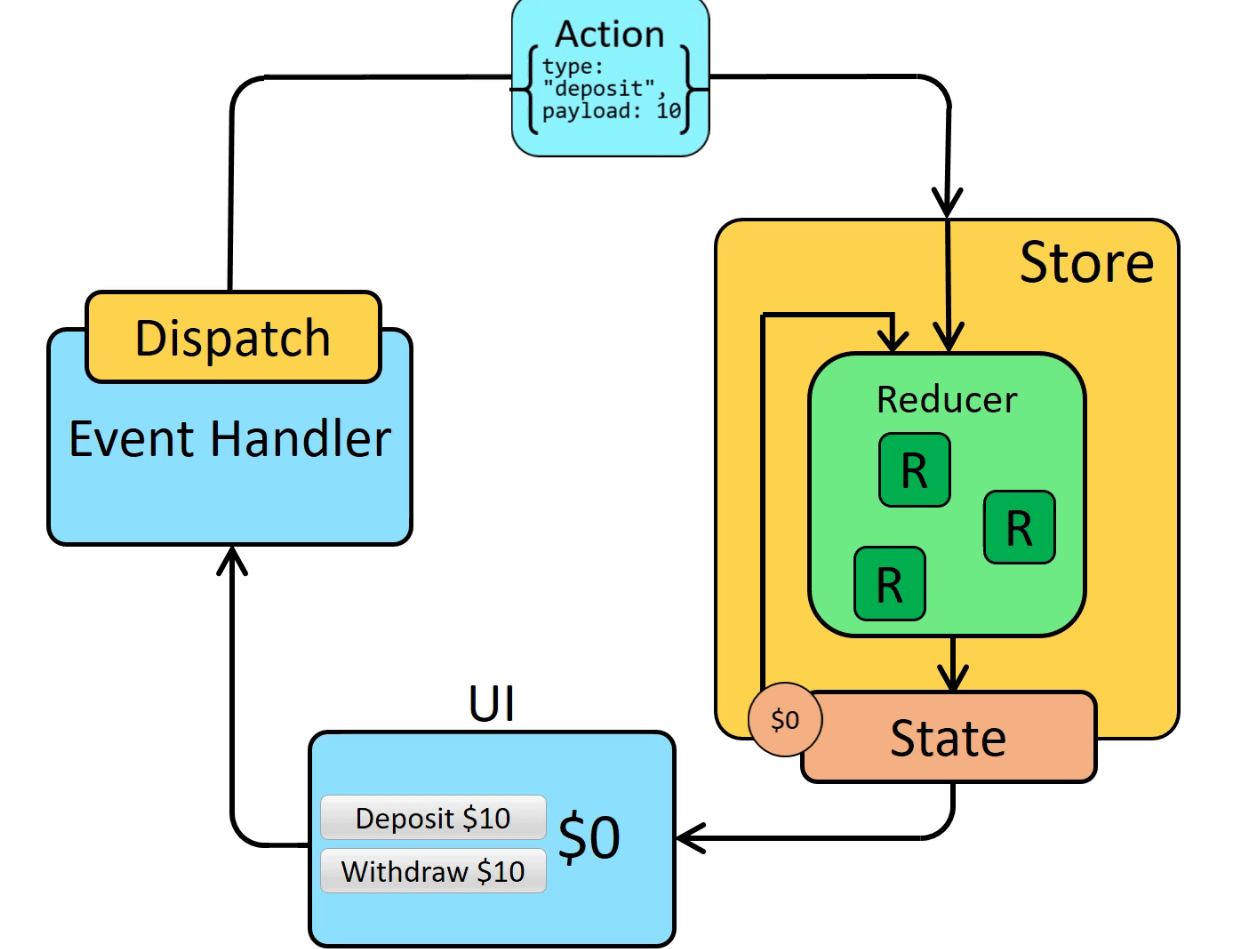
Trong đó, mounted, updated, unmounted là các giai đoạn liên quan đến vòng đời của một component tương đương với quá trình khởi tạo, cập nhật và loại bỏ một component.

UseMemo thường được dùng cho việc ghi nhớ kết quả trả về của một hàm, không cần phải tính toán lại khi không cần thiết, giúp giảm công việc cho mỗi lần re-render của component. Cũng giống như useEffect, nó nhận vào hai tham số là hàm callback và mảng dependencies với tác dụng tương tự. Dưới đây là cách sử dụng:



#### Redux

Redux là công cụ quản lý trạng thái (state) giúp cho ứng dụng chạy bằng Javascript có thể quản lý được máy khách, máy chủ một cách nhất quán và có thể kiểm tra lỗi một cách dễ dàng. Nó quản lý bằng cách lưu tất cả các trạng thái vào store và thay đổi bằng cách mỗi khi người dùng tương tác với giao diện website, nó sẽ gửi một sự kiện gọi là dispatch, nó chứa action là hành động làm sẽ dùng để thay đổi state và payload chứa dữ liệu sẽ được dùng trong việc thay đổi. Reducer sẽ dựa vào “action” để thay đổi, cập nhật lại state trong store và trả về giao diện người dùng. Quá trình thực hiện được miêu tả trong hình 2.13:



Hình 2.13: Quá trình hoạt động của Redux

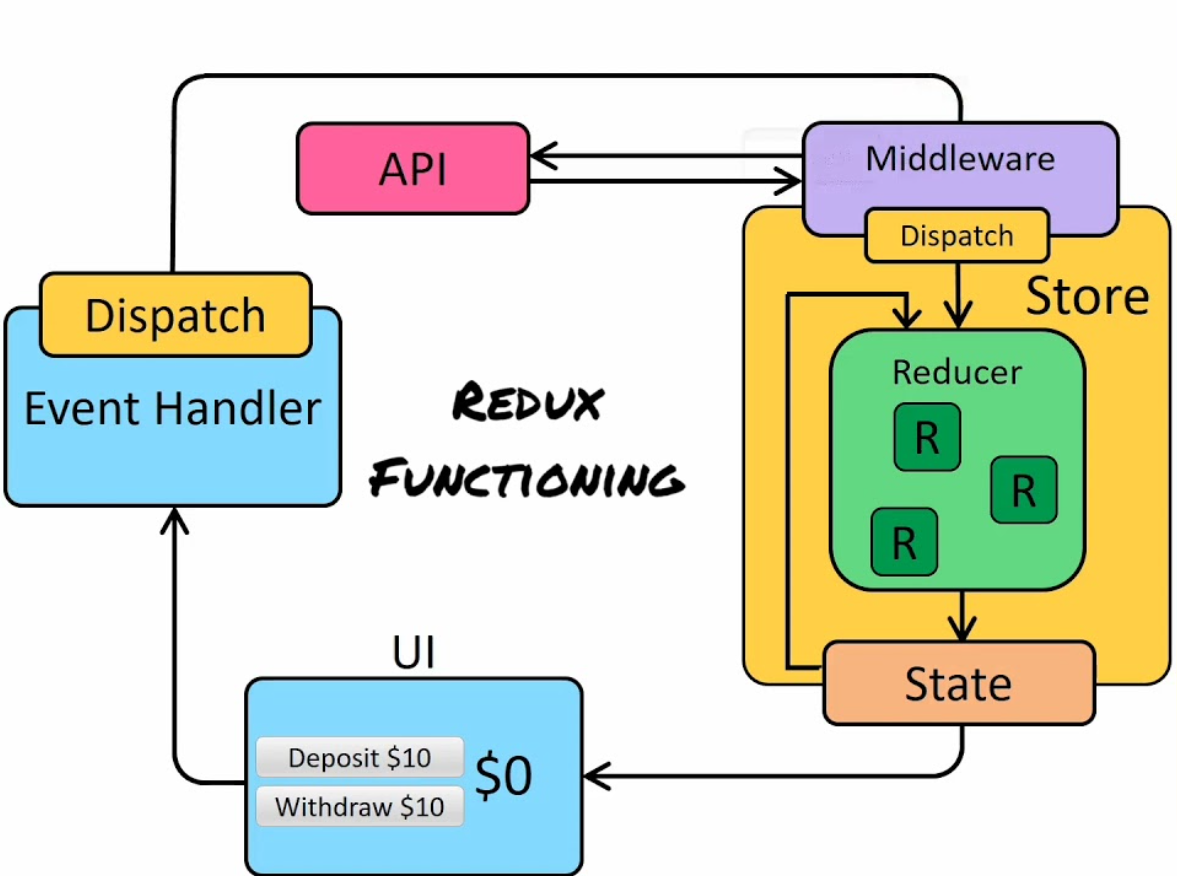
Những ưu điểm của Redux là nó có thể kết hợp với nhiều Javascript framework khác mà không chỉ riêng gì ReactJs và nó tương đối nhẹ chỉ khoảng 2KB. Trong Reactjs, nó một React Hook là useContext có thể làm được việc tương tự đó là quản lý trạng thái nhưng useContext lại sinh ra một vấn đề đó là “useContext hell” và Redux là một trong những giải pháp có thể giải quyết được vấn đề đó.

Cách cài đặt Redux vào dự án ReactJs bằng câu lệnh Terminal sau trong Reactjs App của bạn:



#### Redux-saga

Với Redux, chúng ta có thể quản lý trạng thái khá tốt nhưng nó lại có một nhược điểm đó là chúng ta không thể thực hiện xử lý bất đồng bộ với Redux. Giải pháp để giải quyết vấn đề trên đó là một middleware đứng giữa quá trình trước khi action tới được reducer và xử lý bất đồng bộ ở đó. Và Redux Saga chính là một trong những middleware phổ biên hiện nay. Quá trình được thực hiện như sau:



Hình 2.14: Quá trình hoạt động của Redux khi có Redux-Saga

Cách cài đặt Redux vào dự án ReactJs bằng câu lệnh Terminal sau:



#### Material UI

Material UI là thư viện mở của React được thiết kế tuân theo Google’s Material Design. Nó cung cấp đa dạng component bắt mắt và có tính trải nghiệm người dùng cao bởi vậy nên nó đã trở thành một trong những thư viện React được ưa chuộm nhất, được một cộng đồng lớn liên tục phát triển.

Lý do Material UI dần trở nền phổ biến không chỉ những component đẹp mà còn bởi vì những component này có khả năng tùy chỉnh cao, tùy biến cho nhiều người dùng, nhiều thiết kế.

Cú pháp cài đặt Material UI được cung cấp trên trang chủ như sau:

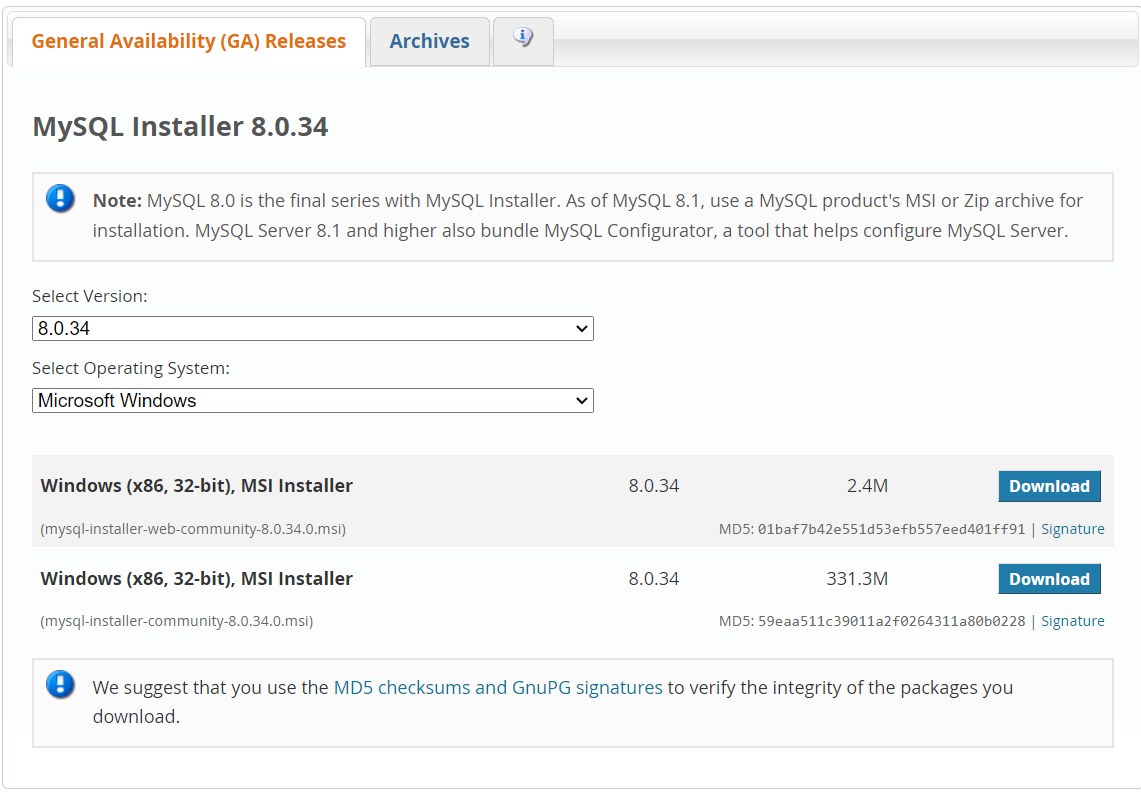


## MySQL

Đầu tiên, chúng ta sẽ tìm hiểu về khái niệm cơ sở dữ liệu quan hệ (Relational database). CSDL quan hệ là CSDL tuân theo mô hình quan hệ dữ liệu. Trong đó, dữ liệu được tổ chức thành các bảng có mối quan hệ với nhau, trong mỗi bảng chứa các trường dữ liệu. Và để hiện thực CSDL quan hệ, hệ quản trị CSDL ra đời (hay còn được gọi là RDBMS).

MySQL là một hệ quản trị CSDL mã nguồn mở rất phổ biến được phát triển và duy trì bởi Oracle Corporation. Một trong những ưu điểm của MySQL chạy rất nhanh, đáng tin cậy, có thể mở rộng và dễ sử dụng, nó có thể hoạt động tốt trên nhiều nền tảng khác nhau và kết nối dễ dàng với nhiều chương trình và thư viện (Spring Boot là một trong số đó)

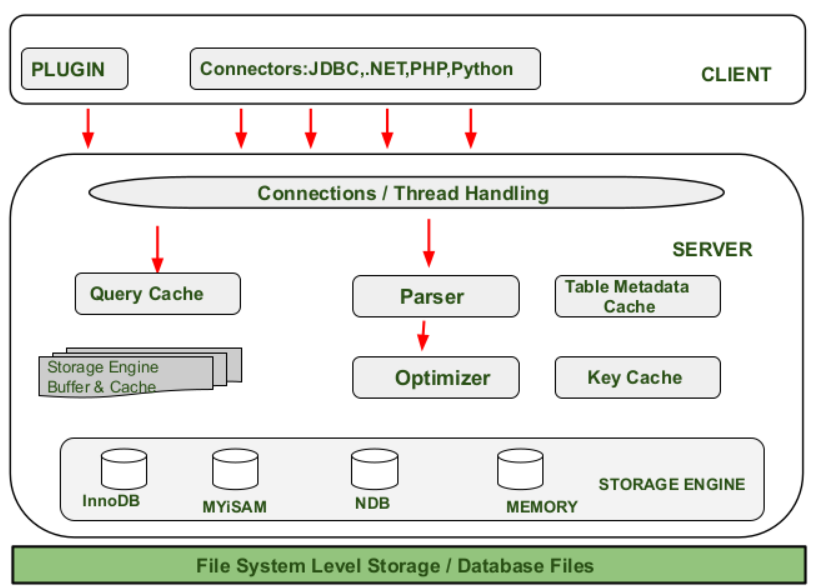
Bạn có thể cài đặt MySQL trên website của nó và chọn phiên bản phù hợp tương ứng với hệ điểu hành của máy tính như hình 2.15 dưới đây:



Hình 2.15: Giao diện cài đặt MySQL

### Kiến trúc

Về cơ bản, kiến trúc của MySQL hoạt động giống mô hình client-server được thể hiện như hình dưới đây:



Hình 2.16: Kiến trúc của MySQL

Đầu tiên là ở Connections, đây là nơi chịu trách nhiệm quản lý kết nối và phân tích câu truy vấn từ phía client, nó sẽ quản lý các cổng kết nối và mỗi khi có một request tới đây, nó sẽ kiểm tra xem có cổng nào đang rảnh không để thực hiện một phiên kết nối giữa người dùng và MySQL gọi là một pool. Ngược lại, request đó sẽ phải chờ đợi cho đến khi có một pool nào đó rảnh. Sau đó, MySQL thực hiện quá trình phân tích câu truy vấn của người dùng thông qua Parser, sau đó sẽ được phân tích lại và trỏ tới bảng phù hợp với câu truy vấn (Optimizer). Và do quá trình này trải qua quá nhiều bước nên để tối ưu hóa sau khi phân tích những câu truy vấn từ người dùng xong, kết quả đó sẽ được lưu trữ trên RAM và mỗi request sau của người dùng, trước khi thực hiện quá trình trên thì MySQL sẽ kiểm tra trong RAM trước xem câu truy vấn hiện tại đã được lưu sẵn chưa, quá trình này gọi là quá trình Query Cache.

### Các thành phần công nghệ được sử dụng

## Amazon Web Service (AWS)

Công nghệ phát triển từng ngày, dữ liệu cũng ngày càng tăng, vậy nên nhu cầu lưu trữ và quản lý dữ liệu trở nên khó khăn đối với các doanh nghiệp. Tự xây dựng một cơ sở hạ tầng của riêng mình tốn rất nhiều chi phí mà không đảm bảo nhiều vấn đề khác như bảo mật, duy trì, … và AWS đã giải được bài toán đó. AWS là nền tảng điện toán đám mây được cung cấp bởi Amazon với hơn 200 tính năng để lưu trữ và tính toán dữ liệu

Một trong những ưu điểm của AWS đầu tiên phải kể đến chính là khả năng tiết kiệm chi phí. Như tôi đã nói ở trên, để tự mình xây dựng một cơ sở hạ tầng thì sẽ tốn rất nhiều chi phí nên AWS đã cung cấp một số dịch vụ như Amazon EC2 để tạo máy ảo và bạn có thể xây dựng cơ sở hạ tầng cloud dành riêng cho doanh nghiệp, dù vậy vấn đề bảo mật của Amazon là điểu doanh nghiệp không cần phải lo lắng. Ngoài ra, AWS còn cung cấp khả năng dễ dàng mở rộng cho cơ sở hạ tầng, việc mở rộng cơ sở hạ tầng vật lý vốn là điều rất phức tạp và tốn nhiều chi phí, nhưng với AWS việc đó lại vô cùng dễ dàng do tiêu chí của AWS là “pay-as-you-go” có thể hiểu là trả theo chi phí mà bạn có, việc còn lại để AWS lo. Khả năng phủ sóng của AWS cũng chính là ưu điểm khó một nền tảng nào có thể cạnh tranh được với Amazon, nó trải dài trên nhiều khu vực như Đông Mỹ, Tây Mỹ, Châu Phi, Châu Á Thái Bình Dương, Châu Âu, … giúp cải thiện tốc độ truyền tải dữ liệu nhanh hơn dù bạn ở đâu trên thế giới. Và trên hết, AWS cung cấp giao diện người dùng tương đối gần gũi giúp doanh nghiệp có thể dễ dàng tương tác mà không cần thông qua những dòng code phức tạp.

Dù cho nhiều ưu điểm là vậy, nhưng AWS cũng có những khuyết điểm mà doanh nghiệp hay người sử dụng không thể tránh. Dù cho AWS đã cung cấp nhiều giao diện tương tác thân thiện với người dùng nhưng để sử dụng được những dịch vụ đó, đòi hỏi doanh nghiệp phải có những chuyên gia về mảng liên quan để có thể sử dụng được hết năng suất vốn có của nó. Tiếp theo là về vấn đề chi phí và giới hạn dung lượng, như đã trình bày ở trên, AWS phủ sóng gần như ở khắp mọi nơi nhưng không phải tất cả nơi đó đều có giá thành giống như nhau do về bản chất Amazon vẫn sẽ phải xây dựng cơ sở hạ tầng vậy lý ở nơi đó. Sự khác nhau về thời tiết, khí hậu, … dẫn đến chi phí bảo trì là khác nhau, chưa kể đến khoảng cách và số lượng cơ sở hạ tầng của Amazon ở quốc gia đó cũng khác nhau, đó là lý do dẫn đến chi phí và giới hạn dung lượng ở mỗi khu vực không giống nhau.

Để sử dụng được dịch vụ Amazon S3 vô cùng đơn giản. Bạn chỉ cần đăng ký tài khoản trên website của Amazon là đã có thể sử dụng được những dịch vụ trên nhưng cần lưu ý là bạn phải có thẻ thanh toán quốc tế như Visa, MasterCard, …

### Kiến trúc

Phần này hơi phức tạp nên em cần thời gian để nghiên cứu thêm ạ

### Các thành phần công nghệ được sử dụng

#### Amazon Simple Storage Service (Amazon S3)

Amazon S3 là một trong những dịch vụ thuộc AWS. Nó cung cấp khả năng lưu trữ dữ liệu với quy mô, tốc độ, bảo mật thuộc hàng đầu thế giới. Khách hàng hay doanh nghiệp không cần phải tốn một chi phí quá lớn để xây dựng cơ sở hạ tầng còn nhiều rủi ro mà chỉ cần một chi phí nhỏ hơn để sử dụng dịch vụ trên internet được quản lý bởi doanh nghiệp thuộc top đầu ngành công nghệ.

# HỆ THỐNG….

## Giới thiệu

## Kiến trúc của hệ thống

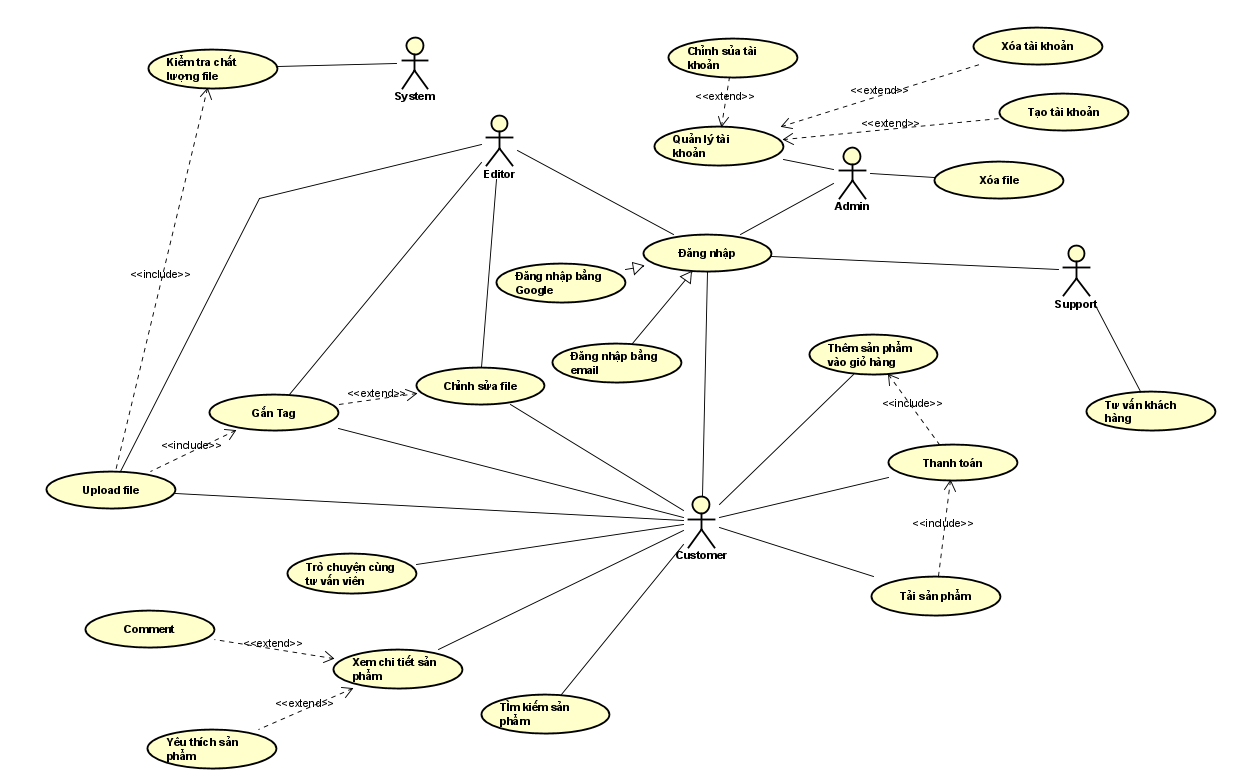
Liên hệ với nhau như thế nào (client server)

Nội dung………………….

## Phân tích và thiết kế hệ thống

### Phân tích use case

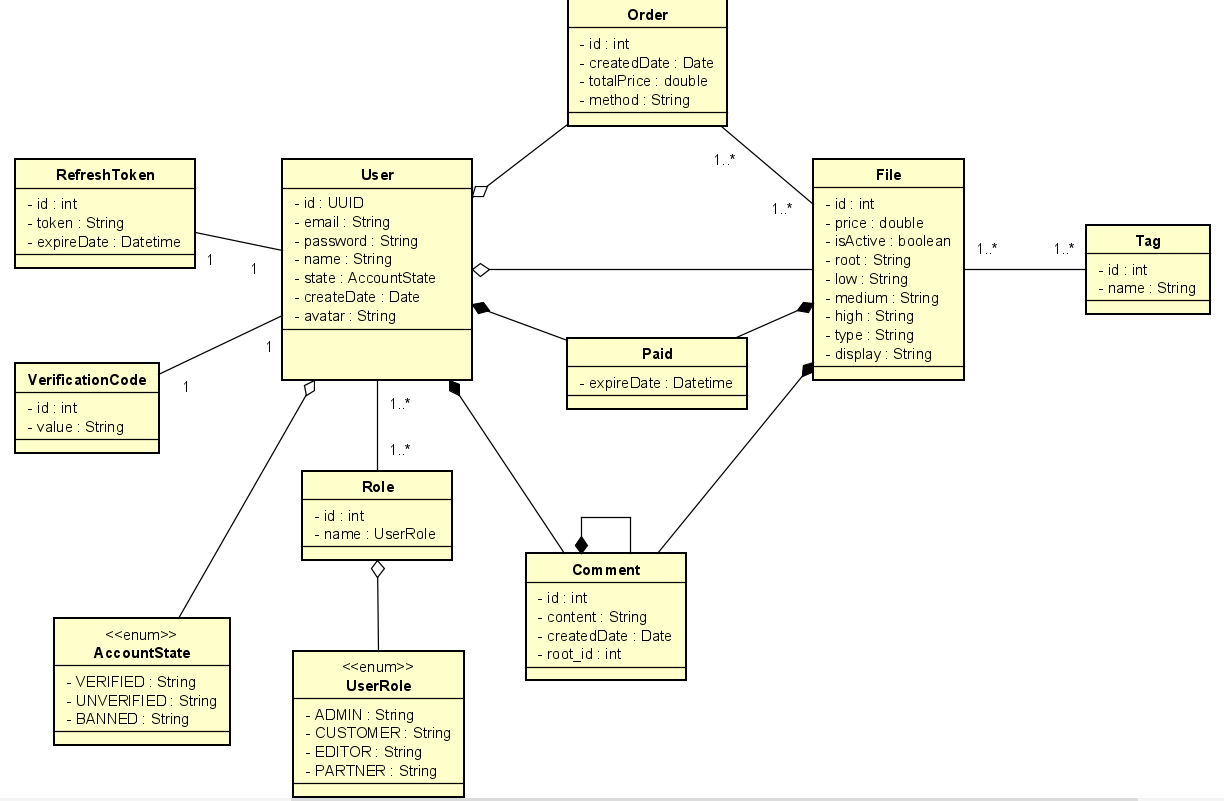
Có thể trình bày cả lược đồ hoạt động ở đây, các hình use case cắt nhỏ nếu quá lớn và đặc tả những chức năng quan trọng



### Thiết kế giao diện [optional]

### Thiết kế CSDL

Chụp hình và trình bày những ràng buộc trong database tại sao lại chọn như vậy, giải quyết vấn đề gì trong thế giới thực và đánh giá với thiết kế này thì đáp ứng được vận tải lớn hay nhỏ



### Các nghiệp vụ của đề tài

Đề tài này đã hoạt thành cách chức năng sau đây và để chạy được đề tài này thì cần có môi trường sau đây cần set up

#### Chức năng 1

Chức năng này làm gì và giải quyết được gì trong thế giới thực

Chụp hình sản phẩm

#### Chức năng 2

# Kết luận và hướng phát triển

Kết luận và hướng phát triển (từ 1 – 2 trang)

## Kết luận

Đề tài này đặt ra vấn đề này giải quyết được chưa, vận dụng được công nghệ đó còn khiếm khuyết gì không

Ưu và khuyết của bản thân (chính bản thân và đề tài)

## Hướng phát triển

Hoàn thành các chức năng cơ bản, hoàn thiện các chức năng và dự định

TÀI LIỆU THAM KHẢO

(Theo chuẩn IEEE – *bỏ dòng này khi viết báo cáo*)

PHỤ LỤC