TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––

🙠----🕮----🙢

Ảnh có chứa văn bản

Mô tả được tạo tự động

**BÁO CÁO THỰC TẬP NGHỀ NGHIỆP**

**ĐỀ TÀI:**

**XÂY DỰNG WEBSITE TÌM KIẾM**

**VÀ THUÊ PHÒNG TRỌ**

**Đơn vị thực tập:** TMA Solutions Bình Định

**Vị trí thực tập:** Java Developer Internship

**Sinh viên thực hiện:** Thái Thị Thanh Thủy

**Lớp :** 44K21.1

**Giảng viên hướng dẫn :** Cao Thị Nhâm

**Đà Nẵng, 7/2021**

**DANH SÁCH THÀNH VIÊN THAM GIA**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Họ và tên** | **Lớp** | **Nhiệm vụ** |
| 1 | Thái Thị Thanh Thủy | 44K21.1 | Thành viên |

**LỜI CẢM ƠN**

Để hoàn thành khóa thực tập nghề nghiệp này, em xin gửi đến quý thầy, cô giáo trong khoa Thống kê – Tin học trường Đại học Kinh Tế - Đại học Đà Nẵng lời cảm ơn chân thành.

Đặc biệt, em xin gởi đến cô Cao Thị Nhâm người đã tận tình hướng dẫn, giúp đỡ em trong quá trình học hỏi kiến thức mới và hỗ trợ em hoàn thành báo cáo trong kỳ thực tập này lời cảm ơn sâu sắc nhất.

Em xin gửi lời cảm ơn đến Ban lãnh đạo, các anh chị Mentors của công ty TMA Bình Định Solution đã tạo điều kiện thuận lợi cho em được thực tập, tìm hiểu nhiều kiến thức mới và tập tình giúp đỡ em rất nhiều trong suốt quá trình thực tập này.

Cuối cùng em kính chúc quý thầy, cô dồi dào sức khỏe và thành công hơn trong sự nghiệp cao quý. Đồng kính chúc các anh chị trong TMA Bình Định sức khỏe, đạt được nhiều thành công tốt đẹp trong cuộc sống.

**LỜI CAM ĐOAN**

Em xin cam đoan kết quả nghiên cứu của đề tài: “Xây dựng Website tìm kiếm và thuê phòng trọ” là trung thực không có bất kỳ sự sao chép từ nguồn nào khác. Ngoài ra, trong bài báo cáo có sử dụng một số nguồn tài liệu tham khảo đã được trích dẫn nguồn và chú thích rõ ràng. Em xin hoàn toàn chịu trách nhiệm, kỷ luật của bộ môn, khoa và nhà trường đề ra nếu như có xảy ra vấn đề.

# MỤC LỤC

[MỤC LỤC ii](#_Toc77374847)

[LỜI MỞ ĐẦU 1](#_Toc77374848)

[CHƯƠNG 1. TỔNG QUAN 2](#_Toc77374849)

[1.1. Tổng quan về ngôn ngữ lập trình Java 2](#_Toc77374850)

[1.1.1. Giới thiệu 2](#_Toc77374851)

[1.1.2. Công cụ sử dụng 2](#_Toc77374852)

[1.2. Tổng quan về MySQL 3](#_Toc77374853)

[1.2.1. Giới thiệu 3](#_Toc77374854)

[1.2.2. Công cụ sử dụng 3](#_Toc77374855)

[1.3. Cơ hội nghề nghiệp 4](#_Toc77374856)

[1.3.1. Mô tả vị trí Java developer 4](#_Toc77374857)

[1.3.2. Các kỹ năng cần có 4](#_Toc77374858)

[1.3.3. Mức lương khởi điểm cho 4](#_Toc77374859)

[CHƯƠNG 2. cƠ SỞ lÝ THUYẾT 5](#_Toc77374860)

[2.1. Java core 5](#_Toc77374861)

[2.2. Java OOP 7](#_Toc77374862)

[2.3. RESTful API 7](#_Toc77374863)

[2.4. Spring Boot 8](#_Toc77374864)

[2.5. Spring Data JPA 8](#_Toc77374865)

[CHƯƠNG 3. TRIỂN KHAI Dự án 9](#_Toc77374866)

[3.1. Phân tích yêu cầu 9](#_Toc77374867)

[3.1.1. Sơ đồ phân cấp chức năng 9](#_Toc77374868)

[3.2. Xây dựng cơ sở dữ liệu 10](#_Toc77374869)

[3.3. Tạo danh sách các API cho dự án 12](#_Toc77374870)

[CHƯƠNG 4. KẾT QUẢ 14](#_Toc77374871)

[4.1. Mục 4.1… 14](#_Toc77374872)

[4.2. Mục 4.2… 14](#_Toc77374873)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 15](#_Toc77374874)

[TÀI LIỆU THAM KHẢO 16](#_Toc77374875)

[PHỤ LỤC (nếu có) 16](#_Toc77374876)

DANH MỤC HÌNH ẢNH

[Hình 1.1 Ngôn ngữ lập trình Java 2](#_Toc77374957)

[Hình 1.2 MySQL 3](#_Toc77374958)

[Hình 1.3 MySQL Workbench 3](#_Toc77374959)

DANH MỤC BẢNG BIỂU

[Bảng 2.1 Kiểu dữ liêu cơ bản 5](#_Toc77374800)

[Bảng 2.2 Các toán tử cơ bản 6](#_Toc77374801)

[Bảng 2.3 Sơ đồ vòng lặp 6](#_Toc77374802)

[Bảng 3.1 Sơ đồ phân cấp chức năng 9](#_Toc77374803)

[Bảng 3.2 Rental 10](#_Toc77374804)

[Bảng 3.3 Users 10](#_Toc77374805)

[Bảng 3.4 Review 10](#_Toc77374806)

[Bảng 3.5 Room\_Share 11](#_Toc77374807)

[Bảng 3.6 Comment 11](#_Toc77374808)

[Bảng 3.7 Province 11](#_Toc77374809)

[Bảng 3.8 Districs 11](#_Toc77374810)

[Bảng 3.9 Ward 11](#_Toc77374811)

[Bảng 3.10 File 12](#_Toc77374812)

[Bảng 3.11 List API 12](#_Toc77374813)

DANH MỤC CÁC TỪ VIẾT TẮT

**SE** : Standard Edittion

**JDK** : **Java Development Kit**

**SQL :** Structured Query Language

**JEE** : Java Enterprise Edition

**ORM** : Object Relational Mapping

**OOP** : Object Oriented Programing

**REST :** **RE**presentational **S**tate Transfer

**API : A**pplication **P**rogramming **I**nterface

**JPA :** Java Persistence API

# LỜI MỞ ĐẦU

1. **Mục tiêu nghiên cứu của đề tài**

* Trước vấn đề tìm kiếm phòng trọ của những sinh viên, công nhân… vì đi học, đi làm ăn xa quê hương phải đi từng con đường lớn đến hẻm nhỏ để tìm được một căn phòng trọ tốt hợp túi tiền. Thì đề tài xây dựng website tìm kiếm và thuê phòng trọ ở Việt Nam sẽ hỗ trợ trợ cho cả người chủ trọ và người đi thuê dễ dàng tìm kiếm được phòng trọ tốt và người thuê phòng.

1. **Nhiệm vụ của đề tài**

* Đầu tiên, giúp cho người đi thuê có nhiều lựa chọn phòng trọ tại khu vực muốn thuê và không mất thời gian tìm kiếm
* Thứ hai, hỗ trợ chủ trọ tìm được người thuê phòng và có thể tương tác với nhau dễ dàng thỏa thuận giá cả
* Cuối cùng, người dùng có thể đánh giá và xem đánh giá về phòng trọ, người thuê trọ và người cho thuê.

1. **Đối tượng và phạm vi đề tài**

* Đối tượng của đề tài là việc tìm kiếm và cho thuê phòng trọ.
* Phạm vi về không gian là lãnh thổ Việt Nam

1. **Kết cấu của đề tài**

Đề tài được tổ chức gồm phần mở đầu, 4 chương nội dung và phần kết luận như sau:

* Mở đầu
* **Chương 1**: Tổng quan
* **Chương 2**: Cơ sở lý thuyết
* **Chương 3**: Triển khai dự án
* **Chương 4**: Kết quả
* Kết luận và hướng phát triển

# TỔNG QUAN

## Tổng quan về ngôn ngữ lập trình Java

### Giới thiệu



Hình . Ngôn ngữ lập trình Java

Ngôn ngữ lập trình Java (Hình 1.1) là ngôn ngữ lập trình thuần hướng đối tượng. Java được phát triển bởi Sun Microsystems (sau này được Oracle ua lại) do James Gosling và đồng nghiệp khởi xướng, ban đầu được gọi là Oak. Phát hành vào dưới dạng Java 1.0 vào năm 1995. Tính đến thời điểm này (2021) Java SE 16 là phiên bản mới nhất được phát hành. Tạo ra với mục đích chung là cho phép các nhà phát triển ứng dụng “Viết một lần, chạy mọi nơi” (Write Once, Run Anywhere – WORA) nghĩa là khi mã Java dã biên dịch có thể chạy trên tất cả các nền tảng hổ trợ Java mà không cần biên dịch lại.

### Công cụ sử dụng

* Hệ điều hành Windows 10
* Cài đặt IntelliJ IDEA Community
* Thiết lập môi trường cài đặt JDK

## Tổng quan về MySQL

### Giới thiệu

Ảnh có chứa văn bản, mẫu họa

Mô tả được tạo tự động

Hình . MySQL

Công ty Thuy Điển MySQL AB phát triển MySQL vào năm 1994. Công ty công nghệ Mỹ Sun Microsystem sau đó giữ quyền sở hữu MySQL sau khi mua lại MySQL vào năm 2008. Năm 2010, gã khổng lồ Oracle mua Sun Microsystems và MySQL thuộc quyền sở hữu của Oracle từ đó.

MySQL là hệ quản trị cơ sở dữ liệu mã nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.

### Công cụ sử dụng

MySQL là một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất hiện nay. Nhưng giao diện dòng lệnh thường không thân thiện với người dùng (đặc biệt là đối với những người mới) và với các cơ sở dữ liệu lớn thì việc quản lý bằng lệnh thường rất khó khăn và kém trực quan hơn. Vì vậy trong dự án này tôi đã **cài đặt công cụ MySQL Workbench.**



Hình . MySQL Workbench

## Cơ hội nghề nghiệp

### Mô tả vị trí Java developer

* Công việc chính của một Java developer là chịu trách nhiệm phân tích nhu cầu của người dùng và mục tiêu kinh doanh, lập trình, xác định các tính năng cũng như chức năng chính của ứng dụng, đề xuất thay đổi đối với các ứng dụng dựa trên Java hiện có, thể hiện toàn bộ chuyên môn của mình trong toàn bộ quá trình phát triển sản phẩm từ lên ý tưởng cho tới kiểm tra.

### Các kỹ năng cần có

* Khả năng thiết kế và phát triển các ứng dụng sử dụng nền tảng Java.
* Biết phân tích hướng đối tượng và sử dụng các mẫu thiết kế phổ biến.
* Hiểu biết sâu về nội bộ Java và JEE.
* Kiến thức về cơ sở dữ liệu, công nghệ SQL và ORM (JPA2, Hibernate).
* Có kinh nghiệm làm việc với Spring Framework

### Mức lương khởi điểm cho

* Theo thống kê của đại học FPT thì mức lương khởi điểm trung bình của lập trình viên mới ra trường chưa có kinh nghiệm tính đến năm 2020 mức lương được trả từ 5 triệu đến 8,3 triệu đồng cho một tháng.

# cƠ SỞ lÝ THUYẾT

## Java core

* Java - Environment Setup: Thiết lập môi trường tương thích cho Java
* Java - Object & Classes
* **Object** - Đối tượng có trạng thái và hành vi. Một đối tượng là một thể hiện của một lớp.
* **Class** - Một lớp có thể được định nghĩa như một khuôn mẫu / bản thiết kế mô tả hành vi / trạng thái mà đối tượng thuộc kiểu của nó hỗ trợ.
* Java - Basic Datatypes

Bảng . Kiểu dữ liêu cơ bản

|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu dữ liệu** | **Giá trị mặc định** | **Bộ nhớ** | **Phạm vi** |
| Boolean | FALSE | 1 bit | True, False |
| Char | '\u0000' | 2 byte | \u0000 (0) đến \uffff (65,535) |
| Byte | 0 | 1 byte | -128 đến 127 |
| Short | 0 | 2 byte | -32,768 đến 32,767 |
| Int | 0 | 4 byte | -2,147,483,648 đến 2,147,483,647 |
| Long | 0L | 8 byte | -9,223,372,036,854,775,808 đến 9,223,372,036,854,775,807 |
| Float | 0.0f | 4 byte | -3.40282347E+38 đến 3.40282347E+38 |
| Double | 0.0d | 8 byte | -1.79769313486231570E + 308 đến 1.79769313486231570E + 308 |

* Java - Variable Types
* Local Variables
* Instance Variables
* Class/Static Variables
* Java - Modifier Types
* Công cụ sửa đổi quyền truy cập Java

Java cung cấp một số công cụ sửa đổi quyền truy cập để thiết lập mức độ truy cập cho các lớp, biến, phương thức và hàm tạo.

* Hiển thị với package, mặc định. Không cần bổ ngữ.
* Chỉ hiển thị trong class (private).
* Hiển thị với bên ngoài (public).
* Hiển thị với gói và tất cả các lớp con (protected).
* Công cụ sửa đổi không truy cập Java

Java cung cấp một số công cụ sửa đổi không truy cập để đạt được nhiều chức năng khác.

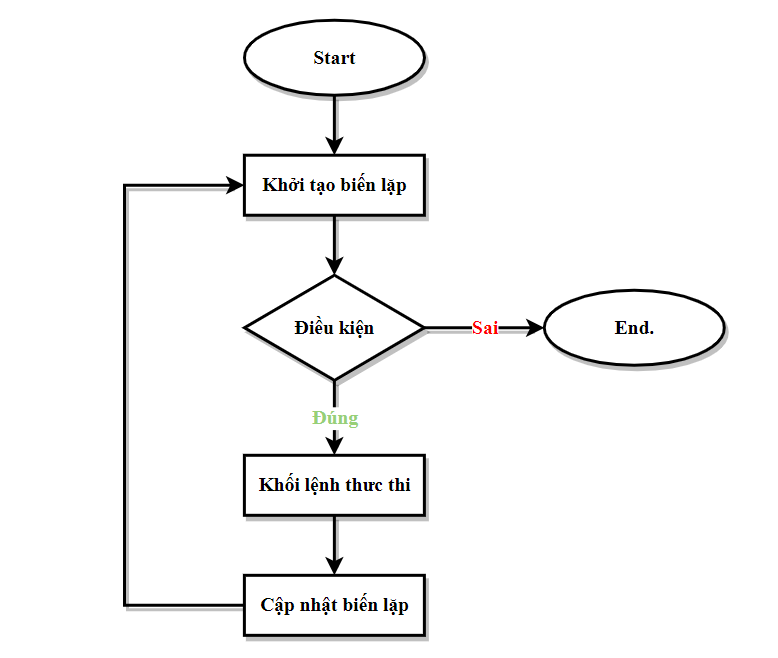
* Công cụ sửa đổi *static* để tạo các phương thức và biến lớp.
* Công cụ sửa đổi *final* để hoàn thiện việc triển khai các lớp, phương thức và biến.
* Công cụ sửa đổi *abstract* để tạo các lớp và phương thức trừu tượng.
* Java - Basic Operators

Bảng . Các toán tử cơ bản

|  |  |
| --- | --- |
| **Các toán tử** | **Sự miêu tả** |
| + (Phép cộng) | Thêm giá trị vào hai bên của toán tử. |
| - (Phép trừ) | Trừ toán hạng bên phải khỏi toán hạng bên trái. |
| \* (Phép nhân) | Nhân các giá trị ở hai bên của toán tử. |
| / (Phép chia) | Chia toán hạng bên trái cho toán hạng bên phải. |
| % (Phần trăm) | Chia toán hạng bên trái cho toán hạng bên phải và trả về phần dư. |
| ++ (Tăng dần) | Tăng giá trị của toán hạng lên 1. |
| - (Giảm dần) | Giảm giá trị của toán hạng đi 1. |

* Java - Loop Control

Bảng . Sơ đồ vòng lặp



* Java – Methods
* Một phương thức về cơ bản là một hành vi. Một lớp có thể chứa nhiều phương thức. Nó nằm trong các phương thức nơi logic được viết, dữ liệu được thao tác và tất cả các hành động được thực thi.
* Java - Files and I/O: Gói java.io chứa gần như mọi lớp mà bạn có thể cần để thực hiện nhập và xuất (I/O) trong Java
* Java – Exceptions: Một ngoại lệ (hoặc sự kiện đặc biệt) là một vấn đề phát sinh trong quá trình thực hiện một chương trình. Khi một Exception xảy ra, luồng chương trình bình thường bị gián đoạn và chương trình / ứng dụng kết thúc bất thường, điều này không được khuyến nghị, do đó, những ngoại lệ này phải được xử lý.

## Java OOP

* Java – Inheritance: Lớp con có thể kế thừa các thuộc tính và phương thức từ lớp cha.
* Java – Overriding: Lớp con có thể triển khai một phương thức của lớp cha dựa trên phương thức thuộc về đối tượng cụ thể của nó.
* Java – Polymorphism: Cho phép thực hiện một hành động theo nhiều cách khác nhau.
* Java – Abstraction: Lớp abstract không thể khởi tạo đối tượng. Phương thức abstract chỉ có thể sử dụng trong lớp abstract và phần thân được thực hiện bởi lớp khác.
* Java – Encapsulation: Cho phép chúng ta liên kết thuộc tính, phương thức của đối tượng lại với nhau.
* Java – Interfaces: Giúp các lập trình viên thực hiện đa kế thừa trong Java
* Java – Packages: Được sử dụng để nhóm các lớp liên quan lại với nhau. Để tránh xung đột tên và có khả năng sử dụng lại.

## RESTful API

* **RESTful API** là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.
* REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.
* GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
* POST (CREATE): Tạo mới một Resource.
* PUT (UPDATE): Cập nhật thông tin cho Resource.
* DELETE (DELETE): Xoá một Resource.

## Spring Boot

* Spring Boot là một dự án phát triển bởi JAV (ngôn ngữ Java) trong hệ sinh thái Spring framework. Nó giúp cho các lập trình viên đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển kinh doanh cho ứng dụng.
* Dưới đây là một số tính năng nổi bật của Spring Boot:
* Tạo các ứng dụng Spring độc lập
* Nhúng trực tiếp Tomcat, Jetty hoặc Undertow (Không cần phải deploy ra file WAR)
* Các starter dependency giúp việc cấu hình Maven đơn giản hơn
* Tự động cấu hình Spring khi cần thiết
* Không sinh code cấu hình và không yêu cầu phải cấu hình bằng XML …

## Spring Data JPA

* Spring Data JPA là một phần của Spring Data giúp dễ dàng triển khai các kho lưu trữ dựa trên JPA.
* Các tính năng cơ bản của Spring Data JPA:
* Hỗ trợ tinh vi để xây dựng kho lưu trữ dựa trên Spring và JPA
* Hỗ trợ các vị từ Querydsl và do đó, các truy vấn JPA an toàn về kiểu chữ
* Kiểm tra minh bạch của lớp miền
* Hỗ trợ phân trang, thực thi truy vấn động, khả năng tích hợp mã truy cập dữ liệu tùy chỉnh
* Xác thực các @Querytruy vấn có chú thích tại thời gian khởi động
* Hỗ trợ ánh xạ thực thể dựa trên XML
* Cấu hình kho lưu trữ dựa trên JavaConfig bằng cách giới thiệu @EnableJpaRepositories.

# Triển khai dự án

## Phân tích yêu cầu

### Sơ đồ phân cấp chức năng

Bảng . Sơ đồ phân cấp chức năng

## Xây dựng cơ sở dữ liệu

Bảng . Rental

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Khóa |
| roomId | VARCHAR(36) | PK |
| userId | VARCHAR(36) |  |
| provinceId | VARCHAR(36) |  |
| districtId | VARCHAR(36) |  |
| wardId | VARCHAR(36) |  |
| address | VARCHAR(255) |  |
| acreage | FLOAT |  |
| roomPrice | INT |  |
| describe | VARCHAR(2048) |  |
| status | ENUM |  |
| date | DATE |  |

Bảng . Users

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Khóa |
| userId | VARCHAR(36) | PK |
| userName | VARCHAR(50) |  |
| password | VARCHAR(255) |  |
| name | VARCHAR(255) |  |
| email | VARCHAR(45) |  |
| address | VARCHAR(255) |  |
| phone | VARCHAR(11) |  |
| birthDate | DATE |  |
| gender | INT |  |
| avatarUrl | VARCHAR(1024) |  |
| role | ENUM |  |

Bảng . Review

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Khóa |
| reviewId | VARCHAR(36) | PK |
| userId | VARCHAR(36) |  |
| content | VARCHAR(1024) |  |
| date | DATE |  |
| evaluate | INT |  |
| parentType | INT |  |
| parentId | VARCHAR(36) |  |

Bảng . Room\_Share

|  |  |  |
| --- | --- | --- |
| ROOM\_SHARING |  |  |
| Tên cột | Kiểu dữ liệu | Khóa |
| roomSharingId | VARCHAR(36) | PK |
| roomId | VARCHAR(36) |  |
|  |  |  |
| DETAIL |  |  |
| Tên cột | Kiểu dữ liệu | Khóa |
| roomSharingsId | VARCHAR(36) | PK |
| roomSharingId | VARCHAR(36) |  |
| userId | VARCHAR(36) |  |
| role | ENUM |  |

Bảng . Comment

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Khóa |
| commentId | VARCHAR(36) | PK |
| userId | VARCHAR(36) |  |
| reviewId | VARCHAR(36) |  |
| content | VARCHAR(1024) |  |
| date | DATE |  |

Bảng . Province

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Khóa |
| provinceId | VARCHAR(36) | PK |
| name | VARCHAR(1024) |  |

Bảng . District

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Khóa |
| districtId | VARCHAR(36) | PK |
| provinceId | VARCHAR(36) |  |
| name | VARCHAR(1024) |  |

Bảng . Ward

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Khóa |
| wardId | VARCHAR(36) | PK |
| districsId | VARCHAR(36) |  |
| name | VARCHAR(1024) |  |

Bảng . File

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Khóa |
| name | VARCHAR(1024) |  |
| type | ENUM |  |
| parentType | INT |  |
| parentId | VARCHAR(36) |  |
| url | VARCHAR(1024) |  |

## Tạo danh sách các API cho dự án

Bảng . List API

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Function** | **Method** | **url** |
| 1 | create account | **POST** | /user |
| 2 | get account | **GET** | /user/{user\_id} |
| 3 | update account | **PUT** | /user/{user\_id} |
| 4 | delete account | **DELETE** | /user/{user\_id} |
| 5 | create room info | **POST** | /rent |
| 6 | get all room info | **GET (all)** | /rent/all |
| 7 | get room info | **GET** | /rent/{room\_id} |
| 8 | update room info | **PUT** | /rent/{room\_id} |
| 9 | delete room info | **DELETE** | /rent/{room\_id} |
| 10 | get province | **GET** | /province/{province\_id} |
| 11 | get district | **GET** | /province/district{district\_id} |
| 12 | get wards | **GET** | /province/district/wards{wards\_id} |
| 13 | create file | **POST** | /file |
| 14 | get file | **GET** | /file/{file\_id} |
| 15 | Search File | **GET** | /file/search |
| 16 | update file | **PUT** | /file/{file\_id} |
| 17 | delete file | **DELETE** | /file/{file\_id} |
| 18 | create room share | **POST** | /sharing |
| 19 | get room share | **GET** | /sharing |
| 20 | delete room share | **DELETE** | /sharing/{room\_sharing\_id} |
| 21 | create room share detail | **POST** | /sharingdetail |
| 22 | Join room share detail | **POST** | /sharingdetail/join/{room\_sharing\_id} |
| 23 | update room share detail | **PUT** | /sharingdetail/{room\_sharings\_id} |
| 24 | delete room share detail | **DELETE** | /sharingdetail/{room\_sharings\_id} |
| 25 | create review | **POST** | /review |
| 26 | get review all | **GET (all)** | /review |
| 27 | get review | **GET** | /review/{review\_id} |
| 28 | update review | **PUT** | /review/{review\_id} |
| 29 | create comment | **POST** | /comment |
| 30 | get comment | **GET** | /comment |
| 31 | get all comment | **GET** | /comment |
| 32 | update comment | **PUT** | /comment/{commentId} |
| 33 | delete comment | **DELETE** | /comment/{commentId} |
| 34 | Search room info (filter) | **GET** | /search/roominfo |
| 35 | Search room info (sort) | **GET** | /search/roominfo |
| 36 | Search owner | **GET** | /search/owner |
| 37 | search rating | **GET** | /search/rating |

## Build project with Spring Boot

# KẾT QUẢ

## Mục 4.1…

## Mục 4.2…

Kết quả được xây dựng dựa trên ngôn ngữ lập trình thể hiện như (Bảng 4.1).

Bảng 4.1 Kiến trúc

|  |  |  |
| --- | --- | --- |
| Lớp | Ý nghĩa | Tham số |
|  |  |  |
|  |  |  |
|  | | |

Hình 4.1 Kiến trúc của mô hình RNN

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Đề tài đã thực hiện được …

# TÀI LIỆU THAM KHẢO

1. Phan, D.-V., et al., *Liver cancer prediction in a viral hepatitis cohort: A deep learning approach.* International Journal of Cancer, 2020. **147**(10): p. 2871-2878.

2. Phan, D.V., et al., *Deep learning approaches for sleep disorder prediction in an asthma cohort.* J Asthma, 2020: p. 1-9.

Tên tác giả, Tên tác giả… *Tên công trình*. Nơi xuất bản, năm, trang

# PHỤ LỤC (nếu có)