Version <1.1>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <30/11/2022> | <1.0> | <details> | All |
| <14/12/2022> | <1.1> | Thêm mục 5 và 6, chỉnh sửa mối quan hệ giữa model và Database | THThien  PGKhiem  NDVNhi |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 5

2. Architectural Goals and Constraints 5

3. Use-Case Model 6

4. Logical View 7

4.1 Component: AuthView 10

4.2 Component: HomeView 11

4.3 Component: SearchingView 11

4.4 Component: RecipeView 11

4.5 Component: UploadRecipeView 11

4.6 Component: CartView 11

4.7 Component: OrderView 12

4.8 Component: BillView 12

4.9 Component: AccountView 12

4.10 Component: NotificationView 13

4.11 Component: ManagementView 13

4.12 Component: AuthController 13

4.13 Component: HomeController 13

4.14 Component: SearchingController 13

4.15 Component: RecipeController 14

4.16 Component: UploadRecipeController 14

4.17 Component: CartController 14

4.18 Component: OrderController 14

4.19 Component: BillController 15

4.20 Component: AccountController 15

4.21 Component: NotificationController 15

4.22 Component: ManagementController 16

4.23 Component: AccountModel 16

4.24 Component: PostModel 17

4.25 Component: SearchingModel 17

4.26 Component: OrderIngredientModel 18

4.27 Component: BillModel 18

4.28 Component: AccountService 18

4.29 Component: PostService 19

4.30 Component: SearchingService 19

4.31 Component: OrderIngredientService 19

4.32 Component: BillService 20

5. Deployment 20

6. Implementation View 21

6.1 Thư mục configs 21

6.2 Thư mục controllers 21

6.3 Thư mục db 21

6.4 Thư mục helpers 21

6.5 Thư mục models 22

6.6 Thư mục node\_modules 22

6.7 Thư mục public 22

6.7.1 Thư mục css 22

6.7.2 Thư mục fonts 22

6.7.3 Thư mục images 22

6.7.4 Thư mục js 22

6.8 Thư mục routers 22

6.9 Thư mục views 22

6.9.1 Thư mục layouts 22

6.9.2 Thư mục partials 22

6.9.3 Thư mục users 22

6.10 Tập tin package-lock.json 22

6.11 Tập tin package.json 22

6.12 Tập tin server.js 22

# Introduction

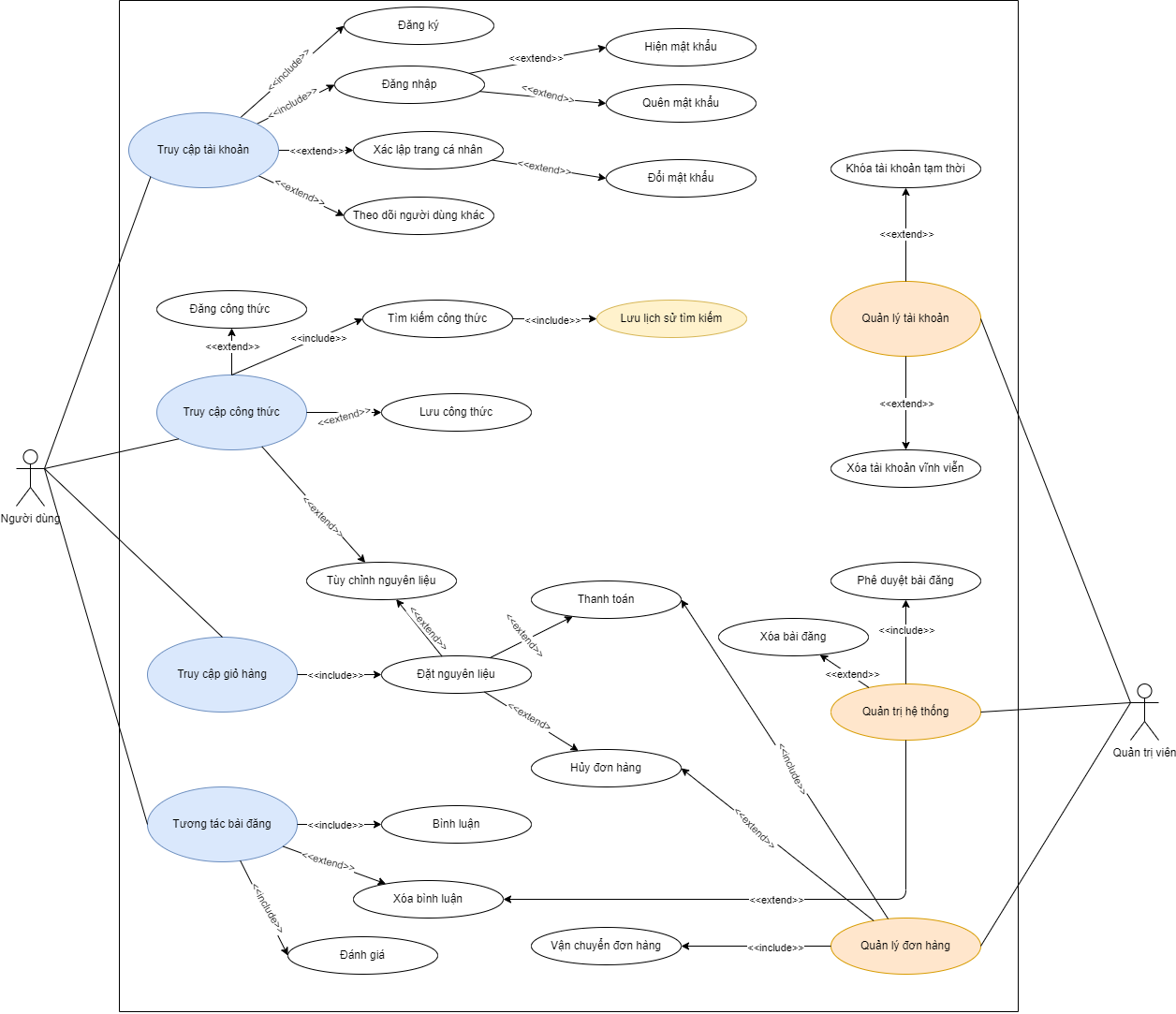
Hệ thống của website ĂN NGON được xây dựng trên Model-View-Controller (MVC), đây là mô hình thiết kế được sự dụng trong kỹ thuật phần mềm. MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia thành ba phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác. Tên gọi 3 thành phần:

* **Model (dữ liệu):** Quản lí, xử lí các dữ liệu.
* **View (giao diện):** Nơi hiển thị dữ liệu cho người dùng.
* **Controller (bộ điều khiển):** Điều khiển sự tương tác của 2 thành phần Model và View.

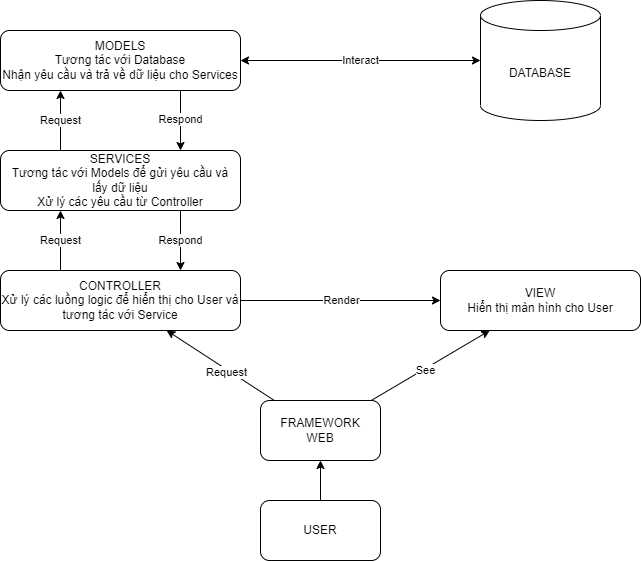
# Architectural Goals and Constraints

* ***Hiệu suất và khả năng mở rộng***: tốc độ tải trang tối đa 3s và đáp ứng nhiều khách hàng truy cập vào website cùng một lúc mà không bị sập server.
* ***Tính di động và khả năng tương thích***:
  + Website hiển thị tốt trên hầu hết thiết bị (PC, laptop, máy tính bảng, điện thoại, …). Bố cục của website được thiết kế phù hợp với nhiều kích cỡ màn hình khác nhau.
  + Website chạy tốt trên hầu hết các trình duyệt như: Google Chrome, Firefox, Microsoft Edge, Safari...
* ***Độ tin cậy, khả năng bảo trì, tính khả dụng***:
* Độ tin cậy: Website hoạt động không bị lỗi trong 90% các trường hợp sử dụng.
* Bảo trì: Thời gian trung bình để khôi phục website sau khi bị lỗi không quá 30 phút.
* Tính khả dụng: Hệ thống đảm bảo vận hành 24/7, nâng cấp tối đa 1 lần trong vòng 3 tháng (khoảng thời gian từ 0h30-1h dùng để bảo trì nâng cấp), downtime mỗi năm không quá 1 tiếng.
* ***Tính bảo mật***:
  + - Thông tin của khách hàng được bảo mật.
    - Cổng xử lý thanh toán tuân thủ PCI DSS.
* ***Tính an toàn***: hệ thống không được làm hại đến môi trường hoặc sức khỏe của con người trong mục đích sử dụng.
* ***Tính bản địa***: nội dung website tuân thủ theo đúng những quy định của luật pháp; phù hợp với bản sắc văn hóa dân tộc, tiền tệ và các khía cạnh khác của quốc gia.
* ***Khả năng sử dụng***:
  + - Giao diện rõ ràng, thân thiện, dễ sử dụng. Người dùng có thể tương tác với website và ghi nhớ ngay khi nhìn thấy bố cục trang web.
    - Website mang đến hiệu quả: người dùng dễ dàng đạt được mục đích (tìm công thức nấu ăn, chia sẻ công thức hoặc đặt mua nguyên liệu) nhanh chóng.
    - Tỷ lệ lỗi của người dùng khi tìm kiếm hoặc thanh toán không vượt quá 10%.
    - Khách hàng hài lòng với thiết kế và chức năng của website sau khi sử dụng.

# Use-Case Model



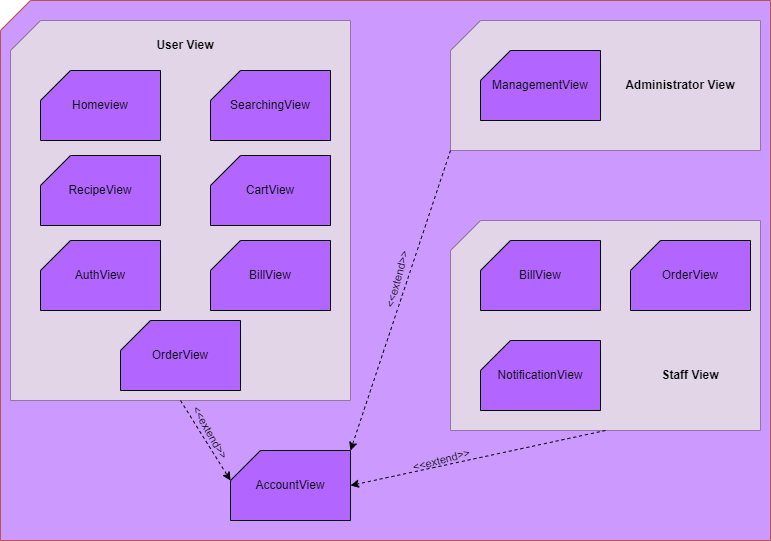
# Logical View



Hình .. Tổng quan về Logical View của ĂN NGON

* **User:** Người sử dụng, bao gồm người dùng và quản trị viên.
* **Framework Web:** Web browser cho user truy cập.
* **View:** Hiển thì màn hình cho User.
* **Controller:** Xử lý các luồng logic để hiển thị cho User và tương tác với Services.
* **Services:** Tương tác với models để gửi yêu vầu và lấy dữ liệu, đồng thời xử lý các yêu cầu từ Controller.
* **Models:** Tương tác với Database, nhận yêu cầu và trả về dữ liệu cho Services.
* **Database:** Dùng PostgreSQL để lưu trữ dữ liệu của chương trình.

**VIEWS:**

****

Hình .. Package Diagram của Views

* **Views:** là thành phần đảm nhận việc thể hiện những dữ liệu của Model. View bao gồm những gì thể hiện trên màn hình như các control, form…

**CONTROLLER:**

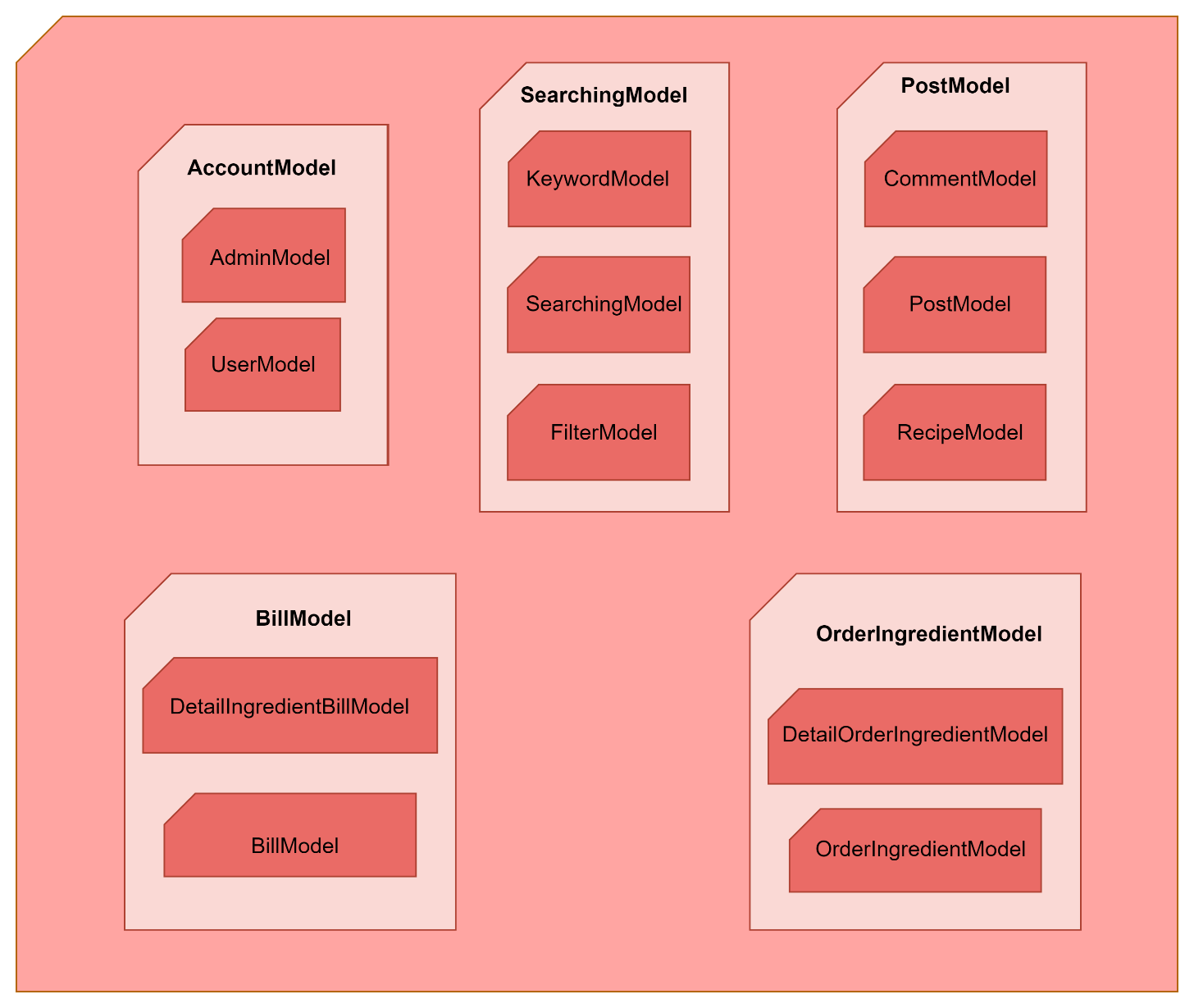
**Diagram, shape

Description automatically generated**

Hình .. Package Diagram của Controller

* **Controller:** là thành phần đảm nhận việc xử lý hồi đáp lại các dữ liệu được đưa vào từ người dùng như các sự kiện chuột, bàn phím, các tương tác lên các control… Controller là cầu nối giữa người dùng và ứng dụng.

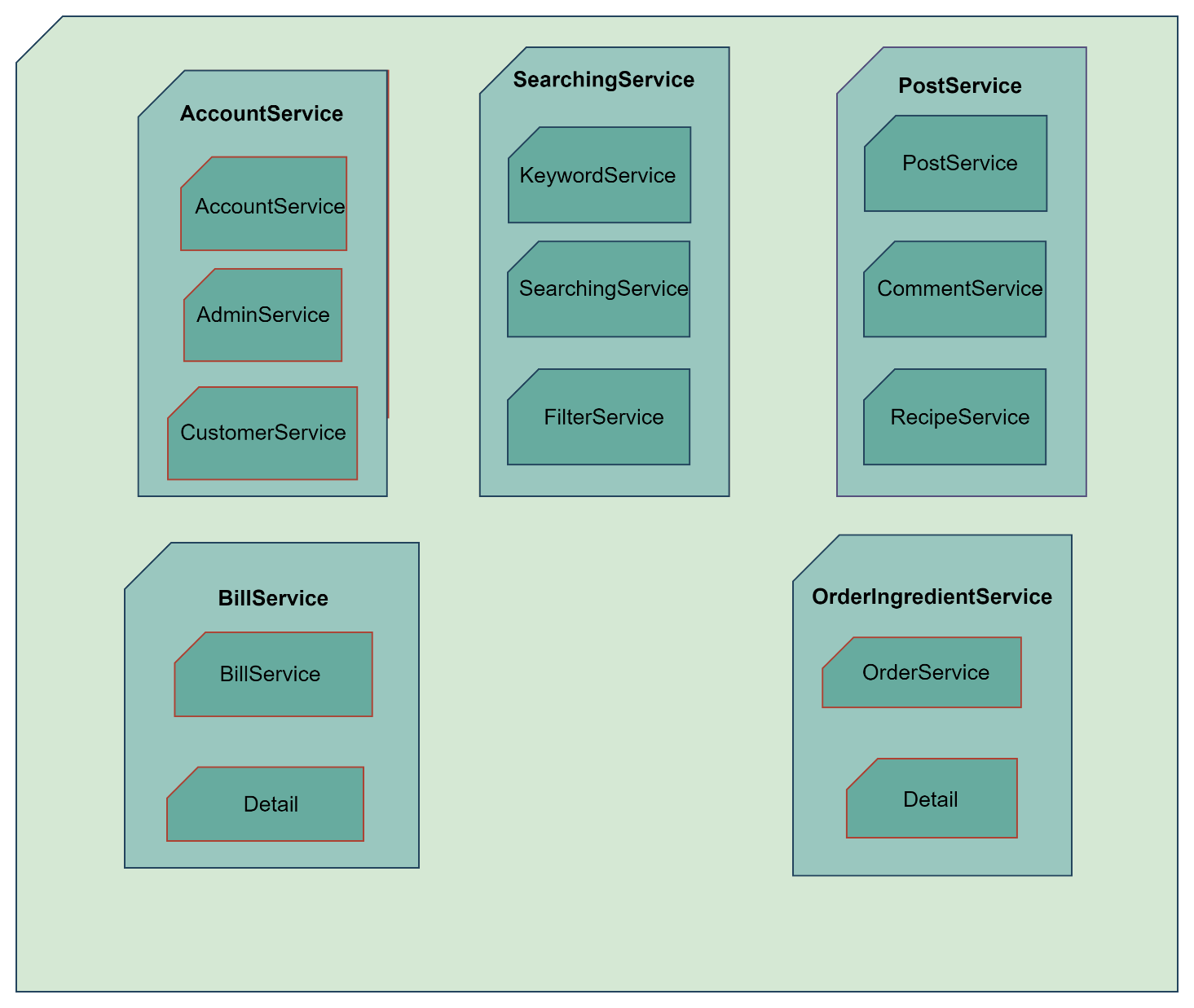
**MODEL:**



Hình 4.4. Package Diagram của Model

* **Model:** chứa dữ liệu và các tính toán xử lý logic để giải quyết vấn đề mà phần mềm hướng tới (business logic).

**SERVICE:**



Hình 4.5. Package Diagram của Service

* **Service:** Là trung gian giữa Controller và Model, thực hiện lấy dữ liệu từ Model và xử lý các business Logic, sau đó trả kết quả về cho Controller.

## Component: AuthView

Mô tả: AuthView là giao diện dùng để hiển thị màn hình đăng nhập/đăng ký của người dùng.

Class diagram:

Diagram

Description automatically generated

Giải thích: Hàm render() trả về giao diện cho người dùng.

## Component: HomeView

Mô tả: HomeView là giao diện dùng để hiện thị trang chủ của website ĂN NGON.

Class diagram:

Table

Description automatically generated with low confidence

Giải thích: Hàm render() trả về giao diện cho người dùng.

## Component: SearchingView

Mô tả: SearchingView là giao diện dùng để hiển thị trang tìm kiếm các công thức món ăn của website ĂN NGON.

Class diagram:

A picture containing table

Description automatically generated

Giải thích: Hàm render() trả về giao diện cho người dùng.

## Component: RecipeView

Mô tả: RecipeView là giao diện dùng để hiển thị chi tiết nội dung công thức của món ăn.

Class diagram:

A picture containing table

Description automatically generated

Giải thích: Hàm render() trả về giao diện cho người dùng.

## Component: UploadRecipeView

Mô tả: UploadRecipeView là giao diện dùng để hiển thị nội dung công thức mà người dùng muốn chia sẽ lên website ĂN NGON.

Class diagram:

A picture containing table

Description automatically generated

Giải thích: Hàm render() trả về giao diện cho người dùng.

## Component: CartView

Mô tả: CartView là giao diện dùng để hiển thị nội dung của giỏ hàng, nơi người dùng lựa chọn và mua các nguyên liệu để làm món ăn.

Class diagram:

Diagram

Description automatically generated

Giải thích:

Hàm render() trả về giao diện cho người dùng.

## Component: OrderView

Mô tả: OrderView là giao diện dùng để hiển thị nội dung, thông tin đặt hàng của người dùng.

Class diagram:

Table

Description automatically generated with medium confidence

Giải thích: Hàm render() trả về giao diện cho người dùng.

## Component: BillView

Mô tả: BillView là giao diện dùng để hiển thị thông tin của đơn hàng mà người dùng đã đặt.

Class diagram:

Diagram

Description automatically generated

Giải thích:

* Hàm render() trả về giao diện cho người dùng.
* BillList: Class danh sách các hóa đơn của người dùng, lớp hiển thị đầu tiên khi truy cập vào BillView.
* BillDetails: Class hiển thị thông tin chi tiết của đơn hàng. Gồm 2 class con là Show và Edit để hiển thị giao diện đơn hàng và chỉnh sửa đơn hàng.

## Component: AccountView

Mô tả: AccountView là giao diện dùng để hiển thị thông tin tài khoản cá nhân của người dùng. Xuất hiện khi người dùng chọn vào user icon nằm ở góc trên bên phải của website. Cho phép người dùng xem và chỉnh sửa trang cá nhân của mình, đổi mật khẩu và xem thông tin, tình trạng các đơn hàng.

Class diagram:

Diagram

Description automatically generated

Giải thích:

* Hàm render() trả về giao diện cho người dùng.
* Profile: Class hiển thị thông tin cá nhân của người dùng.
* ChangePassword: Class thay đổi mật khẩu của người dùng hiện tại.
* MyBill: Class hiển thị thông tin, tình trạng các đơn hàng.

## Component: NotificationView

Mô tả: NotificationView là giao diện dùng để hiển thị các thông báo từ hệ thống cho người dùng.

Class diagram:



Giải thích: Hàm render() trả về giao diện cho người dùng.

## Component: ManagementView

Mô tả: ManagementView là giao diện của administrator (admin) dùng để hiển thị các chức năng quản lý website ĂN NGON. Giao diện xuất hiện khi admin đăng nhập vào trang admin, cho phép admin quản lý nhân viên, tài khoản của người dùng, quản lý các bài đăng, các bình luận của người dùng.

Class diagram:

Giải thích:

* Hàm render(): trả về giao diện cho người dùng.
* StaffManagement: Class quản lý các nhân viên.
* UserManagment: Class quản lý tài khoản của người dùng.

## Component: AuthController

Mô tả: AuthController xử lý các yêu cầu về đăng nhập, đăng ký.

Class diagram:

Table

Description automatically generated

Giải thích:

* Hàm signIn(req, res, next): Thực hiện yêu cầu đăng nhập.
* Hàm signUp(req, res, next): Thực hiện yêu cầu đăng ký.

## Component: HomeController

Mô tả: HomeController xử lý việc điều hướng đến các trang khác nhau.

Class diagram:

Text

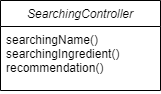
Description automatically generated with medium confidence

Giải thích: Hàm router(req, res, next) thực hiện nhiệm vụ điều hướng.

## Component: SearchingController

Mô tả: SearchingController xử lý việc tìm kiếm công thức.

Class diagram:



Giải thích:

* searchingName(): Hàm tìm kiếm theo tên món ăn.
* searchingIngredient(): Hàm tìm kiếm theo nguyên liệu.
* recommendation(): Hàm xử lý tìm kiếm khi người dùng có thể lựa chọn tìm kiếm từ các đề xuất của hệ thống.

## Component: RecipeController

Mô tả: RecipeController xử lý các thao tác của người dùng trong trang web món ăn (bài đăng).

Class diagram:

Table

Description automatically generated

Giải thích:

* comment(id, vote, cmt): Hàm thực hiện thao tác của các chức năng bình luận, đánh giá giữa người dùng với bài đăng.
* favourite(tenmon): Hàm lưu trữ các bài đăng vào mục yêu thích theo tên món.
* order(list): Chứa danh sách các nguyên liệu mà người dùng muốn mua.

## Component: UploadRecipeController

Mô tả: UploadRecipeController xử lý các yêu cầu của người dùng để thực hiện việc đăng bài, đăng một công thức mới lên website ĂN NGON.

Class diagram:

A picture containing text

Description automatically generated

Giới thiệu: upload(): Hàm trả về giá trị True khi người dùng đăng bài thành công, False khi người dùng đăng bài không thành công.

## Component: CartController

Mô tả: CartController xử lý các yêu cầu của khách hàng trong mục giỏ hàng.

Class diagram:

A picture containing text

Description automatically generated

Giới thiệu: choose() xử lý các lựa chọn của khác hàng, hủy đặt nguyên liệu, hoặc là tùy chỉnh số lượng nguyên liệu muốn mua.

## Component: OrderController

Mô tả: OrderController xử lý các yêu cầu về vấn đề đặt hàng của người dùng trong mục giỏ hàng.

Class diagram:

A picture containing text

Description automatically generated

Giới thiệu: Hàm order() xử lý các yêu cầu của người dùng về việc đặt hay hủy đơn hàng.

## Component: BillController

Mô tả: BillController xử lý các yêu cầu về vấn đề truy vấn thông tin đơn hàng của người dùng (thông tin đơn hàng, hủy đơn hàng, đơn hàng đã giao hay chưa, …)

Class diagram:

Table

Description automatically generated

Giới thiệu:

* + list(req, res, next): Danh sách các đơn hàng.
  + details(req, res, next): Hàm tra cứu thông tin đơn hàng.
  + edit(req, res, next): Hàm giúp người dùng thao tác trên đơn hàng (tùy chỉnh, hủy, hồi đơn hàng, …)

## Component: AccountController

Mô tả: AccountController xử lý các yêu cầu liên quan đến tài khoản của người dùng như thông tin cá nhân, thay đổi mật khẩu, mục yêu thích và hóa đơn.

Class diagram:

Table

Description automatically generated

Giới thiệu:

* profile(req, res, next): Xem thông tin cá nhân.
* changePassword(password): Thay đổi mật khẩu người dùng.
* favourite(req, res, next): Xem các danh mục món ăn yêu thích của người dùng.
* myBill(req, res, next): Xem thông tin các đơn hàng của người dùng.

## Component: NotificationController

Mô tả: NotificationController xử lý các thông báo từ hệ thống.

Class diagram:

Table

Description automatically generated with medium confidence

Giải thích:

* dismiss(): Bỏ qua thông báo.
* access(req, res, next): Truy cập đến nội dung của thông báo.

## Component: ManagementController

Mô tả: ManagementController xử lý các yêu cầu của admin đối với hệ thống, người dùng.

Class diagram:

Table

Description automatically generated

Giải thích:  
Staff: Class giúp admin quản lý nhân viên.

* + information(req, res, next): Quản lý thông tin nhân viên (tạo, xóa, cập nhật, thay đổi,…)
  + actions(req, res, next): Quản lý các hành động, công việc mà nhân viên đã làm trong quá trình hoạt động tại website ĂN NGON.

## Component: AccountModel

Mô tả: Class gồm các thuộc tính là thông tin tài khoản.

Class diagram:

**Shape

Description automatically generated with medium confidence**

## Component: PostModel

Mô tả: Class gồm các thuộc tính là thông tin của công thức món ăn.

Class diagram:

Shape

Description automatically generated with medium confidence

Giải thích:

* vungmien: món ăn đặc trưng ở miền nào (miền Bắc, Trung, Nam)
* cachnau: món ăn là chiên, kho, luộc, hấp, …
* loaimon: món mặn, món ngọt, món chay, món ăn vặt…

## Component: SearchingModel

Mô tả: Class gồm các thuộc tính là thông tin rút gọn của công thức và phương thức tìm kiếm.

Class diagram:

Shape

Description automatically generated with medium confidence

Giải thích: Người dùng có 2 cách tìm kiếm công thức dựa vào type

* type = 1: tìm kiếm bằng cách nhập từ khóa vào thanh tìm kiếm.
* type = 2: tìm kiếm dựa vào bộ lọc (lọc theo vùng miền, loại món, cách nấu, nguyên liệu).

## Component: OrderIngredientModel

Mô tả: Class gồm các thuộc tính là phương thức đặt nguyên liệu.

Class diagram:

Shape

Description automatically generated with medium confidence

## Component: BillModel

Mô tả: Class gồm các thuộc tính là thông tin đơn hàng.

Class diagram:

Shape

Description automatically generated with medium confidence

## Component: AccountService

Mô tả: Thực hiện các chức năng lấy, chỉnh sửa và cập nhật thông tin các tài khoản admin, nhân viên, người dùng.

Class diagram:

Diagram

Description automatically generated

## Component: PostService

Mô tả: Thực hiện chức năng đăng bài, bình luận, đánh giá bài đăng, đưa vào mục yêu thích của các tài khoản admin, nhân viên, người dùng.

Class diagram:

Diagram

Description automatically generated

## Component: SearchingService

Mô tả: Thực hiện chức năng tìm kiếm công thức dựa trên keyword hoặc dựa trên các filter.

Class diagram:

Diagram

Description automatically generated

## Component: OrderIngredientService

Mô tả: Thực hiện chức năng chọn nguyên liệu cần mua: chỉnh sửa thành phần nguyên liệu, thông tin người đặt và nhu cầu đặt hàng của người dùng (được nhân viên quản lý).

Class diagram:

Diagram

Description automatically generated

## Component: BillService

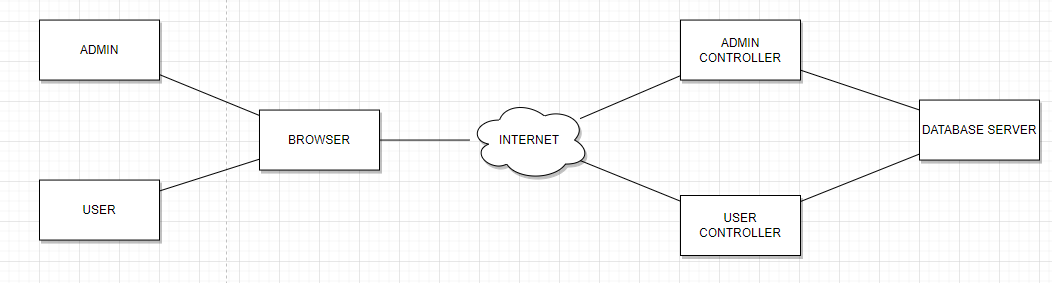
Mô tả: Thực hiện chức năng liên quan đến hóa đơn: tạo, chỉnh sửa, xóa, lấy thông tin của hóa đơn.

Class diagram:

Timeline

Description automatically generated

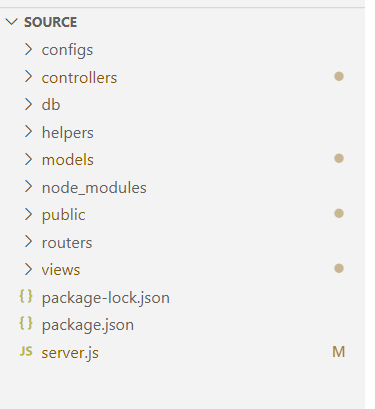
# Deployment



* Admin, User sẽ truy cập đến web thông qua browser. Tương tác trên các layout của trang web để thực hiện các yêu cầu mong muốn và nhận phản hồi từ hệ thống.
* Browser: Môi trường trung gian để truy cập đến web và thực hiện việc trao đổi yêu cầu, phản hồi giữa Admin, User và Server thông qua internet.
* Admin controller: Nhận yêu cầu từ admin thông qua browser, xác định đúng route để gọi các service tương ứng lấy dữ liệu từ Database service, xử lý dữ liệu và phản hồi lại yêu cầu của admin.
* User controller: Nhận yêu cầu từ user thông qua browser, xác định đúng route để gọi các service tương ứng lấy dữ liệu từ Database service, xử lý dữ liệu và phản hồi lại yêu cầu của user.
* Database service: Lưu trữ toàn bộ dữ liệu của hệ thống.

# Implementation View

Cấu trúc thư mục chính của website “Ăn ngon”:



## Thư mục configs

Chứa các file:

* hbs.js: cấu hình template engine express-handlebars, giúp giữ cho trang html đơn giản, sạch đẹp, hỗ trợ thực hiện binding data vào một trang web để hiển thị ra website.
* session.js: cấu hình template engine express-session, giúp quản lý các session trong Express.js.

## Thư mục controllers

Chứa các file hỗ trợ điều khiển tương tác giữa models và views:

* recipes.c.js: nhận yêu cầu từ router, thực hiện xử lý các tác vụ tìm kiếm, xem công thức, đăng tải công thức, đánh giá, … sau đó trả kết quả về trình duyệt.
* Users.c.js: nhận yêu cầu từ router, thực hiện xử lý các tác vụ đăng ký, đăng nhập, lưu trữ đăng nhập, quên mật khẩu, … sau đó trả kết quả về trình duyệt.

## Thư mục db

Chứa các file lưu trữ dữ liệu của trang web dưới dạng json:

* Recipe.json: lưu trữ công thức món ăn.

## Thư mục helpers

Chứa các file helpers.js tự định nghĩa các hàm dùng trong handlebars.

## Thư mục models

Gồm các file định nghĩa, kết nối database và xử lý các yêu cầu về database:

* recipes.m.js: định nghĩa và truy cập vào recipe.json, xử lý yêu cầu lấy dữ liệu món ăn và trả về kết quả cho trình duyệt.
* Users.m.js: định nghĩa và truy cập vào hệ quản trị CSDL PostgreSQL quản lý thông tin của người dùng, xử lý yêu cầu đăng ký, đăng nhập và trả về kết quả cho trình duyệt.

## Thư mục node\_modules

Chứa các modules của Nodejs.

## Thư mục public

Gồm các thư mục con: css, fonts, images, js.

### Thư mục css

Chứa các file css giúp định dạng cho website.

### Thư mục fonts

Chứa các fonts chữ và icon dùng cho website.

### Thư mục images

Chứa hình ảnh sử dụng cho website.

### Thư mục js

Chứa các file javascript giúp cài đặt hiệu ứng cho website, kiểm tra thông tin đăng ký/đăng nhập.

## Thư mục routers

Chứa các file giúp điều hướng các yêu cầu của người dùng đến controller:

* Router.r.js: nhận yêu cầu từ người dùng và điều hướng yêu cầu đến controller tương ứng.

## Thư mục views

Gồm:

* + Các thư mục: layouts, partials, users.
  + Các file handlebars: detail\_recipe.hbs, home.hbs, search.hbs.

### Thư mục layouts

Chứa các layouts tương ứng với từng trang của website.

* Container.hbs: layout cho trang home của website.
* Option\_layouts.hbs: layout cho trang đăng ký/đăng nhập của website.

### Thư mục partials

Chứa các file handlebars, mỗi file chứa một thành phần của website: header, footer, content, carousel, …

### Thư mục users

Chứa các file handlebars: signin.hbs, signup.hbs tương ứng với trang đăng nhập, đăng ký của website.

## Tập tin package-lock.json, package.json

Chứa các modules cần dùng cho website.

## Tập tin server.js

Tập tin tạo server cho website, lắng nghe các kết nối từ clients và các yêu cầu từ client.