**SOFTWARE ARCHITECTURE DOCUMENT**

<Version 1>

FOURSEASONS SHOP MANAGEMENT SYSTEM

**Table of Contents**

[**I. INTRODUCTION**](#_7ck3797hp8ax) **4**

[I.1. Purpose](#_2y2h2qtq12oa) 4

[I.2. Scope](#_sw3d2uwxqx8z) 4

[I.3. Overview](#_p6o15wz4r5bc) 4

[**II. ARCHITECTURAL REPRESENTATION**](#_9zcvmk4vsi1g) **5**

[II.1. Scenarios](#_c2mwk4at9z65) 5

[II.2. Logical view](#_lpffiskmhx4l) 5

[II.3. Data view](#_c2jxh2hza1pd) 5

[II.4. Deployment view](#_6vajw59qemmk) 6

[II.5. Physical view](#_tf8o5g158i36) 6

[**III. ARCHITECTURAL GOALS AND CONSTRAINTS**](#_8r7dmmaq0bwo) **6**

[III.1. Server side](#_vm232pl0qs5v) 6

[III.2. Client side](#_qpc7qqgwmbbg) 6

[III.3. Security](#_hjlqpg9v2r4k) 6

[III.4. Persistence](#_zg3nudltlr5z) 7

[III.5. Reliability/Availability](#_qhbbzj5jljuc) 7

[III.6. Performance](#_68cw27bkofxi) 7

[III.7. Portability and reuse](#_am6afb44zylb) 7

[III.8. Development tools](#_73ghchdtggxl) 7

[III.9. Schedule](#_gke1skvuyjf0) 7

[**IV. USE-CASE VIEW**](#_bwz16qc44rwl) **7**

[IV.1. Actors](#_jif0x794xksz) 8

[IV.2. Use case diagrams](#_829algqsxps7) 8

[IV.2.1. Customer](#_1pu7izu7hl4c) 8

[IV.2.2. Administrator](#_i7vw02nxr9qk) 11

[IV.3. Use-Case Realizations](#_cpv2n9f87bmm) 13

[IV.3.1. Customer](#_f97hrvr8sz32) 13

[IV.3.2. Administrator](#_9eaef2f0fqbt) 17

[**V. LOGICAL VIEW**](#_hw7l8xcqg4gv) **21**

[V.1. Overview](#_iyh3naz8cf8s) 21

[V.1.1. Subsystems](#_ozijgxsezzsj) 21

[V.1.2. Layering](#_qsfcp61cncp3) 22

[V.2. Architecturally significant Design Packages](#_o2t0sdubjrm4) 22

[V.2.1 Class diagram](#_5z0a3zwhz187) 22

[V.2.2 ER diagram](#_tfk3hinh0ye7) 36

[**VI. PROCESS VIEW**](#_z11c32xzczlk) **37**

[VI.1. System sequence diagram](#_gg5nz7b82x0) 37

[VI.2. Activity Diagrams](#_shrb4xk915kw) 44

[**VII. DEPLOYMENT VIEW**](#_p2vhgtpu09cj) **51**

[**VIII. IMPLEMENTATION VIEW**](#_qzs84cuq2jk5) **51**

# **I. INTRODUCTION**

*This document provides a high level overview and explains the architecture of the Distributed Development Monitoring and Mining system.*

*The document defines goals of the architecture, the use cases supported by the system, architectural styles and components that have been selected. The document provides a rationale for the architecture and design decisions made from the conceptual idea to its implementation.*

## **I.1. Purpose**

The Software Architecture Document (SAD) provides a comprehensive architectural overview of the FourSeasons Shop (Clothing Store Management System). It presents a number of different architectural views to depict the different aspects of the system.

## **I.2. Scope**

The scope of this Software Architecture Document is to explain the architecture of the FourSeasons Shop (Clothing Store Management System).

This document describes the various aspects of the FourSeasons Shop system design that are considered to be architecturally significant. These elements and behaviors are fundamental for guiding the construction of the FourSeasons Shop system and for understanding this project as a whole. Stakeholders including designers who require a technical understanding of the FourSeasons Shop system are encouraged to start by reading the Project Plan, Software Requirement Assurance Plan (SQA Plan) and Software Requirement Description documents developed for this system.

## **I.3. Overview**

In order to fully document all the aspects of the architecture, the Software Architecture Document contains the following subsections.

Section 1: describes the use of each view

Section 2: describes the architectural goals and constraints of the system

Section 3: describes the most important use-case realizations

Section 4: describes logical view of the system including interface and operation definitions.

Section 5: describes significant persistence elements.

Section 6: describes how the system will be deployed.

# **II. ARCHITECTURAL REPRESENTATION**

*This document aims to specify the views of the System: Logical view, Process view, Deployment view, Physical view, and Scenarios.*

## **II.1. Scenarios**

**Audience**: All the stakeholders of the system, including the end-users.

**Area**: describes the set of scenarios and use cases that represent some significant, central functionality of the system. Describes the actors for the system, this view presents the needs of the user and is elaborated further at the design level to describe discrete flows and constraints in more detail.

**Related Artifacts** : Use-Case Diagram, Use-Case Specification.

## **II.2. Logical view**

**Audience**: Designers.

**Area**: Requirements: describes the design's object model. Also describes the most important use-case realizations and business requirements of the system.

**Related Artifacts**: Software Requirement Description.

## **II.3. Data view**

**Audience**: Designers.

**Area**: Persistence: describes the architecturally significant persistent elements in the data model as well as how data flows through the system.

**Related Artifacts**: ER Diagram.

## **II.4. Deployment view**

**Audience**: Deployment managers.

**Area**: Topology: describes the mapping of the software onto the hardware and shows the system's distributed aspects. Describes potential deployment structures, by including known and anticipated deployment scenarios in the architecture. This view is also known as the development view.

**Related Artifacts**: Class Diagram.

## **II.5. Physical view**

**Audience**: Designers

**Area**: Persistence: refers to the way data are physically stored and processed in a database.

**Related Artifacts**: Database Design.

# **III. ARCHITECTURAL GOALS AND CONSTRAINTS**

## **III.1. Server side**

FourSeasons Shop Management System uses XAMPP for building LocalHost. phpMyAdmin will be used as the central database server. All communication with clients has to comply with public HTTPs communication protocol standards.

## **III.2. Client side**

Users will be able to access FourSeasons Shop Management System only online. PHP, CSS are major languages for the client side. Users are expected to use a modern web browser such as Mozilla Firefox 10, Google Chrome,... to get full user experience.

## **III.3. Security**

Central security features are handled by the institute officials. They’ll be given full access features both in the website and database levels. Administrator will be provided with a default password which can be changed by administrator. Users can use their accounts with passwords that they created when registering.

## **III.4. Persistence**

All the data will be stored in the central database.

## **III.5. Reliability/Availability**

The system will be subjected to several testing operations (Unit testing, system testing,...) before being deployed in order to make sure that the system is reliable. The database server can respond to many numbers of clients at a given moment without losing consistency and data integrity.

## **III.6. Performance**

The system responds to any requests under standard database and web server script timeouts. System performance can also depend on network and internet connection.

## **III.7. Portability and reuse**

## **III.8. Development tools**

The project uses many development tools:

* Programming: PHP (Visual Studio Code)
* Database: phpMyAdmin
* Diagrams: Visual Paradigm
* Schedule: TeamGantt

## **III.9. Schedule**

The project follows the schedule on TeamGantt. All deliverables and relevant deadlines are considered and satisfied in this work.

# **IV. USE-CASE VIEW**

*The purpose of the use-case view is to give additional context surrounding the usage of the system and the interactions between its components. For the purposes of this document, each component is considered a use-case actor. Section 4.1 lists the current actors and gives a brief description of each in the overall use context of the system. In section 4.2 and 4.3, the most common use-cases are outlined and illustrated using UML use-case diagrams and use-case realizations to clarify the interactions between components.*

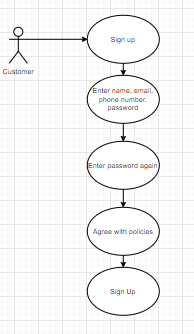
## **IV.1. Actors**

* Administrator
* Customer

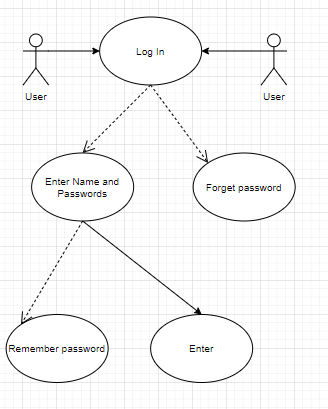
## **IV.2. Use case diagrams**

### ***IV.2.1. Customer***

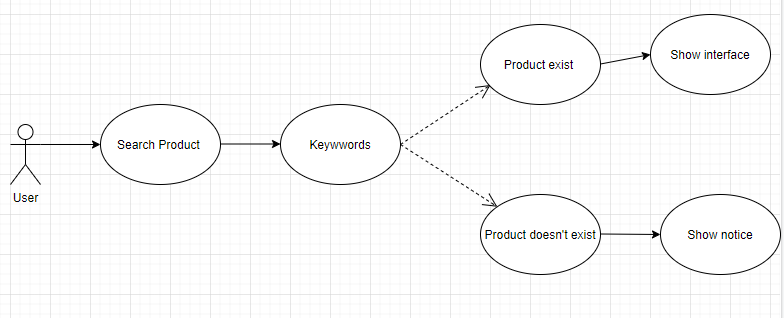
1. Register



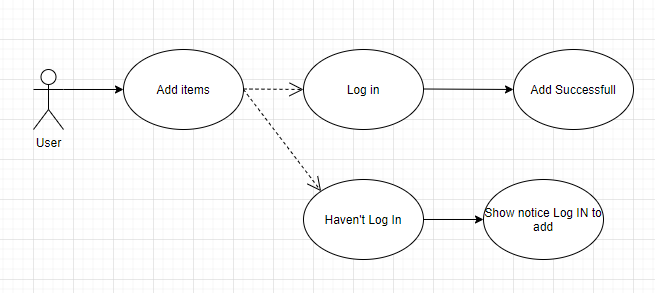
1. Login



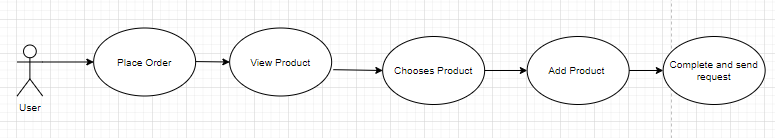
1. Search Product



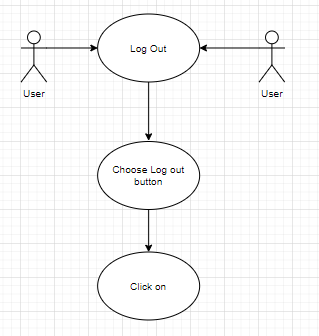
1. Add to cart



1. Place Order

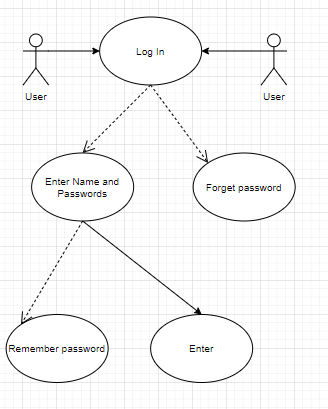


1. Logout

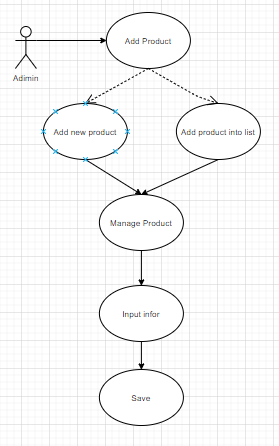


### ***IV.2.2. Administrator***

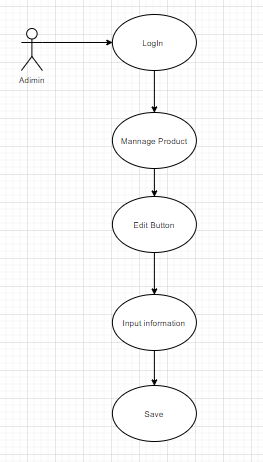
1. Login



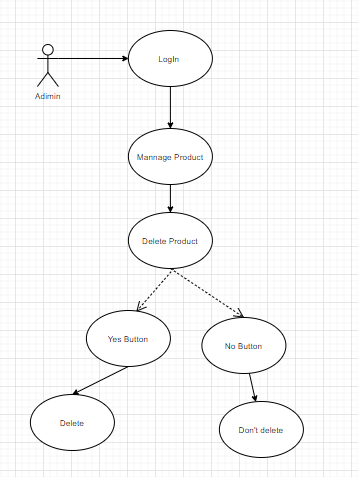
1. Add a new product



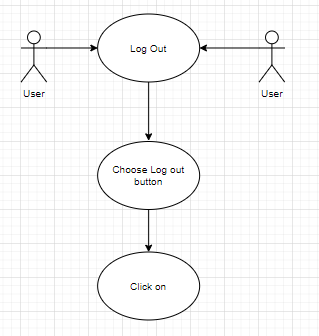
1. Edit Product



1. Delete product



1. Logout



## **IV.3. Use-Case Realizations**

### ***IV.3.1. Customer***

1. Register

|  |  |
| --- | --- |
| **Use case** | Register |
| **Description** | Users create a new account on the website |
| **Actors** | Customer |
| **Triggers** | User want to have an account the website |
| **Pre-Conditions** | User has reality email, name and true password |
| **Post-Conditions** | Sign-in board appears in the screen |
| **Main Event Flow** | - User choose the sign-up button  - Enter account name, password, phone number  - Check password again  - Press sign-in button |
| **Alternative Event Flow** |  |
| **Extension points** | If the information of user isn’t reliable, show the notice and sign-in again |
| **Special Requirements** | Having reliable information. |

1. Login

|  |  |
| --- | --- |
| **Use case** | Login Account |
| **Description** | Users login the website by their account |
| **Actors** | Users (Customer, Admin) |
| **Triggers** | User want to enter the website to buy clothes |
| **Pre-Conditions** | User has accessed to the system |
| **Post-Conditions** | Login board appears in the screen |
| **Main Event Flow** | - User enter the website  - Choose items and check out  - User enter name and password  - Use click login button |
| **Alternative Event Flow** |  |
| **Extension points** | If it is the unsafe password, show the notice. |
| **Special Requirements** | Having legal account |

1. Search Product

|  |  |
| --- | --- |
| **Use case** | Search Product |
| **Description** | Users search products by entering keywords |
| **Actors** | Users (Customer, Guest, Admin) |
| **Triggers** | User want to search for desired or related products |
| **Pre-Conditions** | User has accessed to the system |
| **Post-Conditions** | Searched products will be displayed on the interface if the keyword is matched |
| **Main Event Flow** | - User is in the shop’s homepage  - User enter product’s keyword  - Use click search button  - List of related products is displayed |
| **Alternative Event Flow** | User can search product by using drop down categories menu |
| **Extension points** | No keyword matched |
| **Special Requirements** | Keyword-related products are sorted in order of best matches |

1. Add to cart

|  |  |
| --- | --- |
| **Use case** | Add to Cart |
| **Description** | User adds products that they want to buy into the shopping cart |
| **Actors** | Customer |
| **Triggers** | Customer selects desired item |
| **Pre-Conditions** | User has logged in the system as a customer |
| **Post-Conditions** | Products are successfully added to shopping cart |
| **Main Event Flow** | - User logs in the system as a customer  - Customer chooses desired product  - Customer select features and quantity of the product  - Customer click Add to cart button  - Products are displayed in shopping cart list |
| **Alternative Event Flow** | None |
| **Extension points** | User has not logged in |
| **Special Requirements** | User has basic knowledge of operating computer and user interface |

1. Place Order

|  |  |
| --- | --- |
| **Use case** | Place Order |
| **Description** | Customer orders products by filling in order form and clicking order button |
| **Actors** | Customer |
| **Triggers** | Customer want to buy the products |
| **Pre-Conditions** | Products have added to shopping cart |
| **Post-Conditions** | Orders are sent to the shop owner |
| **Main Event Flow** | - Customer goes to view shopping cart  - Customer chooses products that they want to buy  - Customer clicks on order button  - Customer completes order form  - Customer clicks Send order button |
| **Alternative Event Flow** | None |
| **Extension points** | Customer checks the order again after having sent the order. |
| **Special Requirements** | Accuracy is required when filling in the order form in order to save time and effort. |

1. Logout

|  |  |
| --- | --- |
| **Use case** | Log-out Account |
| **Description** | Users log-out the website by their account |
| **Actors** | Users (Customer, Admin) |
| **Triggers** | User want to exit the website |
| **Pre-Conditions** | User has accessed and login account successfully to the system |
| **Post-Conditions** | Log-out board appears in the screen |
| **Main Event Flow** | - User choose the log-out button  - Press enter log-out |
| **Alternative Event Flow** |  |
| **Extension points** |  |
| **Special Requirements** | Having legal and login account |

### ***IV.3.2. Administrator***

1. Login

|  |  |
| --- | --- |
| **Use case** | Login Account |
| **Description** | Users login the website by their account |
| **Actors** | Users (Customer, Admin) |
| **Triggers** | User want to enter the website to buy clothes |
| **Pre-Conditions** | User has accessed to the system |
| **Post-Conditions** | Login board appears in the screen |
| **Main Event Flow** | - User enter the website  - Choose items and check out  - User enter name and password  - Use click login button |
| **Alternative Event Flow** |  |
| **Extension points** | If it is the unsafe password, show the notice. |
| **Special Requirements** | Having legal account |

1. Add a new product

|  |  |
| --- | --- |
| **Use case** | Add Product |
| **Description** | Admin inputs information of a new product and adds this product into product list |
| **Actors** | Admins |
| **Triggers** | Admins want to add new products to the shop's business directory |
| **Pre-Conditions** | User login with an admin account and come to admin page |
| **Post-Conditions** | New products will be inserted to the database and be displayed in products list on admin page as well as on customer pages. |
| **Main Event Flow** | - Admin is in the admin page  - Admin clicks “Manage product” button in the navigation sidebar  - Admin clicks “Add new product” in manage product page  - Admin inputs product information and click save |
| **Alternative Event Flow** | None |
| **Extension points** | No keyword matched |
| **Special Requirements** | Input values must satisfy the constraints of corresponding attributes |

1. Edit Product

|  |  |
| --- | --- |
| **Use case** | Edit Product |
| **Description** | Admin edits information of an available product |
| **Actors** | Admins |
| **Triggers** | Admins want to edit information of an available product to satisfy current product condition |
| **Pre-Conditions** | User login with an admin account and come to admin page |
| **Post-Conditions** | Information of an available product will be edited in products list on admin page as well as on customer pages and updated in database |
| **Main Event Flow** | - Admin is in the admin page  - Admin clicks “Manage product” button in the navigation sidebar  - Admin clicks Edit button in row of product he/she want to edit  - Admin inputs product information and click save |
| **Alternative Event Flow** | None |
| **Extension points** | No keyword matched |
| **Special Requirements** | Input values must satisfy the constraints of corresponding attributes |

1. Delete product

|  |  |
| --- | --- |
| **Use case** | Delete Product |
| **Description** | Admin deletes a product from product list |
| **Actors** | Admins |
| **Triggers** | Admins want to delete a product from the shop's business directory |
| **Pre-Conditions** | User login with an admin account and come to admin page |
| **Post-Conditions** | A product will be deleted from the database and from products list on admin page as well as on customer pages. |
| **Main Event Flow** | - Admin is in the admin page  - Admin clicks “Manage product” button in the navigation sidebar  - Admin clicks Delete button in the raw of product he/she want to delete  - In alert dialog, Admin click “Yes” to delete or “Cancel” to cancel delete |
| **Alternative Event Flow** | None |
| **Extension points** | No keyword matched |
| **Special Requirements** | None |

1. Logout

|  |  |
| --- | --- |
| **Use case** | Log-out Account |
| **Description** | Users log-out the website by their account |
| **Actors** | Users (Customer, Admin) |
| **Triggers** | User want to exit the website |
| **Pre-Conditions** | User has accessed and login account successfully to the system |
| **Post-Conditions** | Log-out board appears in the screen |
| **Main Event Flow** | - User choose the log-out button  - Press enter log-out |
| **Alternative Event Flow** |  |
| **Extension points** |  |
| **Special Requirements** | Having legal and login account |

# **V. LOGICAL VIEW**

## **V.1. Overview**

### **V.1.1. Subsystems**

System FSMF(FourSeasons Shop Management System) can be divided into 2 main subsystems:

-Customer management subsystem

-Administrator management subsystem

*V.1.1.1 Customer management sub system*

This sub system provides the facilities that cover all the user management functionalities. Main use cases that comes under this subsystem includes

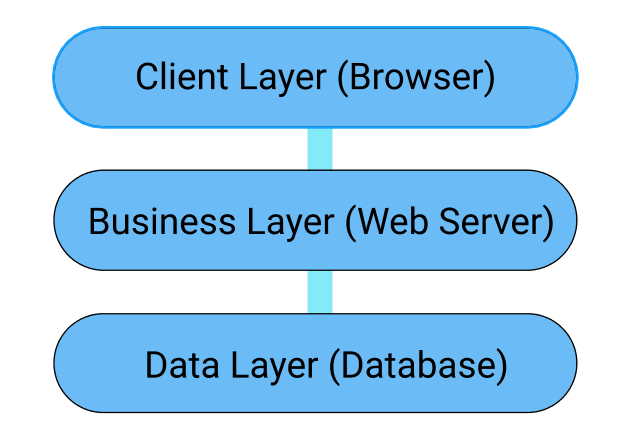
* Register
* Login
* Logout
* Search Product
* Place Order
* Add to cart

*V.1.1.2 Administrator management subsystem*

* Add a new product
* Edit product
* Delete product

### **V.1.2. Layering**

FourSeasons Shop Management System is divided into 3 layers:

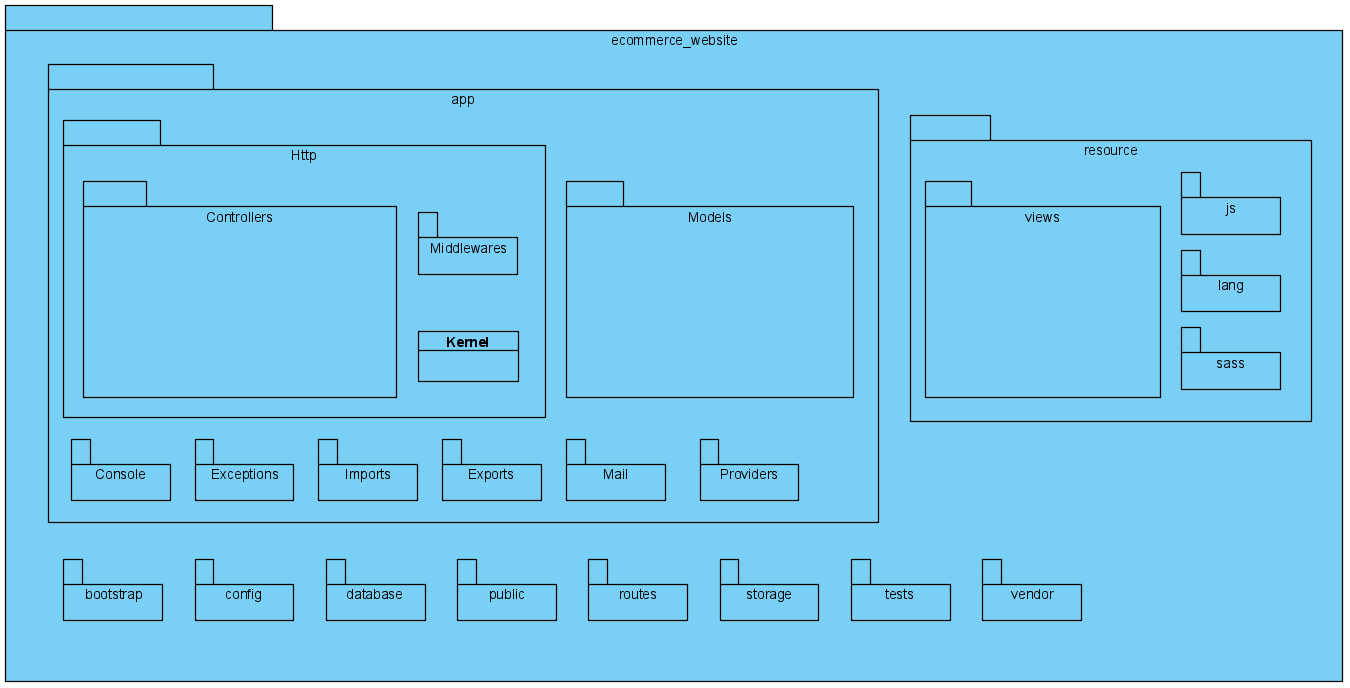


## **V.2. Architecturally significant Design Packages**

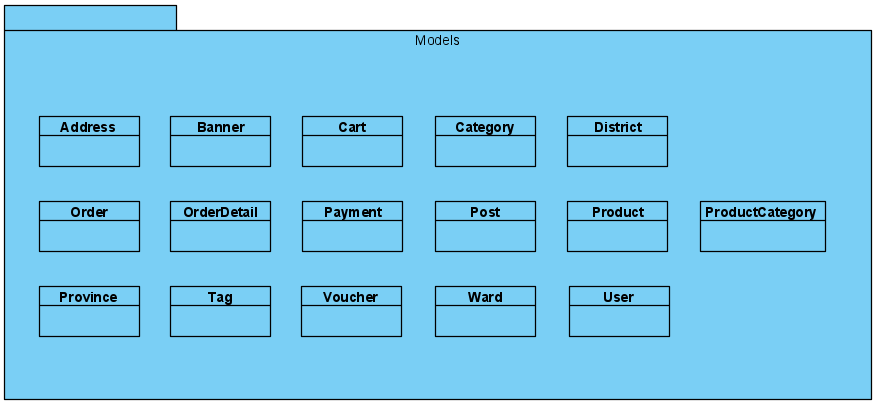
### **V.2.1 Class diagram**

FourSeasons Shop Management System is designed based on MVC design pattern and using Laravel Framework. In several diagrams below, Backend means the Admin side of the system and Frontend means the Customer side.

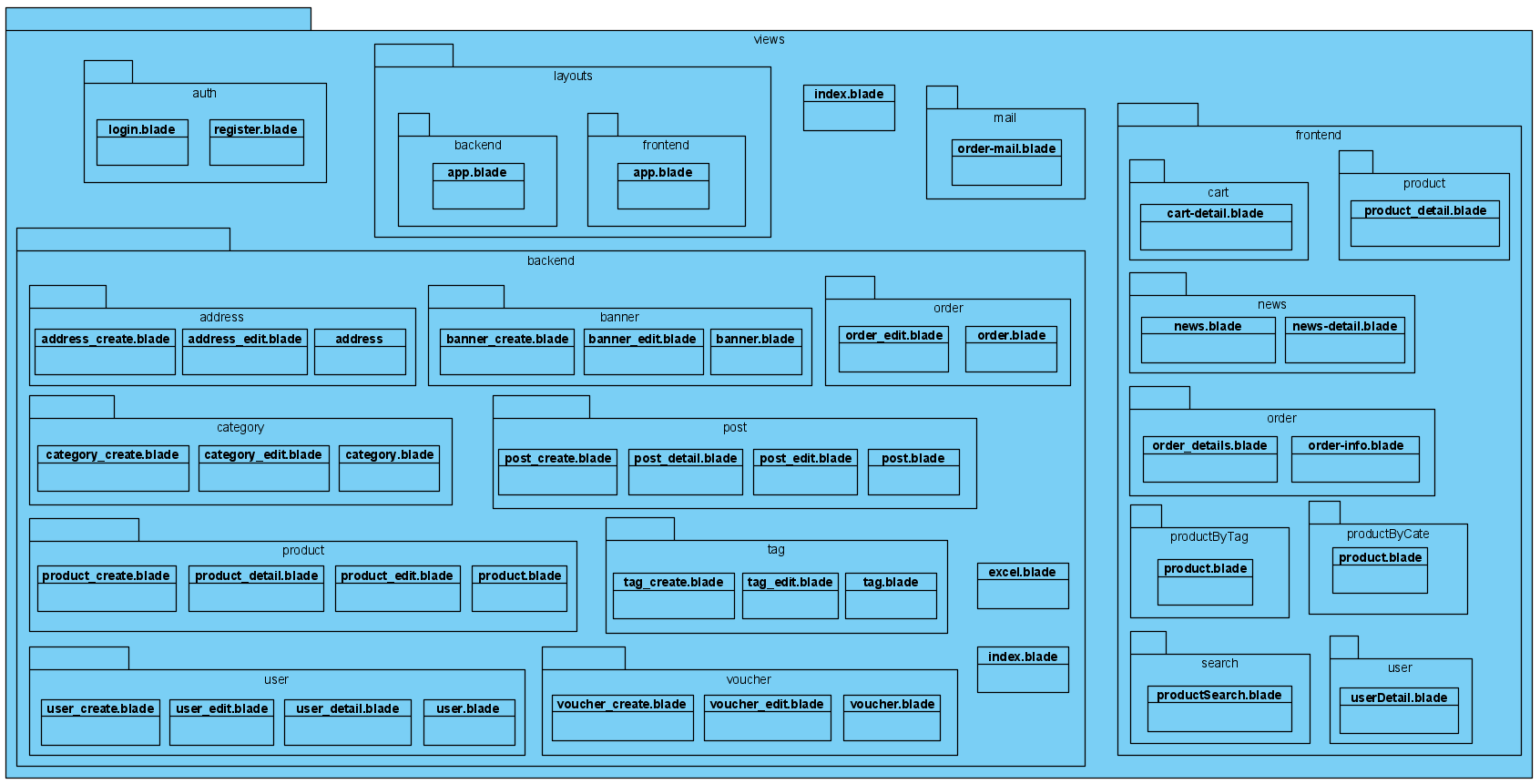
Project Directory Structure:



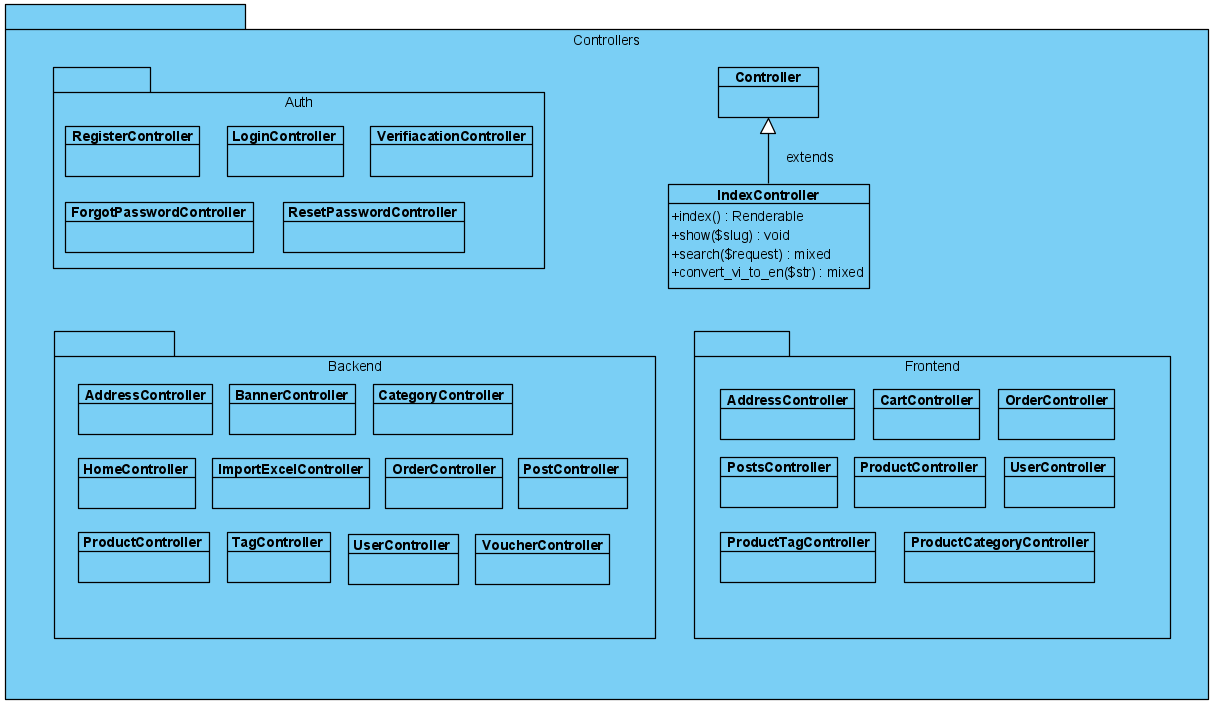
Models:



Views:

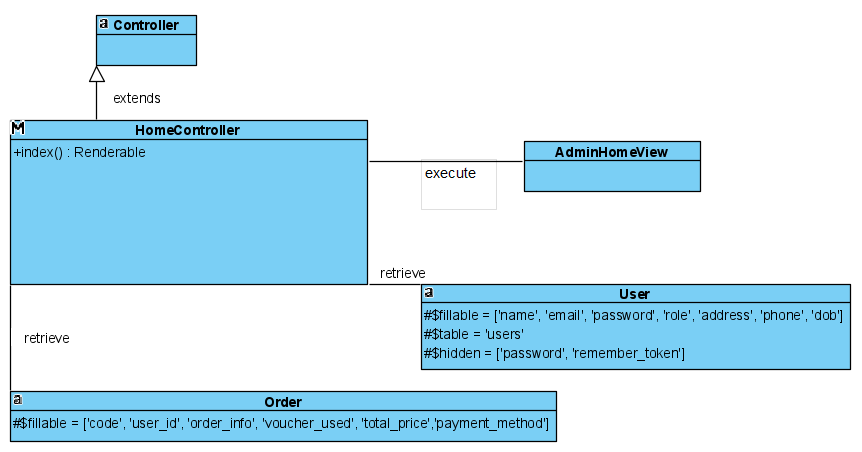


Controllers:

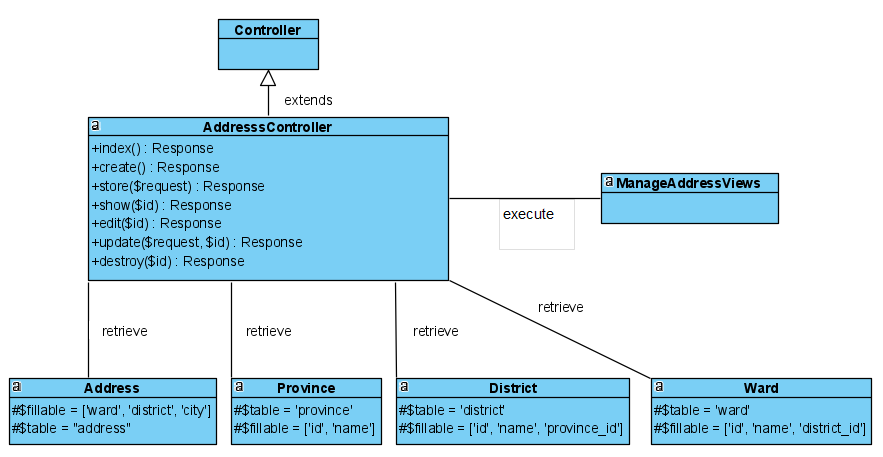


The interaction between classes follows the MVC model:

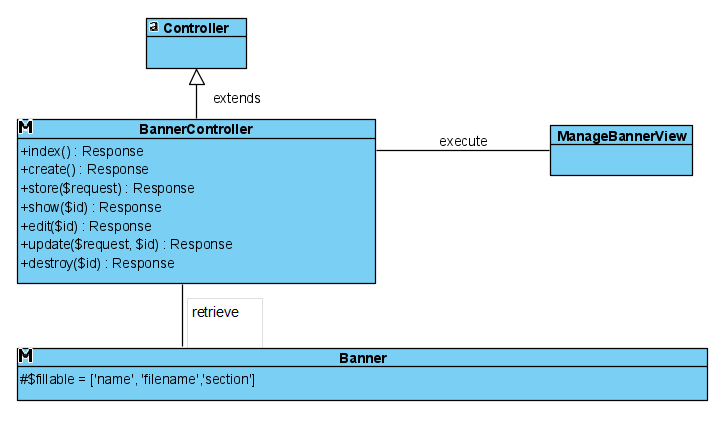
* Backend (Admin) side:
* Render Admin HomePage (Dashboard):



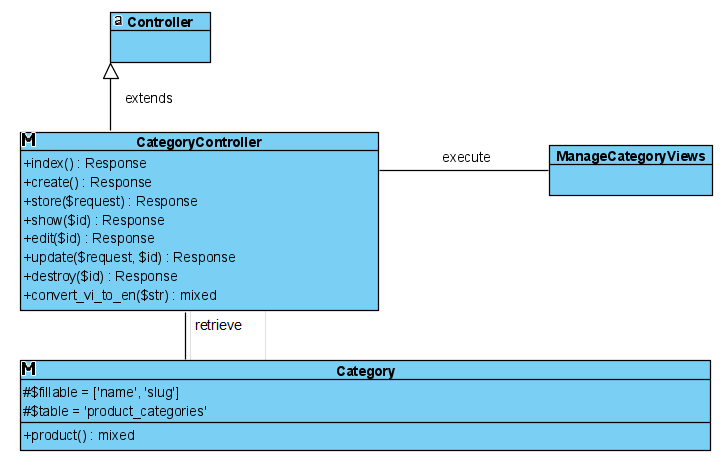
* Manage Address of store’s branches:



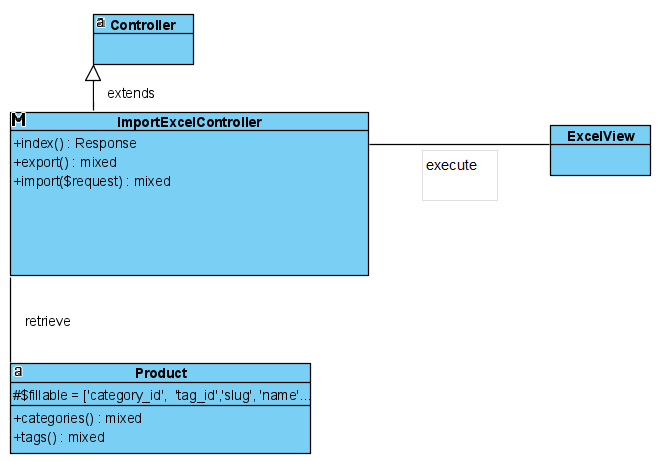
* Manage Banner:



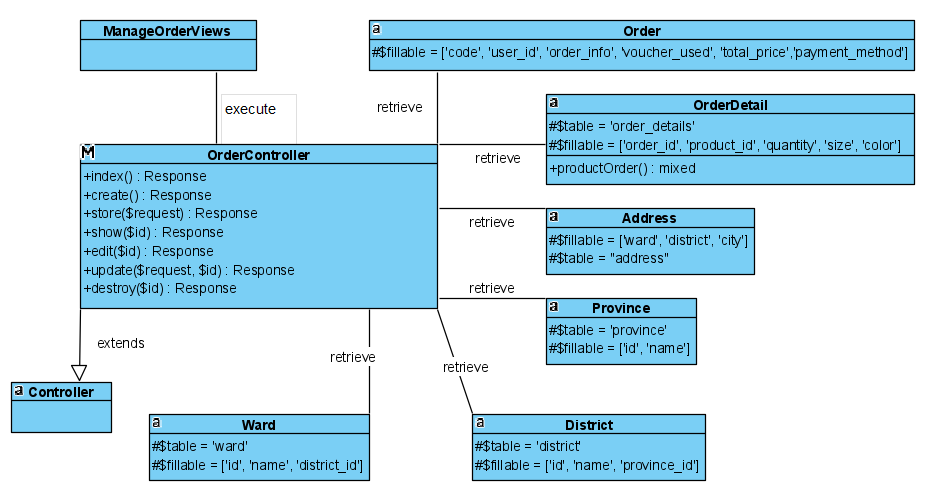
* Manage Category:



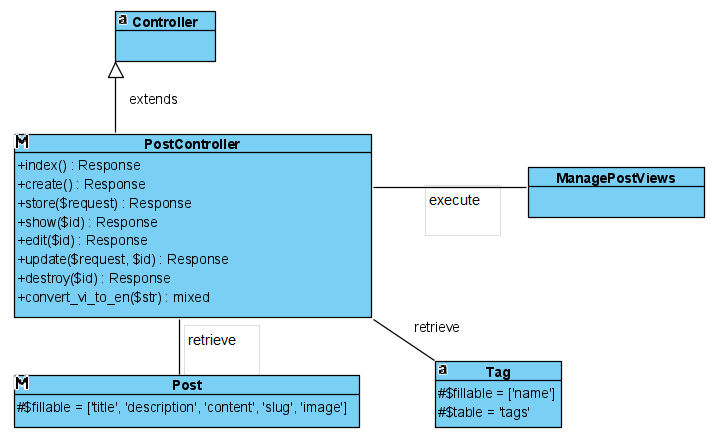
* Manage import/export excel files about product detail:



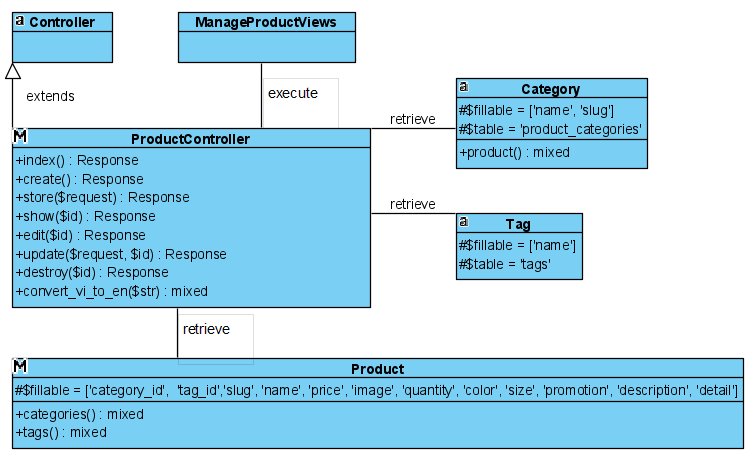
* Manage order:



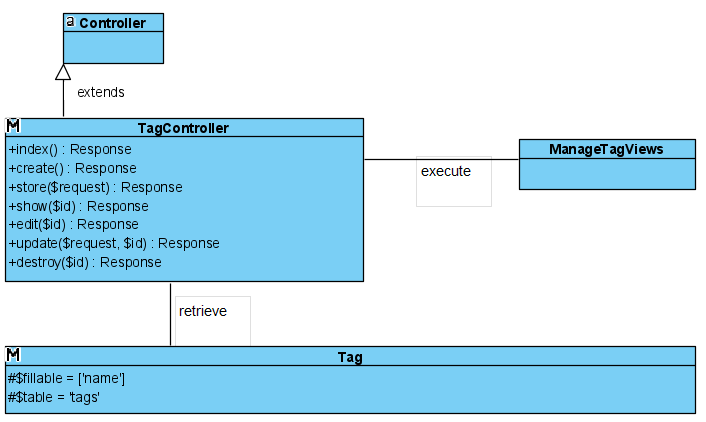
* Manage post:



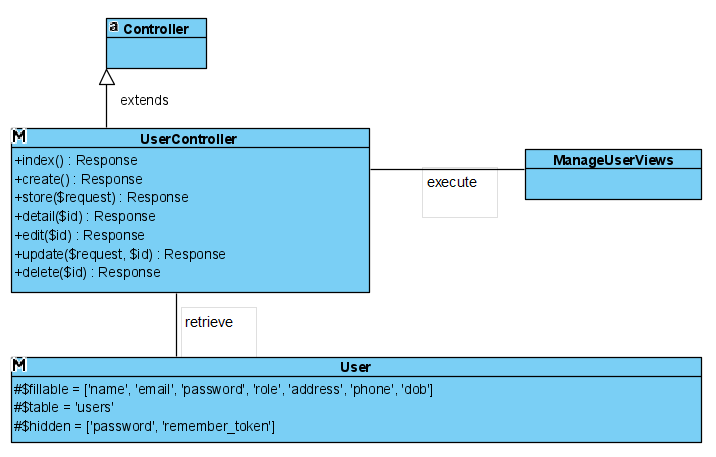
* Manage Product:



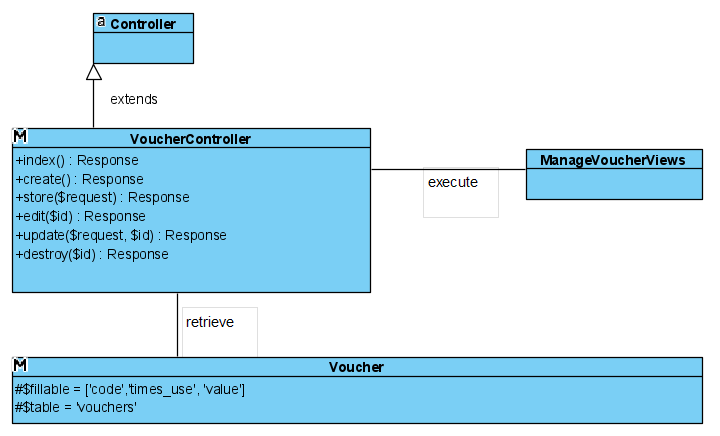
* Manage Tag:



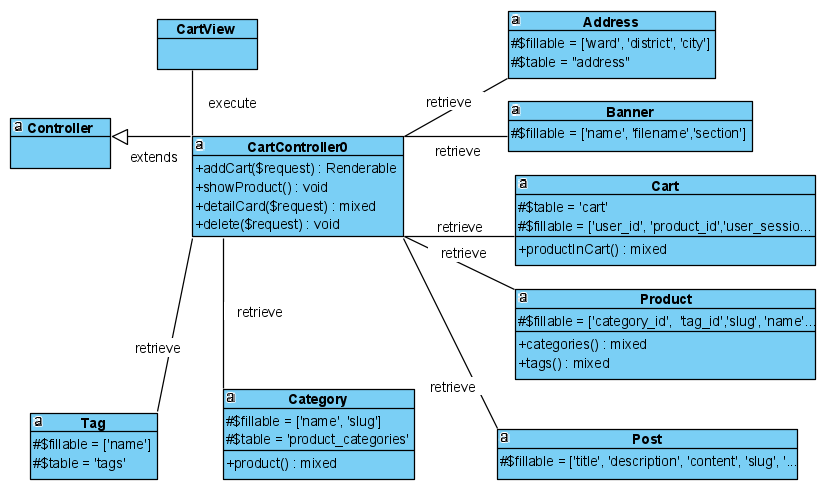
* Manage User:



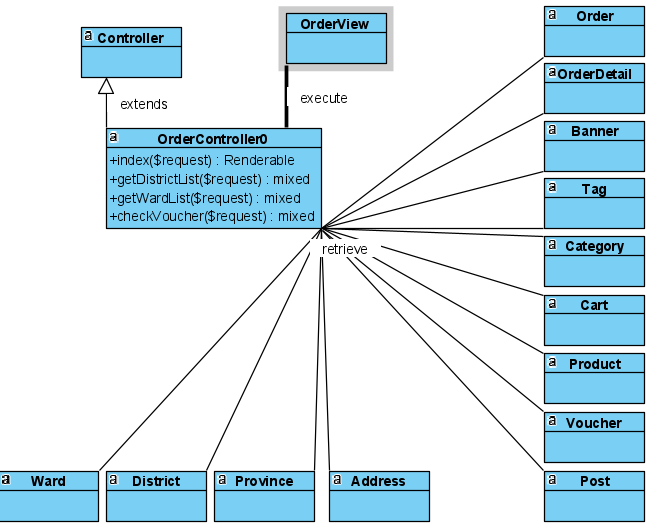
* Manage Voucher:



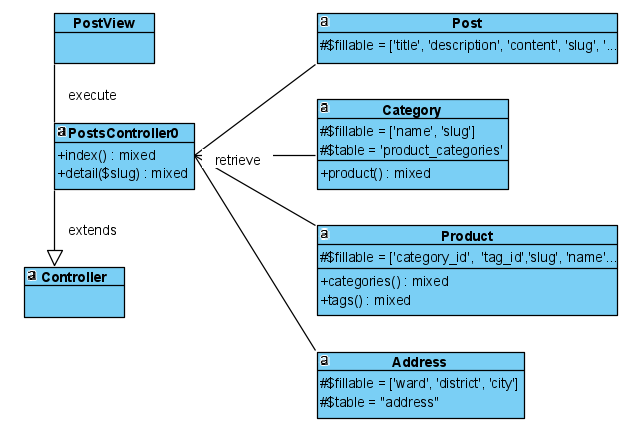
* Frontend (Customer) side (“0” is added at the end of class name of controllers in class diagram because Visual Paradigm do not allow existence of 2 classes with the same name in a project, means that in code, class name of controllers do not include “0”):
* Manipulate with Cart:



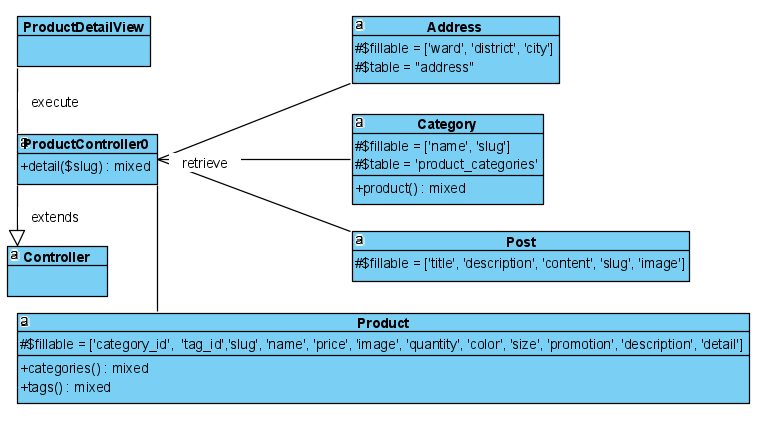
* Order products in cart:



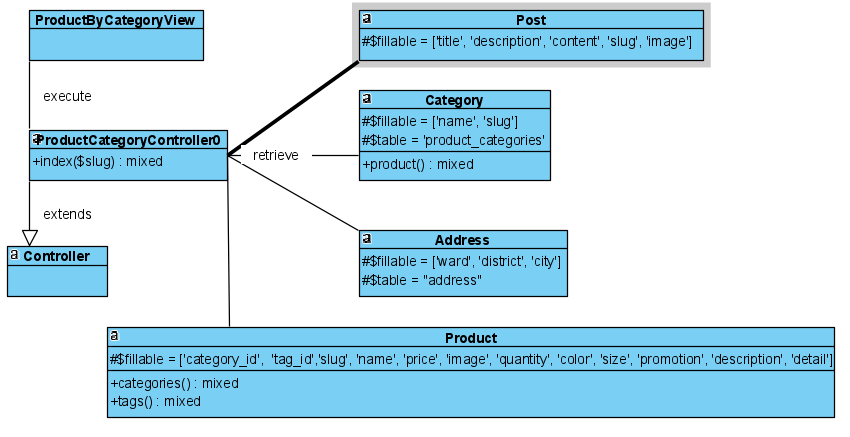
* View Post:



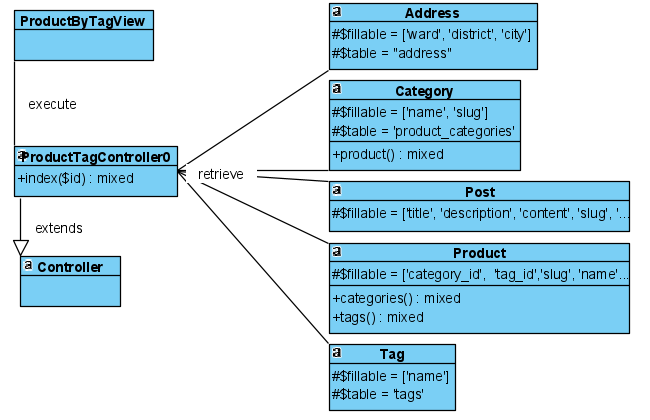
* View Product details:



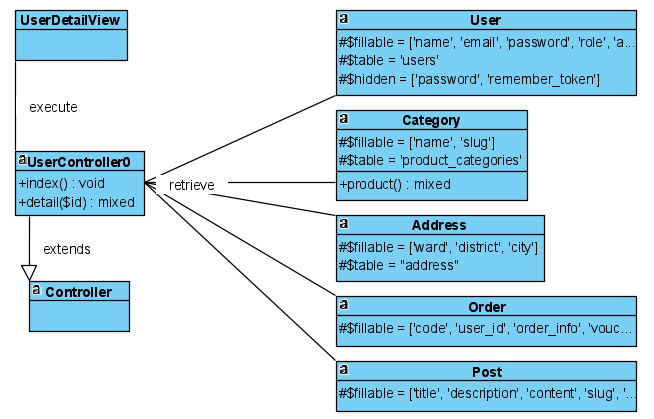
* View Product By Category:



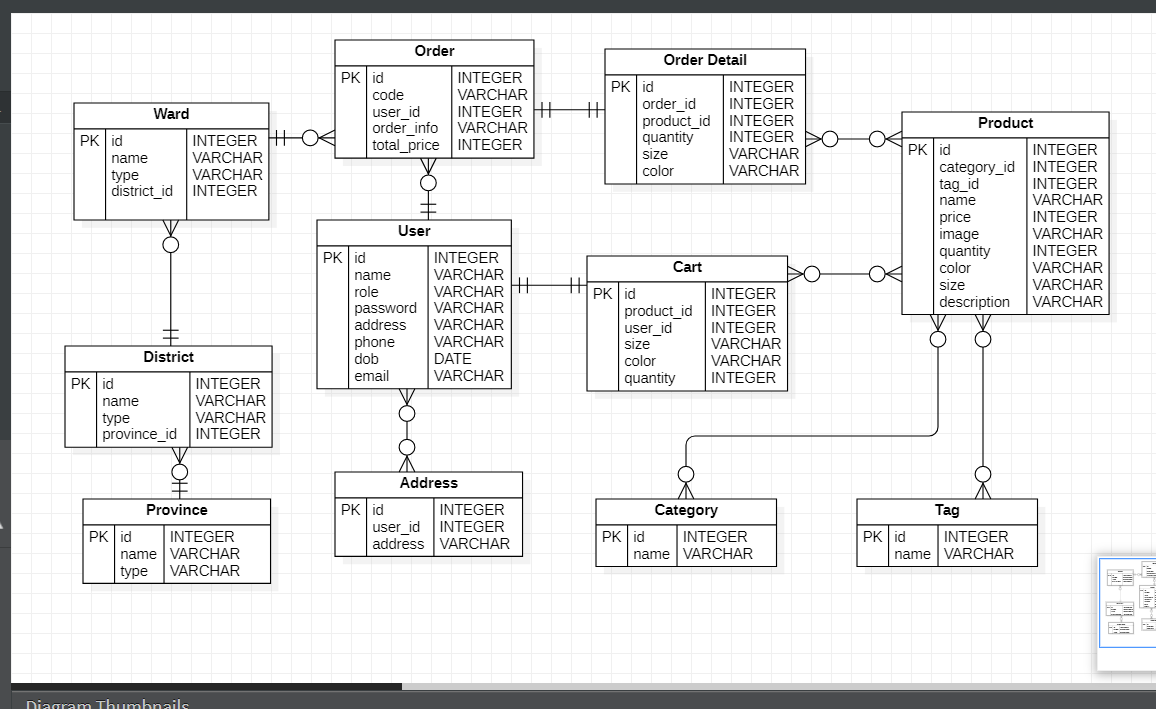
* View Product By Tag:



* View User details:



### **V.2.2 ER diagram**

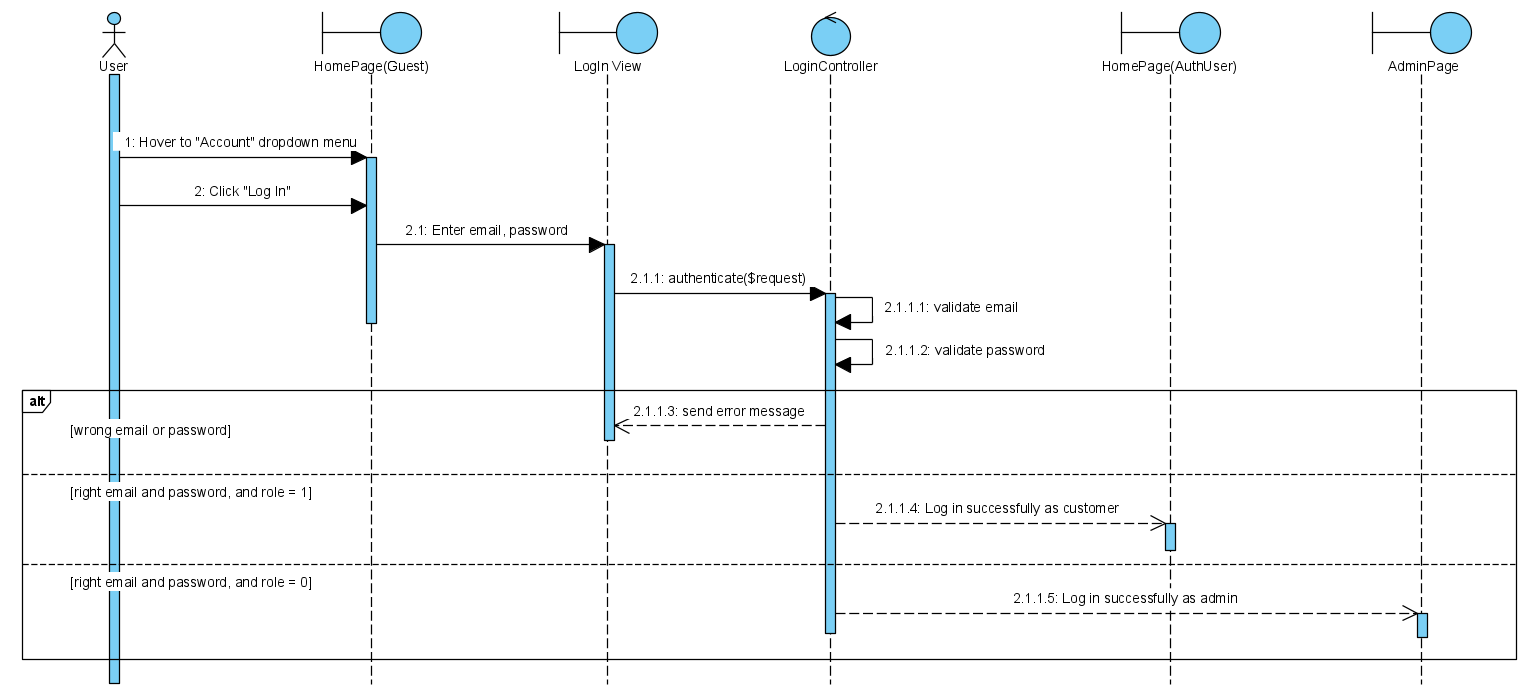


# **VI. PROCESS VIEW**

## **VI.1. System sequence diagram**

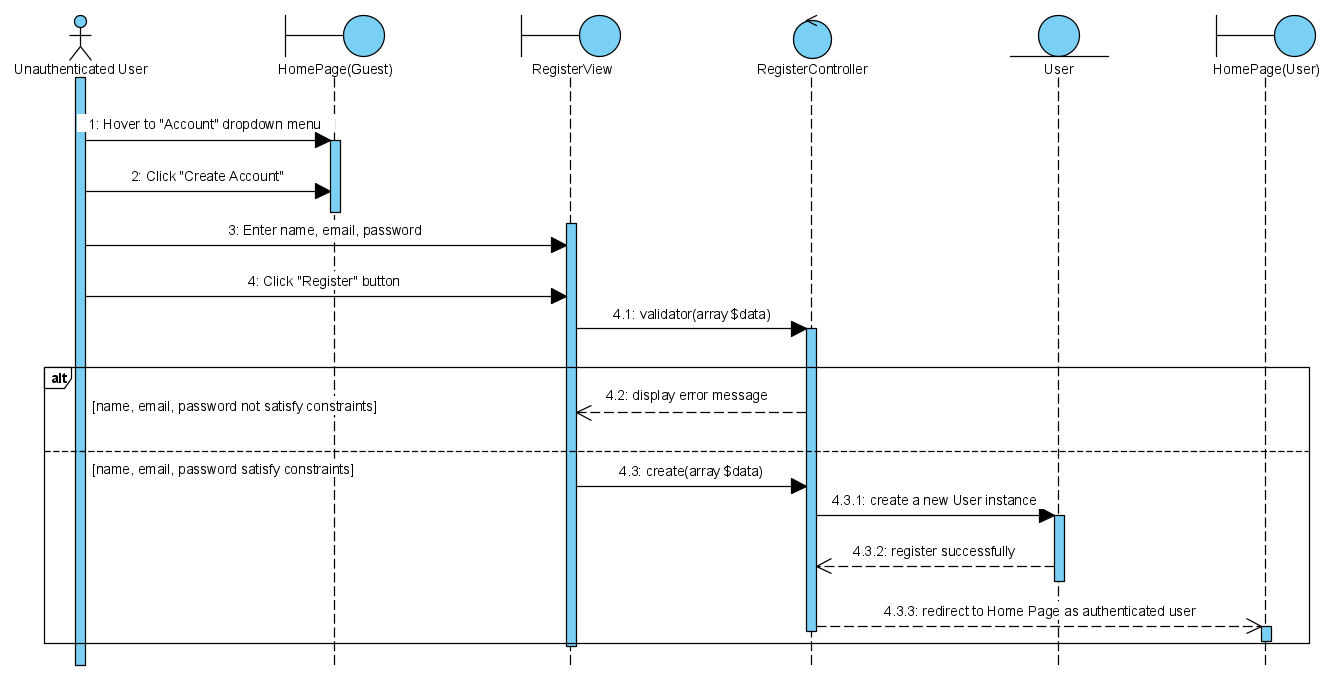
1. ***User log into the system (for both admin and customer)***

If the users click the Login function when hovering to the “Account” dropdown menu, the system will navigate to another page with a login form which contains 2 fields: email and password. They fill in the form with their account and choose to save their login information for the next time by ticking the box "Remember Password". After that, they can click the Login button. The system checks for the validity and the matching of input values and redirects to the home page of the user (Admin or Customer). If the email address or password is wrong, the system will prevent customers from signing in and display a warning message that requires customers to enter again.



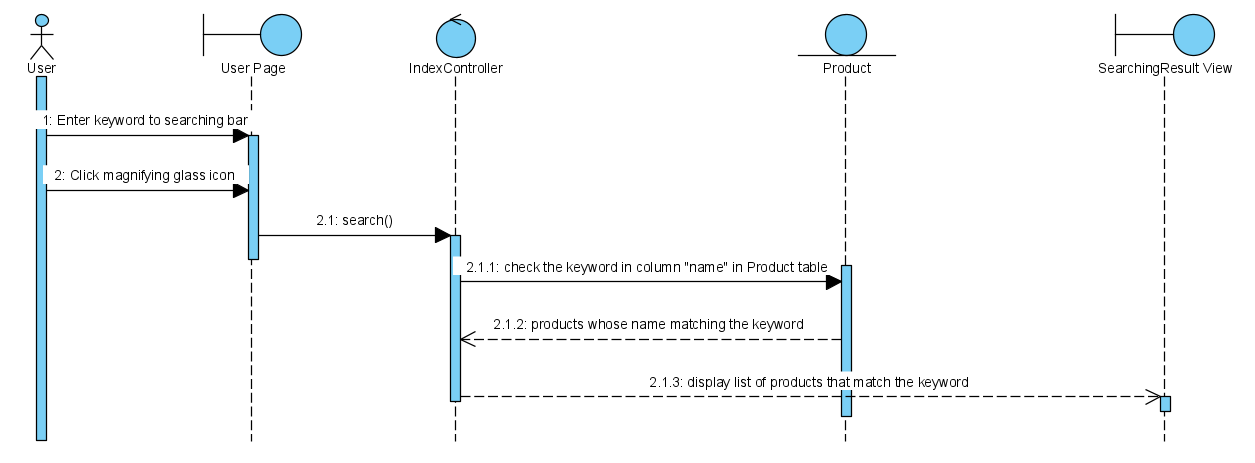
1. ***Customers sign up to the system***

If the customers do not have any account, they should click the “Register” button when hovering to the “Account” dropdown menu. The system will navigate to another page with a register form, which requires the customer to enter the necessary information (full name, email, password and confirm password). After entering all information and ticking the box “I agree to the terms”, customers click the “Register” button. If all information is valid, the system will create a new account in the database and navigate to Home Page as this account. In some cases (email is duplicated or password and confirm password does not match), the system shows a warning and prevents adding an account.



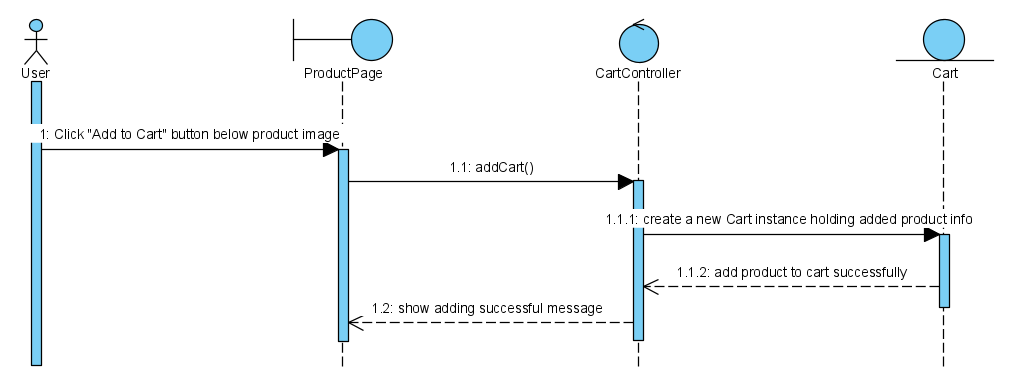
1. ***Customers search a product from the website***

When customers want to find a product from the website, they can click the “search” button and enter the keywords related to them. After that, customers click the icon "Search", the system will run the query and announce the number of matching search results. If there are products that match the keywords, they will be displayed on the website.



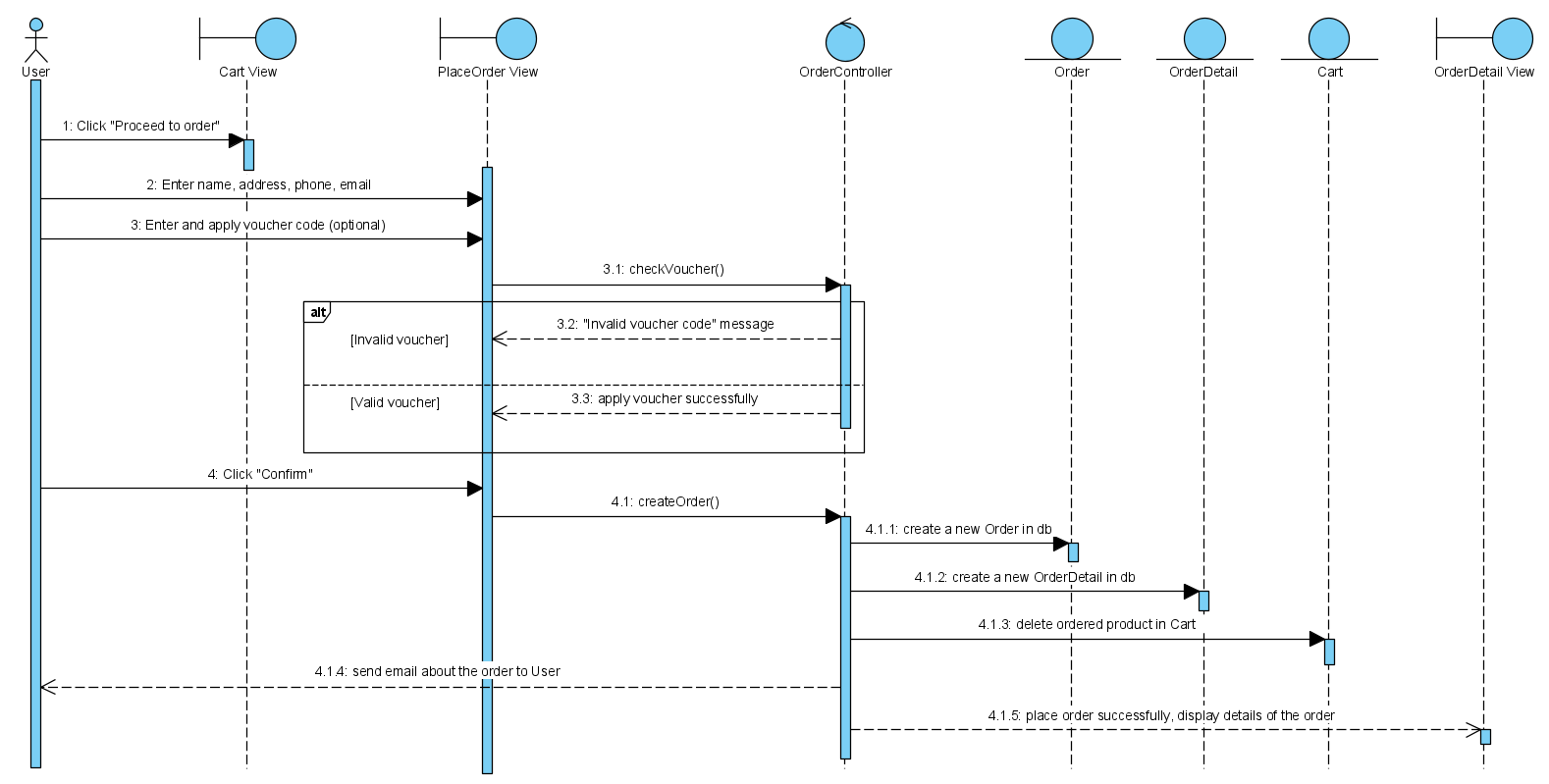
1. ***Customers add product to shopping cart***

When customers find some interesting products, they want to store information about these products. They only need to click the “Add to cart” button below each product’s image. This action will keep these products in the shopping cart.



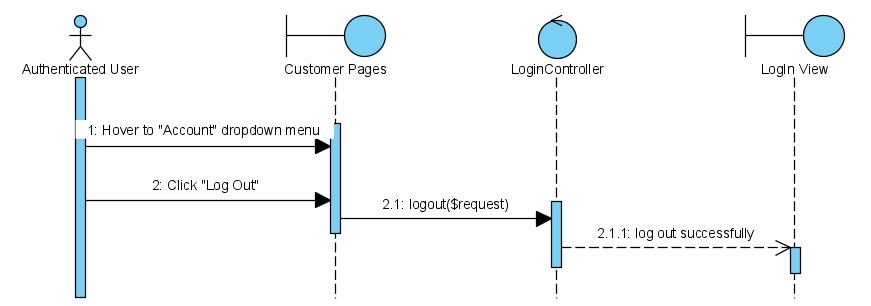
1. ***Customers place an order***

When the customers click the "Proceed to order" button at the right corner of the Cart page, the system will take them to the checkout page. There is an order form which requires customers to fill some information: Full Name, Phone Number, Email and Address. They can select the payment method: Cash on delivery or Payment via bank transfer, then enter a discount code (if applicable). After that the system checks the voucher, if invalid, an error message will appear. Otherwise, customers will receive an email about their order after clicking the "Order Confirmation" button. The screen will display a message (“Place order successfully”) and the details of this order.



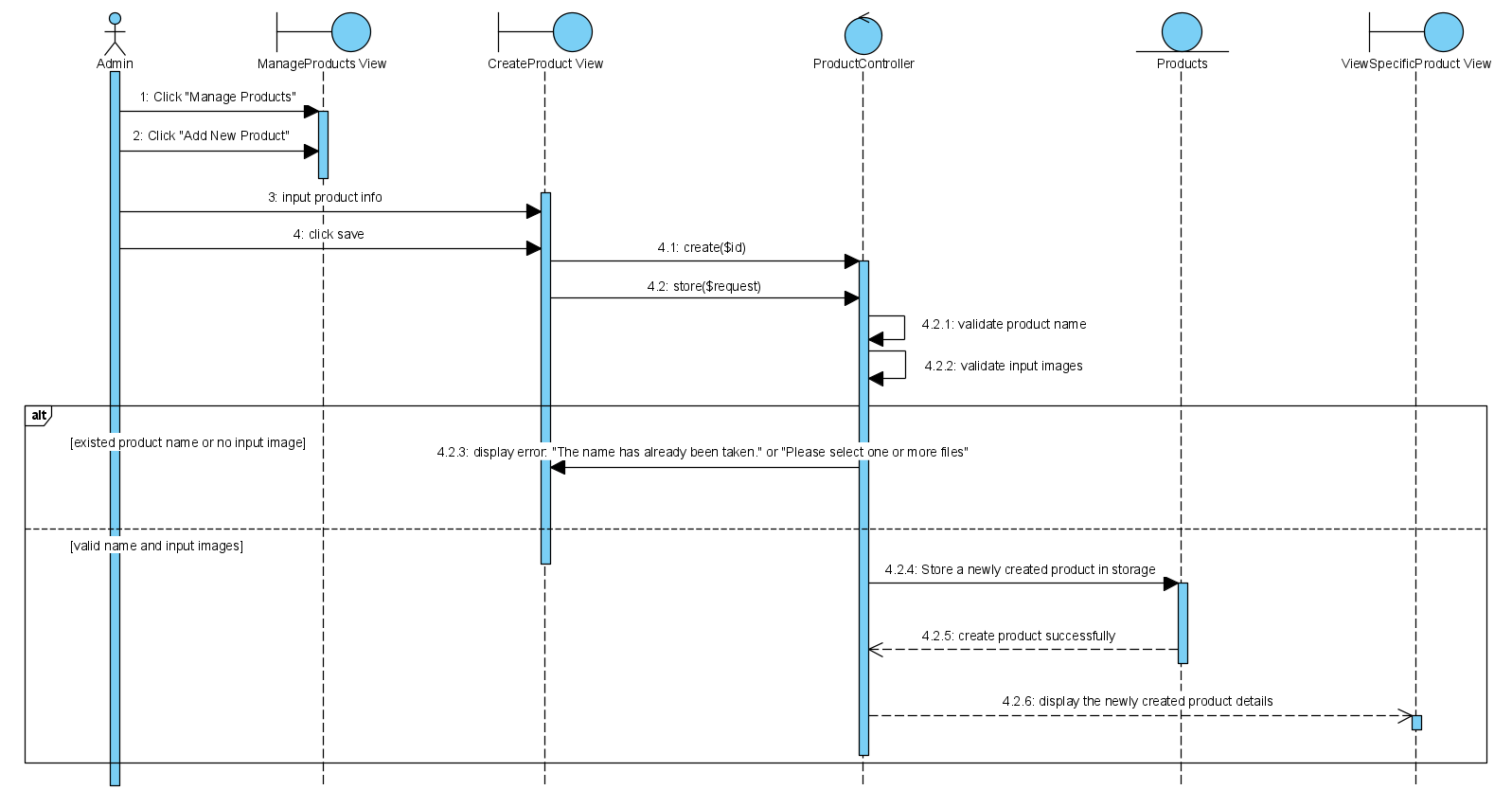
1. ***Customers log out the system***

If the customers choose the “Logout” button when hovering to the “Account” dropdown menu, the system will log out of their account and return to the website as a visitor.



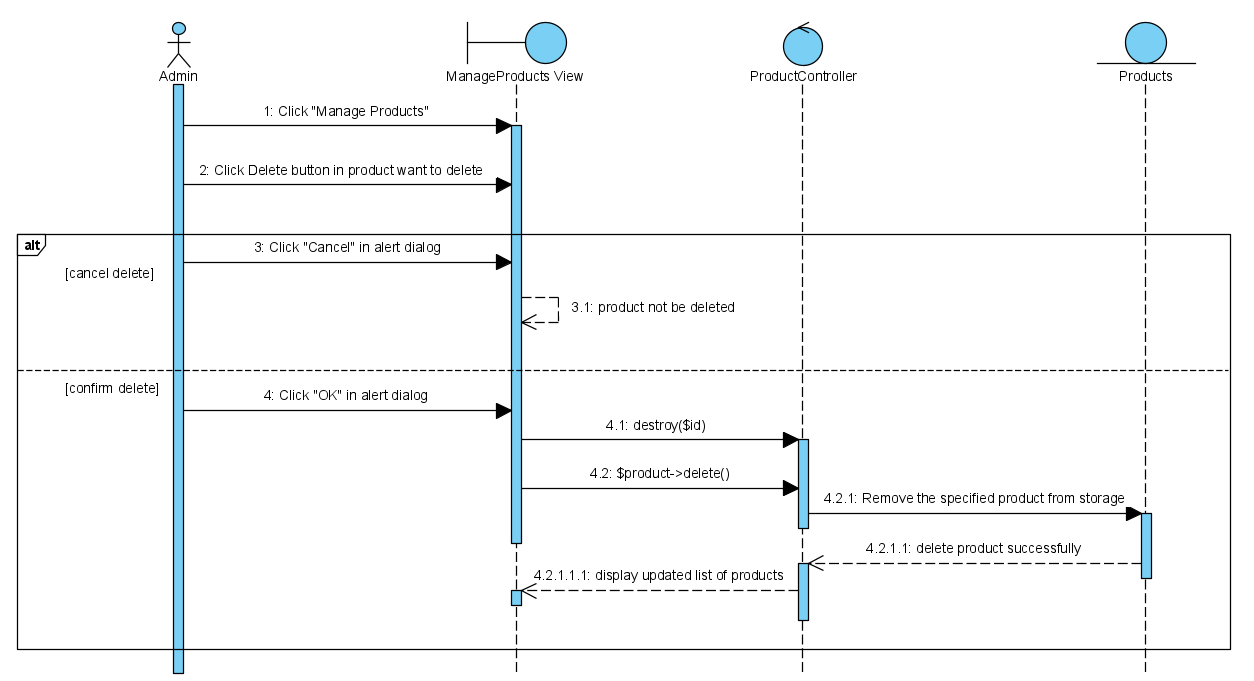
1. ***Administrator adds a new product***

The administrator clicks “Product Management” in the Sidebar Menu of Admin Page and chooses the “Add a new product” button. When clicking on this function, there is a product form for the admin to insert information for the new product. After filling in the product form, the admin clicks the “add” button at the end of this form. This action will add this product to the database if all fields are valid, and display an error message if there is an invalid field.



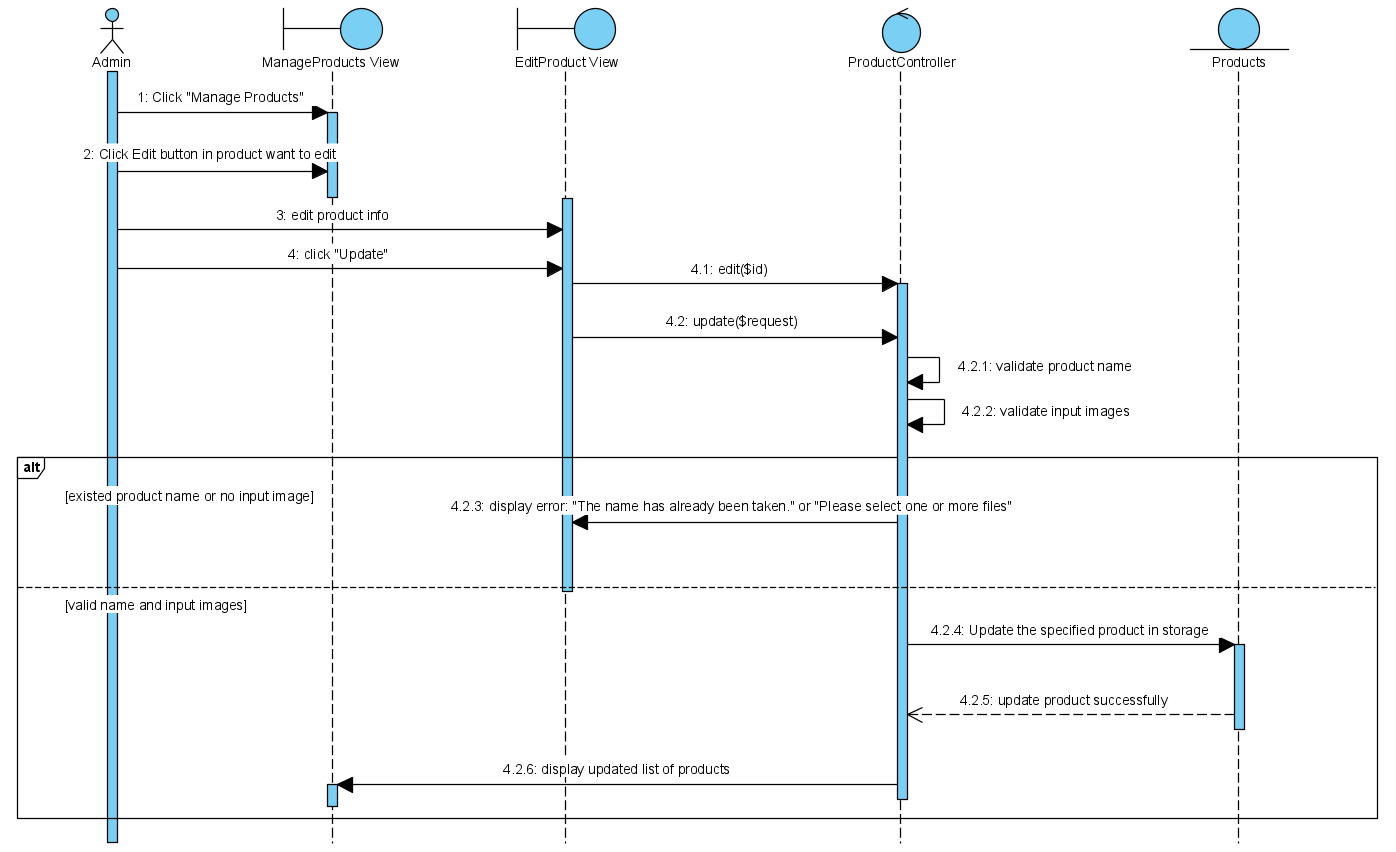
1. ***Administrator deletes a product***

After going to the “Product management” page, the administrator can delete a product from the website by clicking the “delete” icon in the “Action” column. This product is deleted from the website and database if the admin clicks “OK” in “Delete Confirmation” dialog.



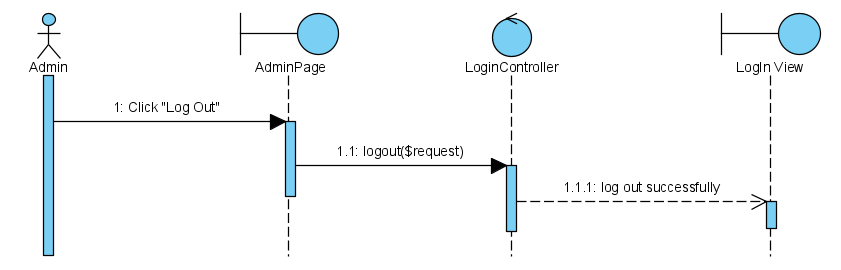
1. ***Administrator updates a product***

After going to the “Product management” page, the administrator can update product’s information by clicking the “edit” icon in the “Action” column. The edit form is shown on the website. Administrator enters information and chooses the “update” button. This modification will be updated on the website.



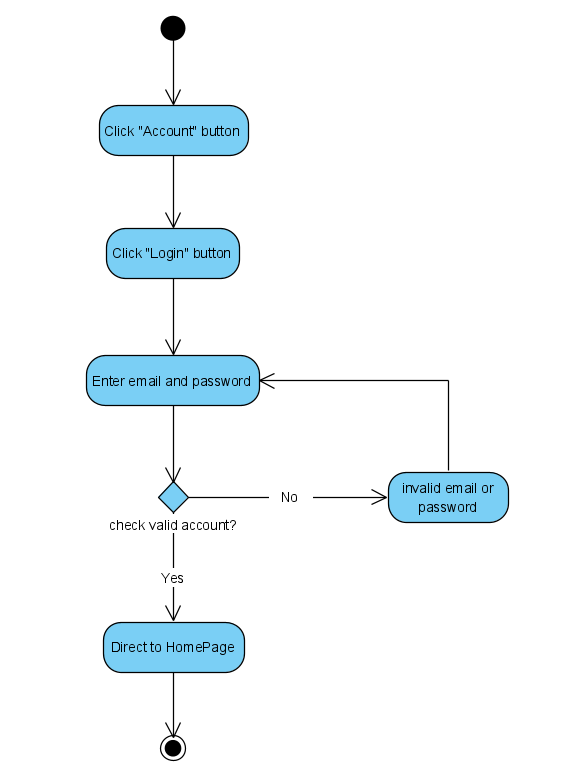
1. ***Administrator logs out the system***

The admin clicks the “Logout” button, the system will log out of their account and return to the Login Page.

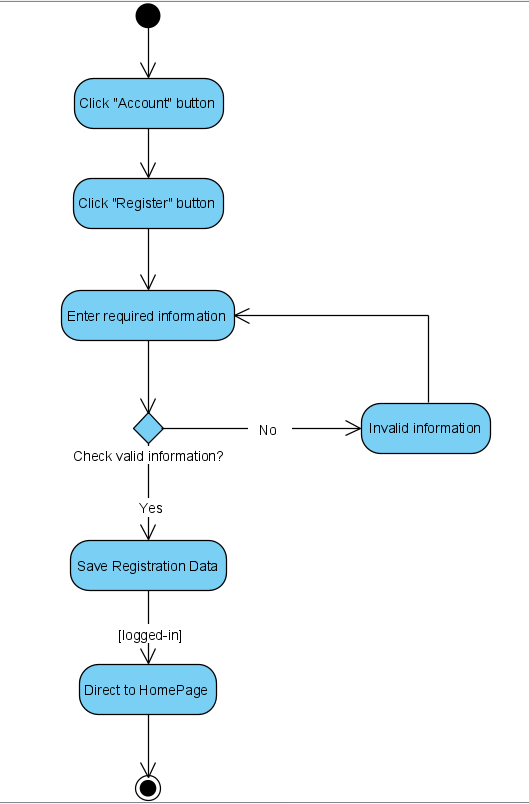


## **VI.2. Activity Diagrams**

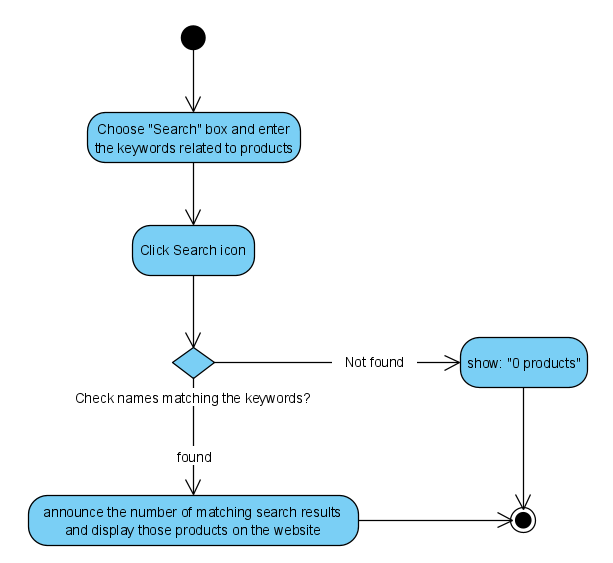
1. ***Login (both administrator and Customer)***



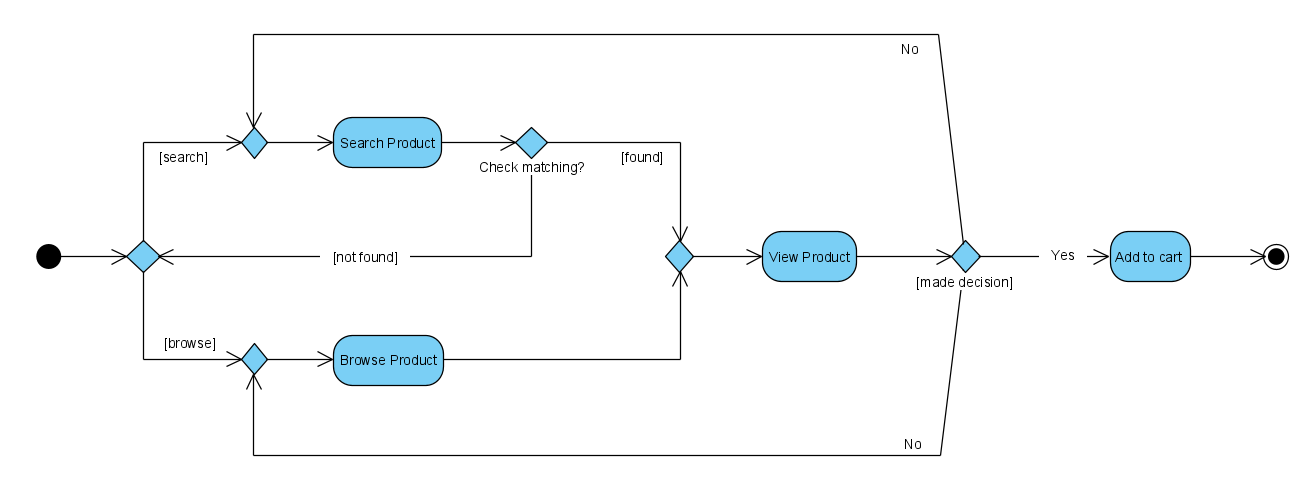
1. ***(Customer) Register***



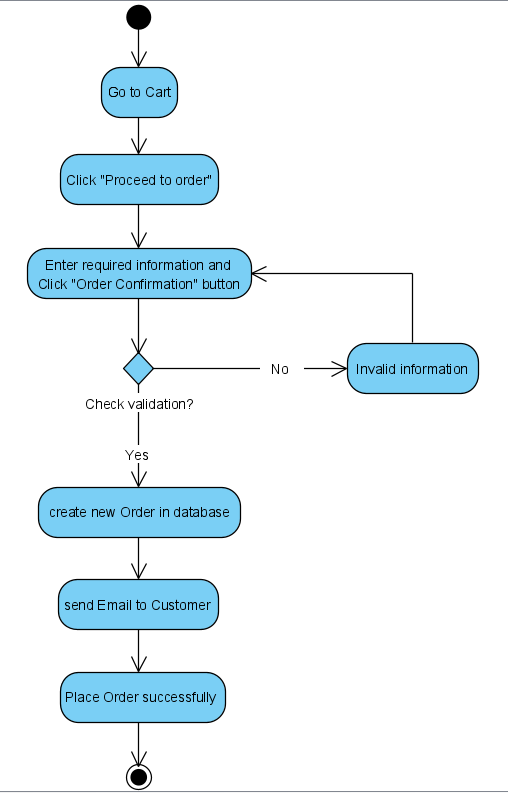
1. ***(Customer) Search product***

******

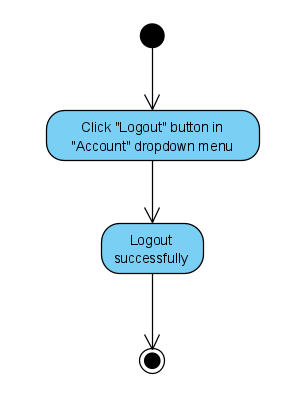
1. ***(Customer) Add product to cart***



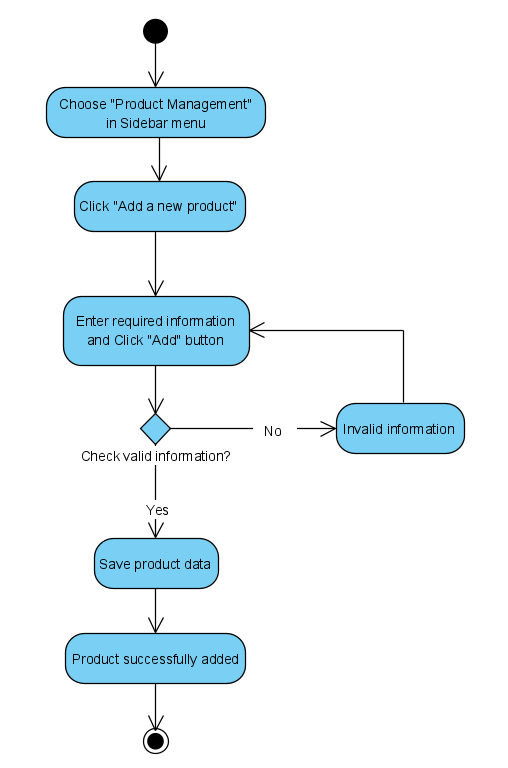
1. ***(Customer) Place Order***



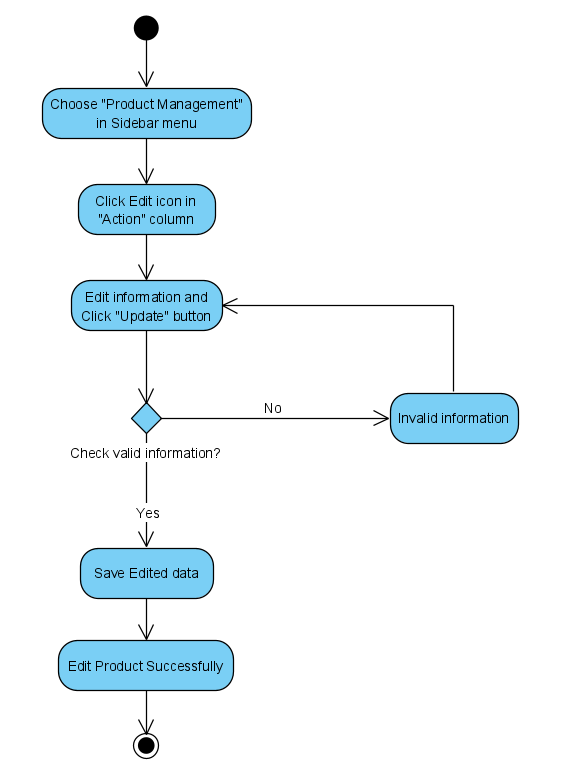
1. ***(Customer) Logout***

**

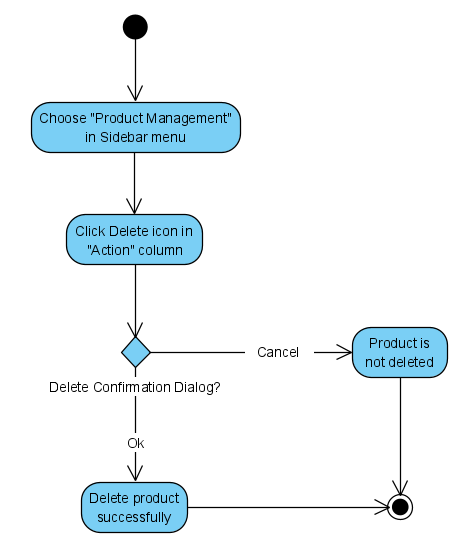
1. **(Administrator) *Add a new product***



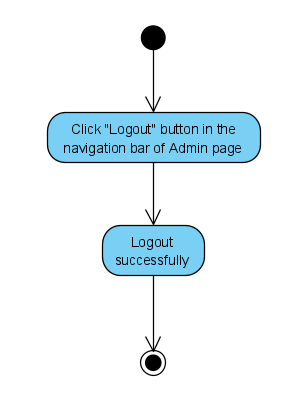
1. ***(Administrator) Update product***



1. ***(Administrator) Delete product***



1. ***(Administrator) Logout***



# **VII. DEPLOYMENT VIEW**

Being a web application this system is hosted in a remote server. The database will be hosted in some other hosting space. Except for the statistical graph generation, all the processing is done in the backend, so that the client computer does not have to spend much CPU power. In case of statistical graphs generation, all the data processing will be done in the front end, so that the client computer will need a particular level of performance.

# **VIII. IMPLEMENTATION VIEW**

FSMS is a website that follows the MVC architecture pattern. Main reason to use this pattern is to separate functions into layers thus improving the maintainability and reusability. View layer (AKA client layer) contains the graphical user interfaces (web pages). The actions of these web pages are handled by controller classes. Controller classes invoke and instantiate objects of model classes that contain the business login. Separating software packages in this manner reduces the complexity. The model classes can be subdivided into two layers. Business layer and the data access layer. Data access layer is manipulated using Bootstrap framework. Business layer contains the main entity classes such as Administrator, Customer and etc. For each entity in the database there exists a class in the software that provides data access operations to that entity.