Comnhalam.com

Version 1.0.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| November 14, 2018 | 1.0.0 | Initial document | Trần Hải Phong |
| December 1, 2018 | 1.0.1 | Detail for class diagrams and fix MVC framework | Trần Hải Phong |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 References 4

1.4 overview 4

2. Architectural Goals and Constraints 4

3. Use-Case Model 5

4. Logical View 5

4.1 Component: Controller 7

4.1 Component: Model 8

4.1 Component: Serivce 10

5. Deployment 10

6. Implementation View 10

# Introduction

## Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

## Scope

## This Software Architecture Document provides an architectural overview of the Comnhalam.com system. Comnhalam.com is being developed by Bitmap-FIT@HCMUS to support online selling food. This document has been generated directly the Comnhalam.com analysis & design model.

## References

[1] Vision document, *the comnhalam.com system.*

[2] Use-case specification – user, *the comnhalam.com system.*

[3] Use-case specification – administrator, *the comnhalam.com system.*

[4] Use-case specification – seller, *the comnhalam.com system.*

[5] Software Architecture document template, *Dr. Nguyen Van Vu* @FIT-HCMUS.

## Overview

This document presents the architecture as a series of views; use case view, logical view, deployment view and implementation view. There is no separate process view described in this document. These are views on underlying diagram created using IntelliJ IDEA system.

# Architectural Goals and Constraints

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

1. The information of user is private. The system must protect it as safe as possible.
2. All of users must provide real information of address, email, phone number.
3. The system must ensure complete protection of data from unauthorized access. All remote accesses are subject to user identification and password control.
4. All performance and loading requirements, as stipulated in the Vision Document, must be taken into consideration as the architecture is being developed.

# Use-Case Model

A close up of a logo

Description automatically generated

# Logical View

A close up of text on a white background

Description automatically generated

*Diagram 1: String MVC Architecture*

## Component: Controller

A screenshot of a cell phone

Description automatically generated

*Diagram 3.1: Controller class diagram*

1. ***AdminController class***

This class would be used to create administrators of the system. It includes some main methods such as banUser, statistic and activeSeller.

1. ***CartController class***

This class would be used to make a cart for a user (buyer). Based on this class, user could pay products added to the cart by purchase method.

1. ***HomeController class***

This class contains some method that helps the website shows some special products to user on the homepage.

1. ***SellerController class***

This class would be used to create users-seller of the system. It includes some main methods such as viewComment, statistic, editProfile and postProduct. These methods help seller can interact with their customers.

1. ***UserController class***

This class is the one of the most important classes of the system. It helps users can do some important activities such as registration, resetting password, verification email, editing profile or posting comments by their methods.

## Component: Model

A close up of a map

Description automatically generated

*Diagram 3.2: Model class diagram*

1. ***User***

This class was created for Users. It contains the features for users such adding a comment or remove a comment. In particular, when a new user comes to the system, all of information of them would be store in this class.

1. ***Comment***

This class would be used when a user comments on or rank a post. The class supports the system to identify the information of the user and do other necessary things to make the activity of user comes true.

1. ***Order***

The class helps to order products they are interested in. Besides, it includes some necessary methods such as removing products out the cart, getting total or listing order details.

1. ***Product***

The class would be used to create a new product for the system. The methods of this class help the system get all of information of a product such price, quantity, name, comments, rank or tags, etc.

1. ***ResetPasswordToken***

The class includes all necessary methods for resetting password features such as setToken, getToken, setUser, getUser,…

1. ***VeritificationEmailToken***

Similar to ResetPasswordToke, the class also contains all of main methods for veritification email feature of user. It includes setToken, getToken or veritificationToken.

1. **Tags**

This class helps the system to divide all products among several particular tags. As a result, all products would have their own tag and base on that, the searching process may be better and faster.

## Component: Service

A close up of a sign

Description automatically generated

*Diagram 3.3: Service class diagram*

This component is the place that makes Controller component and Model component can interact to each other to handle some complex tasks which need lots of activities. There are four classes:

1. AdminService
2. SellerService
3. ProductSerivce
4. UserService

# Deployment view

