Shopee

Software Architecture Document

Version <1.0>

Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 17/12/2023 | 1.0 | Final | Nguyen Huyen Anh  Nguyen Thi Hien Anh |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Introduction 4](#_heading=h.2et92p0)

[1.1 Purpose 4](#_heading=h.z337ya)

[1.2 Scope 4](#_heading=h.3j2qqm3)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_heading=h.1ci93xb)

[1.4 References 4](#_heading=h.3whwml4)

[1.5 Overview 4](#_heading=h.2bn6wsx)

[2. Architectural Representation 4](#_heading=h.qsh70q)

[3. Architectural Goals and Constraints 4](#_heading=h.3as4poj)

[4. Use-Case View 4](#_heading=h.1pxezwc)

[4.1 Use-Case Realizations 5](#_heading=h.49x2ik5)

[5. Logical View 12](#_heading=h.2p2csry)

[5.1 Overview 13](#_heading=h.147n2zr)

[5.2 Architecturally Significant Design Packages 13](#_heading=h.3o7alnk)

[6. Process View 15](#_heading=h.23ckvvd)

[7. Deployment View 15](#_heading=h.ihv636)

[8. Implementation View 15](#_heading=h.32hioqz)

[9. Data View (optional) 15](#_heading=h.1hmsyys)

[10. Size and Performance 15](#_heading=h.41mghml)

[11. Quality 16](#_heading=h.2grqrue)

Software Architecture Document

# Introduction

## Purpose

This document provides a comprehensive architectural overview of the system, using multiple architectural views to depict different aspects of the system. It establishes the overall structure for each architectural view: the decomposition of the view, the grouping of elements, and the interfaces between these major groupings. Therefore, in contrast with the other roles, the software architect's view is one of breadth, as opposed to depth. This document is for all stakeholders on project.

## Scope

This Software Architecture Document will provide an architecture overview of Shopee system which is an e-commerce platform. This has influenced the Analysis and Design Model of Shopee. This document uses the template of Software Architecture.

## Definitions, Acronyms, and Abbreviations

## References

BMP01, version 0.1: Business Architecture Document

RD01, version 1.0: Supplementary Specifications

## Overview

In the rest of the document, this will describe the system from multiple views: Use-Case, Logical, Process, Deployment and Implementation. This also provides the size, performance and the quality.

# Architectural Representation

This document outlines the architecture through various perspectives, including the use case view, logical view, process view, and deployment view. It does not include a distinct implementation view. These perspectives are based on a Unified Modeling Language (UML) such as package diagram, use case diagram and sequence diagram model created using draw.io.

# Architectural Goals and Constraints

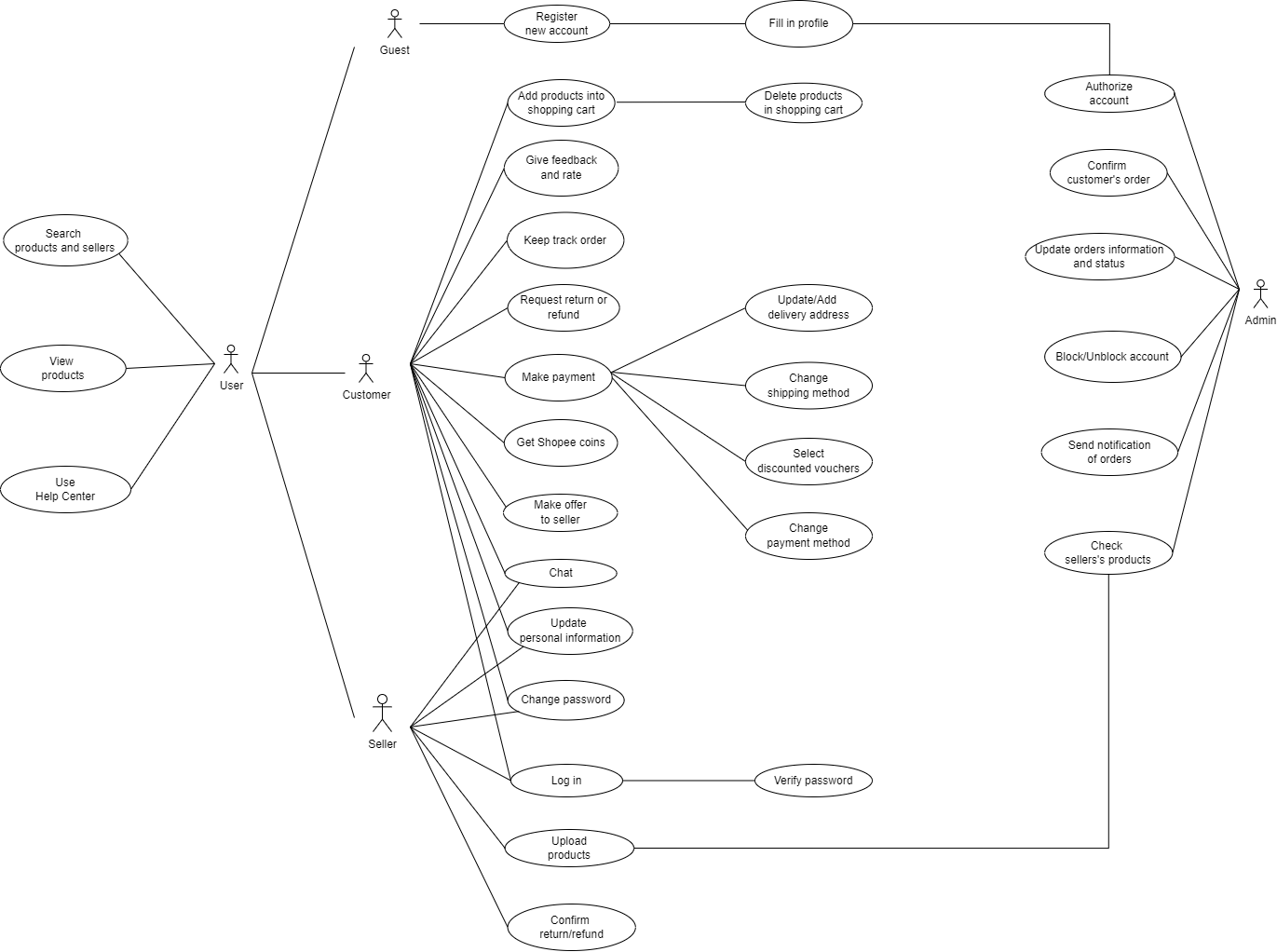
* The existing interface must retrieve all updated trending products and the current product cost based on each event. The Shopee system should support data formats and DBMS.
* All user and administrator functionality must be available well regardless of location.
* Shopee system has to make certain about the comprehensive security of data against unauthorized access. Every access needs to be authorized by user identification and password.
* The system will be structured as a client-server system
* All performance and loading criteria specified in the Supplementary Specification must be considered during the development of the architecture.

# Use-Case View

The Use Case View is an important input to the selection of the set of scenarios and/or use cases that are the focus of an iteration. It describes the set of use cases that represent some significant, central functionality and delicate point of the architecture. The Shopee system significant use cases are (9 use cases):

* Log in
* Register new account
* View product
* Search products and sellers
* Add products into shopping cart
* Make payment
* Upload products
* Confirm orders
* Manage progress of order

These use cases are initiated by the customer, guest, seller and administrator.



## Use-Case Realizations

**Register new account**

This use case allows a Guest to register a new account. When guest browses the product, if they want to save products or purchase products, they need to create an account to continue.

| **Use Case Name** | Register new account |
| --- | --- |
| **Description** | User want to create a new account to using application |
| **Actors** | Guest, Admin |
| **Trigger** | User click on button “Sign up” |
| **Pre-Condition(s)** | User does not have any account using this phone number or email before  User connects to Internet successfully |
| **Post-Condition(s)** | User register an account successfully  User can log in to the application and use other functionalities of application |
| **Basic Flow** | User clicks on “Sign up” button to redirect to Register page.  User enters his/her email address, creates a username, full name, birthdate and chooses a strong password.  User completes additional fields to update his/her profile such as address, gender, phone number and security questions.  User clicks on the “Send Email” button. After that, a confirmation email will be sent to the email address user provided.  User clicks the link in the email to verify your ownership of the email address and activate the account.  System redirect user back to application.  User clicks the "Create Account" or "Sign Up" button to submit data to system. |
| **Exception Flow** | **The user’s email has already used**  The screen will display an error message “This email already exists” to inform that the email can not be used. It will require the user to enter another email to continue.  **User does no click in the link in email to verify**  After 3 minutes when the email was sent, if the user did not confirm, the system would delete the user's registration form. Users will come back to the homepage.  **No response from the user**  If the user does not do anything in 30 seconds, the system will display the notification ‘Session expired’ to require users to rewrite the registration form. |

**Search products and sellers**

This use case describes how a user searches the information that they want to find. The actors starting this use case are Guest, Customer and Seller.

| **Use Case Name** | Search products and sellers |
| --- | --- |
| **Description** | User want to search products or sellers |
| **Actors** | Customer, Guest, Seller |
| **Trigger** | User enter into the search bar and click “Go” |
| **Pre-Condition(s)** | User connects to Internet successfully |
| **Post-Condition(s)** | User get a list of products and sellers that have name matching with the input |
| **Basic Flow** | When the users want to search, they touch the search bar to start, the searching view including the search history and search suggestion appear.  The system shows the search history based on previous searches (if there are), and the suggestion based on both the search history and other aspects such as the trend or users’ behaviors, so that the users can search easier.  Users can search by typing what they want to search on their own, or choose search from search history and search suggestion.  After the user chooses what to search, the system processes the input from the user, displays the results that match the details in users’ input in a results view.  When the users choose a product, they will be directed to the product view or when they choose a seller, they will be directed to the seller’s page. |
| **Exception Flow** | **The input does not match with data in model**  The search page will display a text “Not found”. The page will display others products and sellers that have been accessed the most to suggest for users.  **No response from the user**  If the user does not do anything in 30 seconds, the system will display the notification ‘Session expired’ to require users to rewrite the registration form. |

**View product**

This use case describes how a user views a product in a detailed way with pictures, name, price and description. The actors starting this use case are Guest, Customer and Seller.

| **Use Case Name** | View product |
| --- | --- |
| **Description** | User want to see the detailed information of products |
| **Actors** | Guest, Customer, Seller |
| **Trigger** | User click on the box containing product displayed in interface |
| **Pre-Condition(s)** | User connects to Internet successfully |
| **Post-Condition(s)** | None |
| **Basic Flow** | User clicks on the box containing the product's information(picture, name, price) displayed in the interface.  This redirects users into the product detail page.  This page displays product images, name, description. Users can view various images of the product from different angles or in use, read the product name and description for details like features, materials, and size, check the price, available variations (color, size, etc.), and stock availability.  The product detail page also contains a feedback section where users can read reviews from other buyers to get their opinions on the product.  Page also displays a gray box to inform the seller's rating, shop policies, and location.  In this page, this loads another additional functions such as Add To Cart, Share, Chat With Seller, Select Vouchers. |
| **Exception Flow** | None |

**Log in**

This use case describes how a Customer and Seller log in to the Shopee platform.

| **Use Case Name** | Log in |
| --- | --- |
| **Description** | User logs in to the system |
| **Actors** | Customer, Seller |
| **Trigger** | User wants to log in Shopee platform |
| **Pre-Condition(s)** | Account of users have been created  Account has been authorized by administrator  User connects to Internet successfully |
| **Post-Condition(s)** | User logs in successfully  System save user’s login activity into cookie |
| **Basic Flow** | Users click on the button “Log in”.  Users enter their username/email and password.  System validates this data and confirms this activity.  System saves the user’s login activity into cookie. |
| **Exception Flow** | **Login is not successful**  The system will display the notification “Error!”. It requires users to quit from the login page.  **User choose “Forget Password”**  This will redirect users into the Take Password page. |

**Add products into shopping cart**

This use case allows customers to add their favorite products into a shopping cart. They can change the quantity or type of product, they also can delete products from cart. If customers want to buy one of these products, they will just click on this and button Buy Now. The actors starting this use case is Customer.

| **Use Case Name** | Add products into shopping cart |
| --- | --- |
| **Description** | User adds products into shopping cart to purchase |
| **Actors** | Customer |
| **Trigger** | User click on product detail page |
| **Pre-Condition(s)** | User needs to log in  User connects to Internet successfully |
| **Post-Condition(s)** | Product that users choose must be added into their shopping cart |
| **Basic Flow** | Users view the detail page of the product.  Users click on the button “Add to cart”.  This will display an information box that contains the picture, price, category and quantity of products.  Users can select the type of products they want and its quantity.  The information that users choose will be saved into the Shopping Cart model of users. |
| **Exception Flow** | **One of product’s category is out of stock**  The information box still displays this category. However, this is out of stocks displayed in a gray transparent rectangle, meanwhile others are in black gray rectangle.  **Quantity which users select is over 10**  If the user chooses the number of quantities greater than 10, the system will inform a message “You need to choose quantity under 10”. Therefore, it requires users to select quantity again. |

**Make payment**

This use case allows customers to purchase products. The actors starting this use case is Customer.

| **Use Case Name** | Make payment |
| --- | --- |
| **Description** | User want to purchase products |
| **Actors** | Customer |
| **Trigger** | User click on the button “Buy Now” |
| **Pre-Condition(s)** | User need to log in  User connects to Internet successfully |
| **Post-Condition(s)** | User purchase product successfully  Order will be sent to seller  User should keep track with the order’s progress |
| **Basic Flow** | User click on button “Buy Now”.  Payment page displays personal information of the user (such as phone number, name of consignee, address), the information about products, shipping methods, vouchers, and payment methods.  The user reviews the contents of the order summary to ensure accuracy.  The user reviews the total price, shipping costs, and payment information.  The user confirms the order, and the system processes the payment.  This directs users to a confirmed page. |
| **Exception Flow** | **Insufficient funds**  If your chosen payment method (such as bank account, e-wallet) lacks sufficient funds, the transaction will fail. You'll be notified and prompted to choose another method or add funds.  **Invalid payment information**  Incorrectly entered card details, expired cards, or invalid security codes can cause payment failures. Double-check and re-enter your information carefully.  **Technical issues**  Sometimes, temporary technical glitches with your bank or Shopee's payment system can disrupt transactions. Retry after a short while or try another payment method.  **Voucher expires**  If the choose voucher is expired or run out of stock, it will require users to select another voucher to continue |

**Upload products**

This use case allows a seller to upload publicly his/her products for customer views or buys. The Seller is the actor of the use case.

| **Use Case Name** | Upload products |
| --- | --- |
| **Description** | User want to upload products into Shopee platform to sell |
| **Actors** | Seller |
| **Trigger** | User click on the box containing product displayed in interface |
| **Pre-Condition(s)** | User need to log in  Account of user is authorized as Seller  User connects to Internet successfully |
| **Post-Condition(s)** | Product information will be sent successfully to the admin system to check |
| **Basic Flow** | Users Go to the Seller Centre and click Add New Product.  Users need to Fill up the product name. Three recommended categories will appear automatically, and they can either choose one of them or configure their preferred category manually.  Users Insert the product description and relevant product attributes into the given form. Then, users Insert the price, stock count, and variations (if any) for their product.  Users upload high resolution and professionally-taken photos of your product.  Users will update the shipping settings such as weight, dimension (parcel size) and shipping fee.  Users click button “Save and publish” to complete. |
| **Exception Flow** | **Incomplete details**  Missing essential information like product name, description, category, price, or stock availability can lead to product rejection. Ensure all mandatory fields are filled accurately.  **Invalid pricing**  Prices exceeding allowed limits or using unsupported symbols might trigger rejections. Check Shopee's product pricing guidelines and use the correct format.  **Prohibited content**  Uploading products containing restricted content like illegal items, copyrighted material, or offensive images will result in product removal. Familiarize yourself with Shopee's prohibited product list.  **Image violations**  Using low-quality images, improper sizing, or watermarks exceeding allowed limits can cause rejections. Refer to Shopee's image requirements for optimal product presentation. |

**Confirm orders**

This use case allows a seller to accept or cancel customer’s orders. If the seller cancels the order, the customer’s order is removed. The actor of this use case is Seller.

| **Use Case Name** | Confirm orders |
| --- | --- |
| **Description** | User accepts or denies the order |
| **Actors** | Seller |
| **Trigger** | User click on Seller Centre |
| **Pre-Condition(s)** | User need to log in  Account of user is authorized as Seller  User connects to Internet successfully |
| **Post-Condition(s)** | Order’s status will be updated successfully to customers and administrator |
| **Basic Flow** | Users Go to Seller Centre and click on My Shipment. This page will display all user’s orders in each status: unpaid, to ship, shipping, completed, cancellation and return/refund.  Users click on Mass Ship. Users can see a list of orders that have not been confirmed.  Users Click on Generate Documents to modify these pages.  Users select the orders that they want to accept. |
| **Exception Flow** | **System overload**  During peak periods or due to technical glitches, Shopee's order confirmation system might experience temporary overload, leading to delays or errors. Be patient and try again later, or contact Shopee customer support for assistance.  **App or website errors**  Bugs or glitches in the Shopee app or website can sometimes cause unexpected issues during order confirmation. Try refreshing the page, restarting the app, or updating your software.  **Invalid order**  If the order did not meet the requirements that users have set, the system will automatically cancel this order. The system will send to customers a notification about that. |

**Manage order’s progress**

This use case allows admin to control and update status of orders for both customer and seller. The Admin is the actor of the use case.

| **Use Case Name** | Manage order’s progress |
| --- | --- |
| **Description** | User want to manage the progress of the process purchasing products |
| **Actors** | Admin |
| **Trigger** | User access to manage progress |
| **Pre-Condition(s)** | User need to log in  Account of user is authorized as Administrator  User connects to Internet successfully |
| **Post-Condition(s)** | Order’s progress should be updated successfully to both Customer and Seller |
| **Basic Flow** | Users access Manage Order center to see all information from others Users send into that.  Users use method filter to get only information of each order that needs to be managed.  Users update into the Manage Order page.  The system will send that data to other users. |
| **Exception Flow** | **Lost or damaged packages**  Carriers can lose or damage packages, requiring investigation and potentially re-shipment.  **Weather delays**  Severe weather conditions or natural disasters can disrupt shipping schedules.  **Customs delays**  International shipments might face delays due to customs clearance procedures.  **Vehicle breakdowns or accidents**  Carrier issues like breakdowns or accidents can cause delays or rerouting.  **Security concerns**  Security threats or suspicious activity can lead to shipment holdouts or inspections.  **Incorrect items**  Customer receives wrong items or an incomplete order.  **Missing documents**  Required documents like invoices or certificates missing from the shipment. |

# Logical View

This part is a description of the logical view of architecture. Describes the most important classes, their organization in service packages and subsystems, and the organization of these subsystems into layers. Also describes the most important use-case realizations, for example, the dynamic aspects of the architecture. Class diagrams may be included to illustrate the relationships between architecturally significant classes, subsystems and packages.

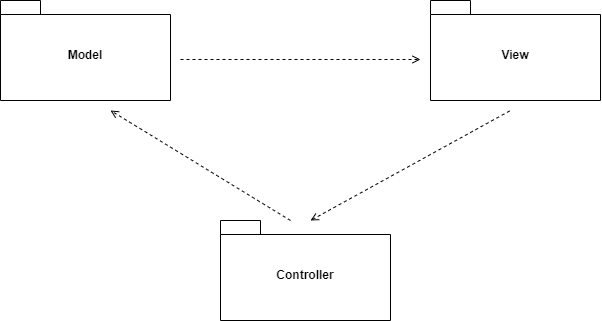
The logical view of the course registration system is comprised of the 3 main packages:

The View Package contains classes for handling the interface layouts based on the data received from the model.

The Model Package includes entity classes for defining what data the application should contain.

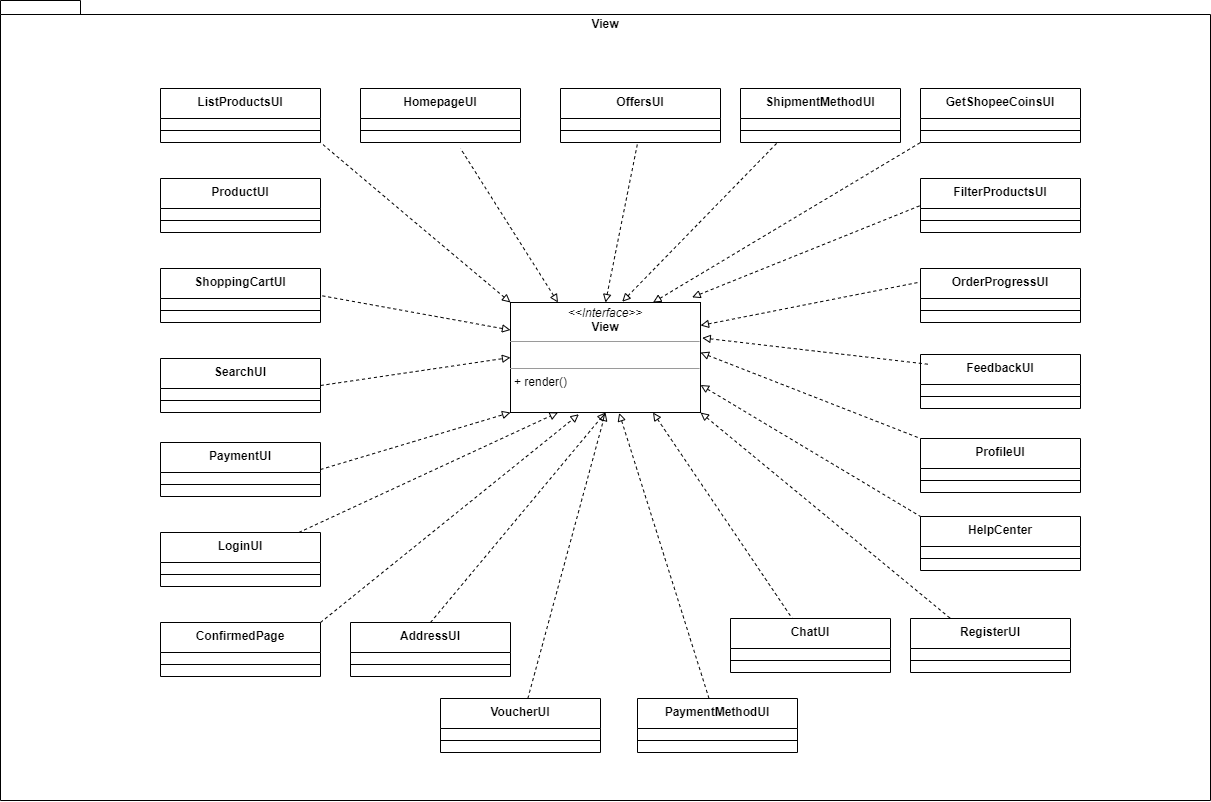
The Controller Package contains classes for updating the model and viewing in response to input from the users of the application.

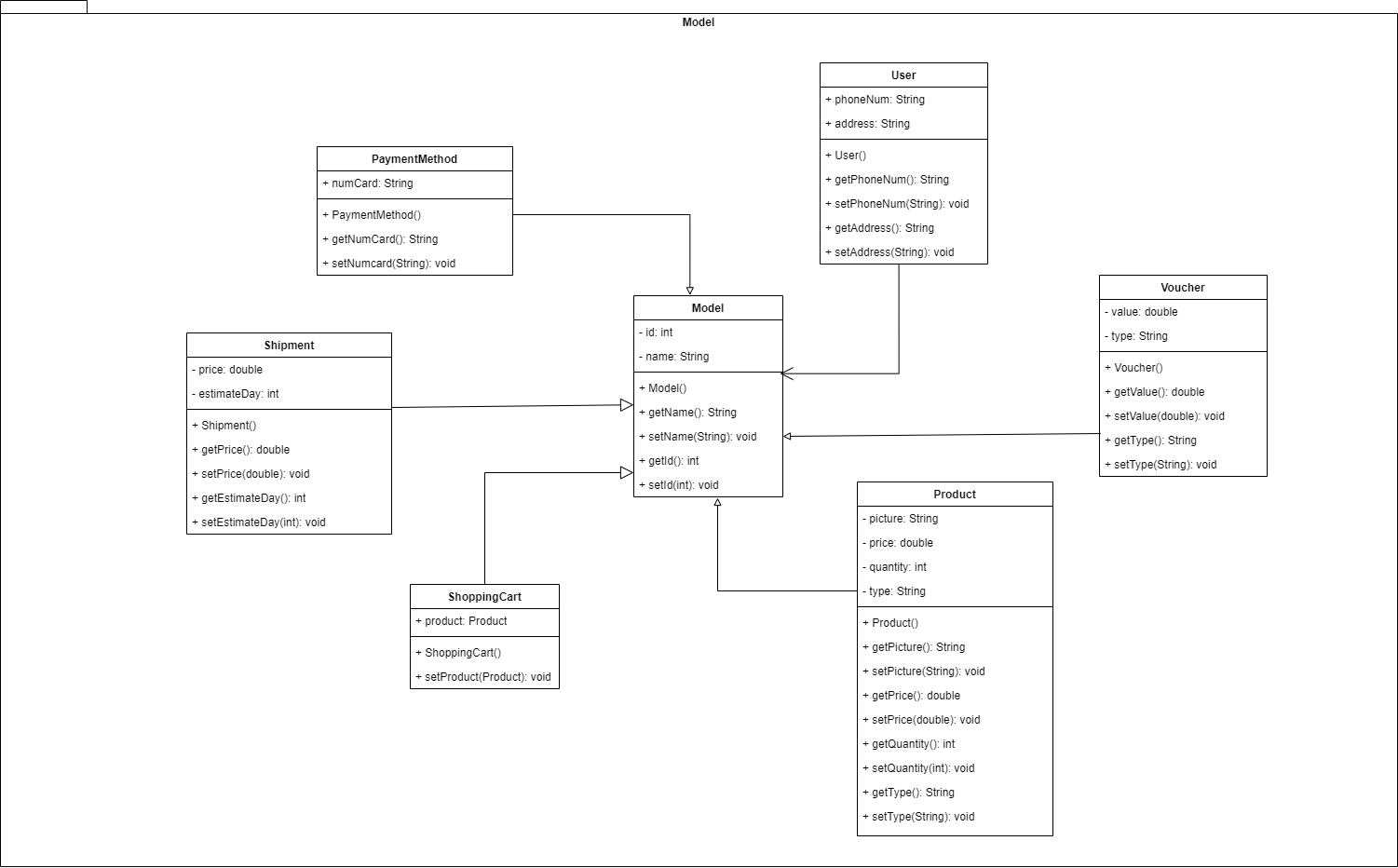
## Overview



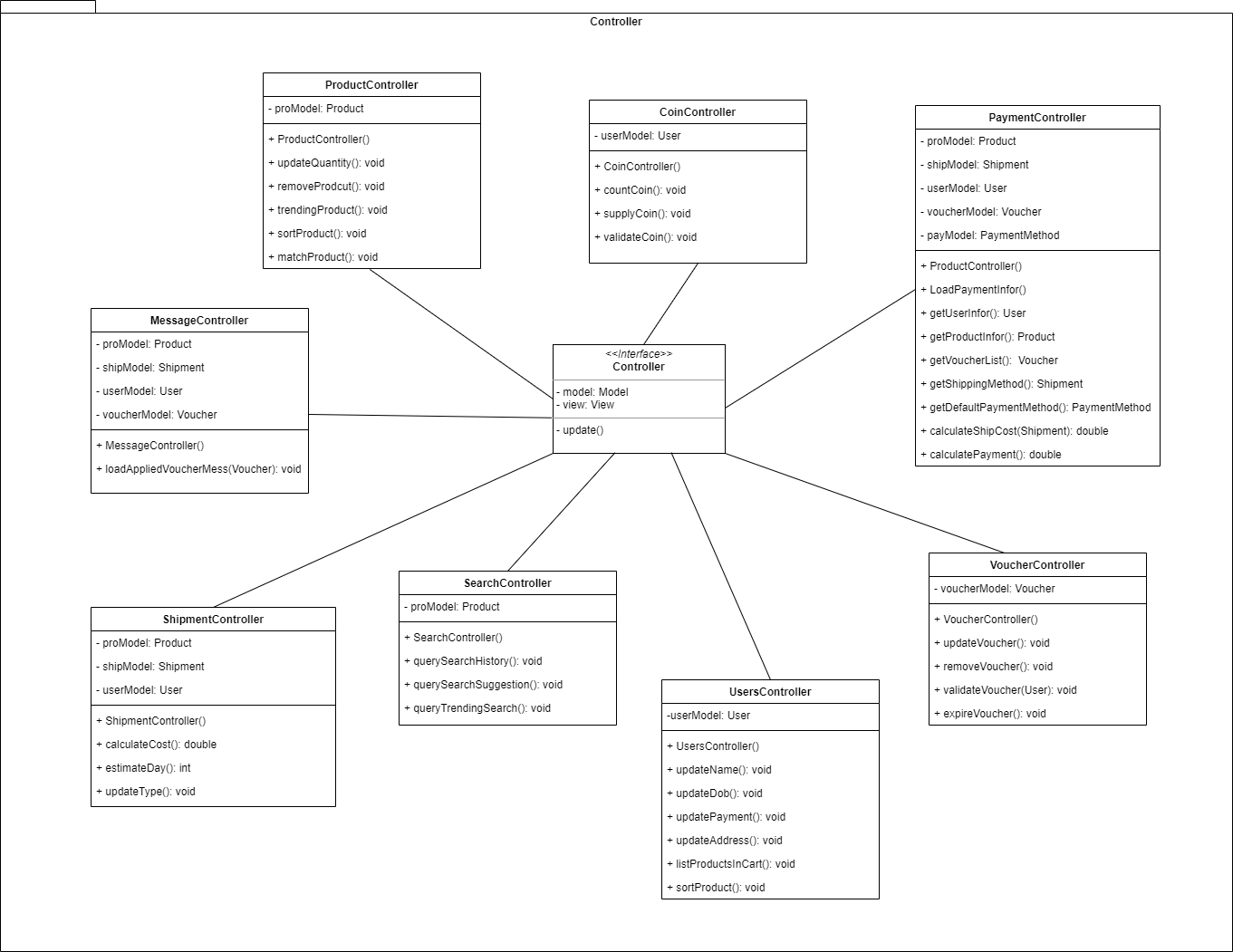
## Architecturally Significant Design Packages

**Package View**



**Package Model**

**Package Controller**

****

# Process View

# Deployment View

At its core, the Shopee System resides on a single server cluster. Think of it as a secure hub with dedicated rooms for different tasks. One room, the web server, handles all incoming website requests. Another, the database server, stores and retrieves all your valuable training data. And for extra peace of mind, a backup database server acts as a vigilant security guard, keeping a copy of everything just in case. To power the web application itself, we use a dedicated NodeJs server, while two trusty MySQL databases handle all the data magic – one for everyday use and the other for safekeeping. To keep data more secure, using Big Data is the better choice. To access the system, all we need is an internet connection, making it easily reachable from anywhere at any time. This is just a simplified overview of the Shopee setup, giving a clear picture of how it all works together.

# Implementation View

This document is based on design, therefore, this does not have an implementation view.

# Data View (optional)

# Size and Performance

* The system should support up to 5000 simultaneous users at any given time.
* The system shall provide access to the legacy course catalog database with no more than a 10 second latency.
* The system must be able to complete 80% of all transactions within 2 minutes.
* The client portion must require less than 20 MB disk space and 32 MB RAM.

# Quality

* System should be user-friendly and always accessible
* The registration system's interface will be intuitive and straightforward, tailored for users comfortable with computers. No extra training is needed – navigating the system feels natural.
* Built-in online e-commerce shopping comes with every feature, offering clear step-by-step instructions and helpful definitions for unfamiliar terms and acronyms. It's like having a friendly guide within the system itself.
* Register anytime, anywhere. The system is available 24/7, with minimal downtime of only 2% annually.
* Downloading updates for the user software is a breeze with online features, no need for installation hassles.
* This updated phrasing emphasizes the user-friendly design, seamless accessibility, and convenient upgrade system, making the Shopee System feel modern and reliable.