

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
School of Information and Communication Technology

Software Design Description

Version: final

eStoreManager

Course: ITSS Software Development

Group 1

Nguyễn Việt Anh

Nguyễn Sỹ An

Theo Mercurio

Hanoi, December 10th, 2018

Index

1	Introduction.....	3
1.1	Objective.....	3
1.2	Scope.....	3
1.3	Glossary.....	4
1.4	References.....	4
2	Architecture Design.....	5
2.1	Software Architecture.....	5
2.1.1	Main design.....	5
2.1.2	Server side.....	7
2.1.3	Client side.....	8
2.2	Overall Design.....	9
2.3	Package Detail Design.....	11
2.4	Interaction Diagram.....	12
2.4.1	Interaction diagram – Add bill.....	12
2.4.2	Interaction diagram – Return product.....	13
2.4.3	Interaction diagram – Import product.....	14
2.4.4	Interaction diagram – Change password.....	15
2.4.5	Interaction diagram – Manage Supplier.....	15
2.4.6	Interaction diagram – Copy Bill.....	16
2.4.7	Interaction diagram – Delete Bill.....	16
2.4.8	Interaction diagram – Log in.....	17
3	Interface Design.....	18
3.1	User Interface.....	18
3.1.1	Screen Transition.....	18
3.1.2	Screen design and screen specification.....	18
4	Class Design.....	31
4.1	Class Diagram.....	31
5	Data Model.....	32
5.1	Entity Relation Diagram.....	32
5.2	Logical Data Model.....	33
5.3	Detailed Design.....	33
5.3.1	Users.....	33
5.3.2	Roles.....	34
5.3.3	User_roles.....	34

5.3.4 Customers.....	34
5.3.5 Products.....	35
5.3.6 Suppliers.....	35
5.3.7 Buys.....	35
5.3.8 Buy_items.....	35
5.3.9 Sells.....	36
5.3.10 Sell_items.....	36
5.3.11 Invoices.....	36

1 Introduction

1.1 Objective

This document presents the detailed description for eStore Manager – a store management system user group and their usable function at run time. This document also describes the objectives and features of the system, interfaces and constraints of the system in response to external action.

This document is for stakeholders and related software developers.

We are trying to implement a desktop application, which:

- + Have multiple language. Firstly we support English and French.
- + Have clean structure for ease in development and maintainent.
- + *Have a good UI design using EJS Templating.*

1.2 Scope

In reality, any software needs to have the function of managing users or group of users, and dynamically granting permissions for usage of certain functions in the system.

This software will be built as a management tool for a small store, with only 1 branch.

The software's goal includes creating following components:

- Interface to make bills and sell products.
- Interface to manage suppliers, import products.
- Interface to manage employees.
- Interface to manage customer list.
- Interface to manage store finance.

Each function may be utilised by many roles of users in: Admin, cashier, warehouse manager.

1.3 Glossary

- **Admin:** administrator in this software often is the store owner. This user can interact with the system in full permission.
- **Cashier:** This user is a role of shop keeper: make bills and sell products. Cashier also can have access to history of bills, remove a bill in the past (to modify some information in the bill).
- **Warehouse:** is a place where goods may be stored before their export or distribution for sale.
- **Warehouse manager:** is a user who can manage suppliers, add import bills and be responsible to manage goods in warehouse.
- **Supplier:** a vendor who provide product for the store at whole sale price. The store imports goods there to sell to their customers.

1.4 References

An online store management system: Kiot Viet: <https://kiotviet.vn>

2 Architecture Design

2.1 Software Architecture

2.1.1 Main design

For this project, we use passive Model-View-Controller (MVC) as our design pattern for both client-side and server-side (See detail in 2.1.2 and 2.1.3).

In passive MVC, model is inactive. It is not in action. It does not notify View when it is changes by Controller. The notification task is done by Controller. That's the major difference to the classic MVC.

- * View: View represents the user interface. The data it shows, gets form Model Component. View gets updated notification from Controller when it updates the Model.

- * Controller: Controller acts as intermediate between View and Model. When user fires an event, then View receives it and handover the control to Controller. Then Controller updates the Model and notifies View about update.

- * Model: Model contains data for the Application.

For communicating between client-side and server-side, we build a RESTful Web Service.

Resource Oriented Architecture (ROA) is used as the architecture for building RESTful Web Service.

ROA is an abbreviation of “Resource Oriented Architecture” and defines the basic architecture for building a Web Service in accordance with REST architecture style (rules).

It is important to thoroughly understand ROA architecture when creating RESTful Web Service.

This section explains following 7 elements of ROA architecture.

These form important architectural elements for building RESTful Web Service. However, it is not always necessary to apply all of these elements.

Necessary elements should be applied after considering the characteristics of the application to be developed.

Following five architectural elements must be applied regardless of the application characteristics.

Sr.

No. Architecture

Architecture overview

- | | |
|---|---|
| (1) Publishing as a resource on Web | It is published as a Web resource through which information stored in the system is provided to the client. |
| (2) Identifying the resource URI (Universal Resource Identifier) that can uniquely identify | |

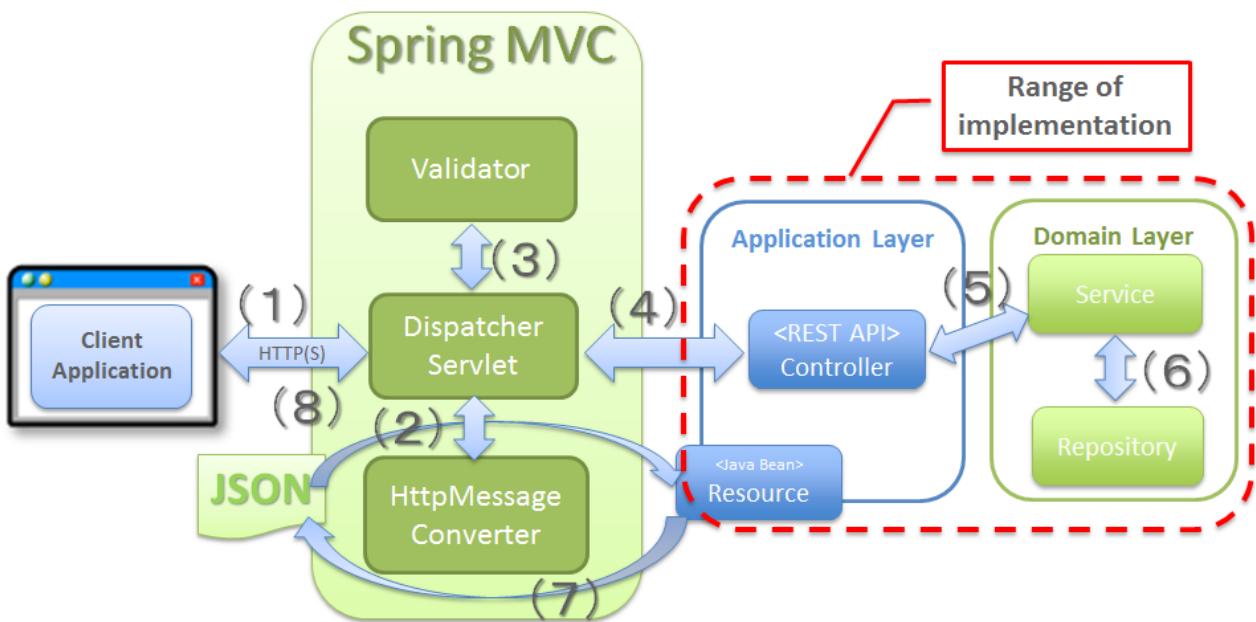
Sr.	No. Architecture	Architecture overview
	using URI	a Web resource is assigned to the resource published to the client.
(3)	Resource operations using HTTP methods	Resource related operations are implemented by using different HTTP methods (GET, POST, PUT and DELETE).
(4)	Using an appropriate format	JSON or XML that represents the data structure, is used as resource format.
(5)	Using the appropriate HTTP status code	Appropriate HTTP status code is set in the response returned to the client.

Following two architectural elements are applied depending on the characteristics of an application.

Sr.	No. Architecture	Architectural elements
(6)	Stateless communication between client and server	This element enables to perform the process only by the information requested from client, without retaining the application status on the server.
(7)	Link to related resource	It includes links to other resources (URI) inside a resource that are related to the specified resource.

2.1.2 Server side

When RESTful Web Service is developed using Spring MVC, the application is configured as given below. Among these, implementation is necessary for the portion marked with red frame



(1) : Spring MVC receives a request from client and determines the REST API (handler method of Controller) to be called.

(2): Spring MVC converts the JSON format message specified in request BODY to Resource object by using `HttpMessageConverter`.

(3): Spring MVC performs input validation for the value stored in Resource object using `Validator`.

(4): Spring MVC calls REST API.

Here, the Resource that has been converted from JSON and for which input validation is carried out, is delivered to REST API.

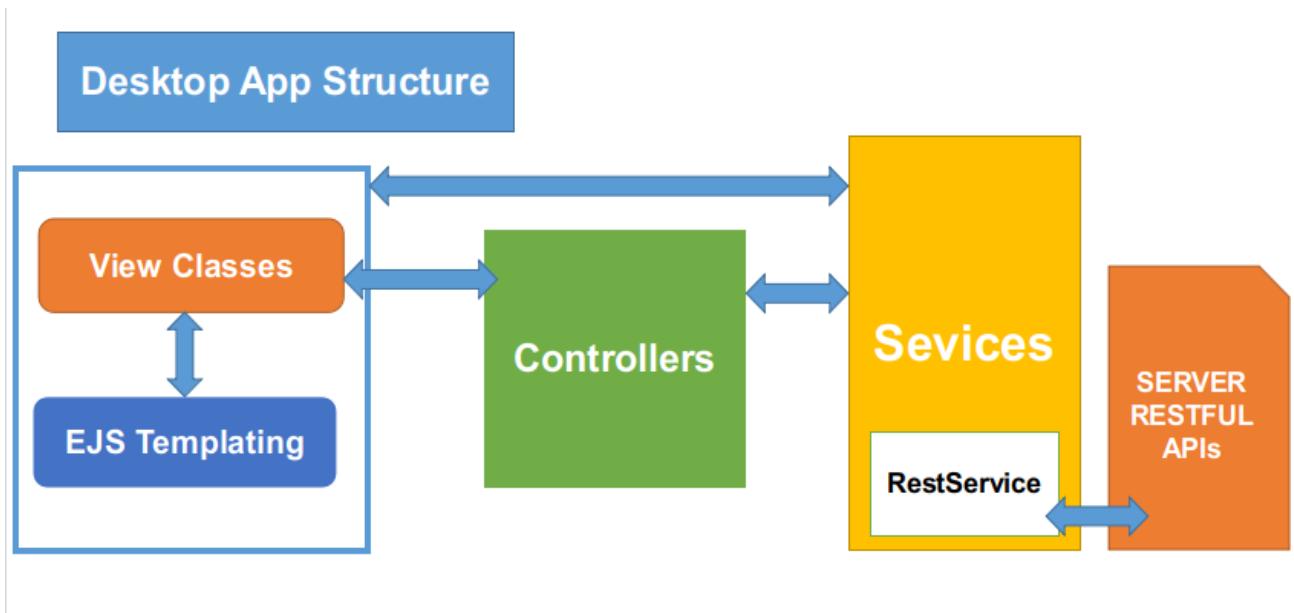
(5): REST API calls Service method and performs the process for DomainObject such as Entity etc.

(6): Service method calls the Repository method and performs CRUD process for the DomainObject such as Entity etc.

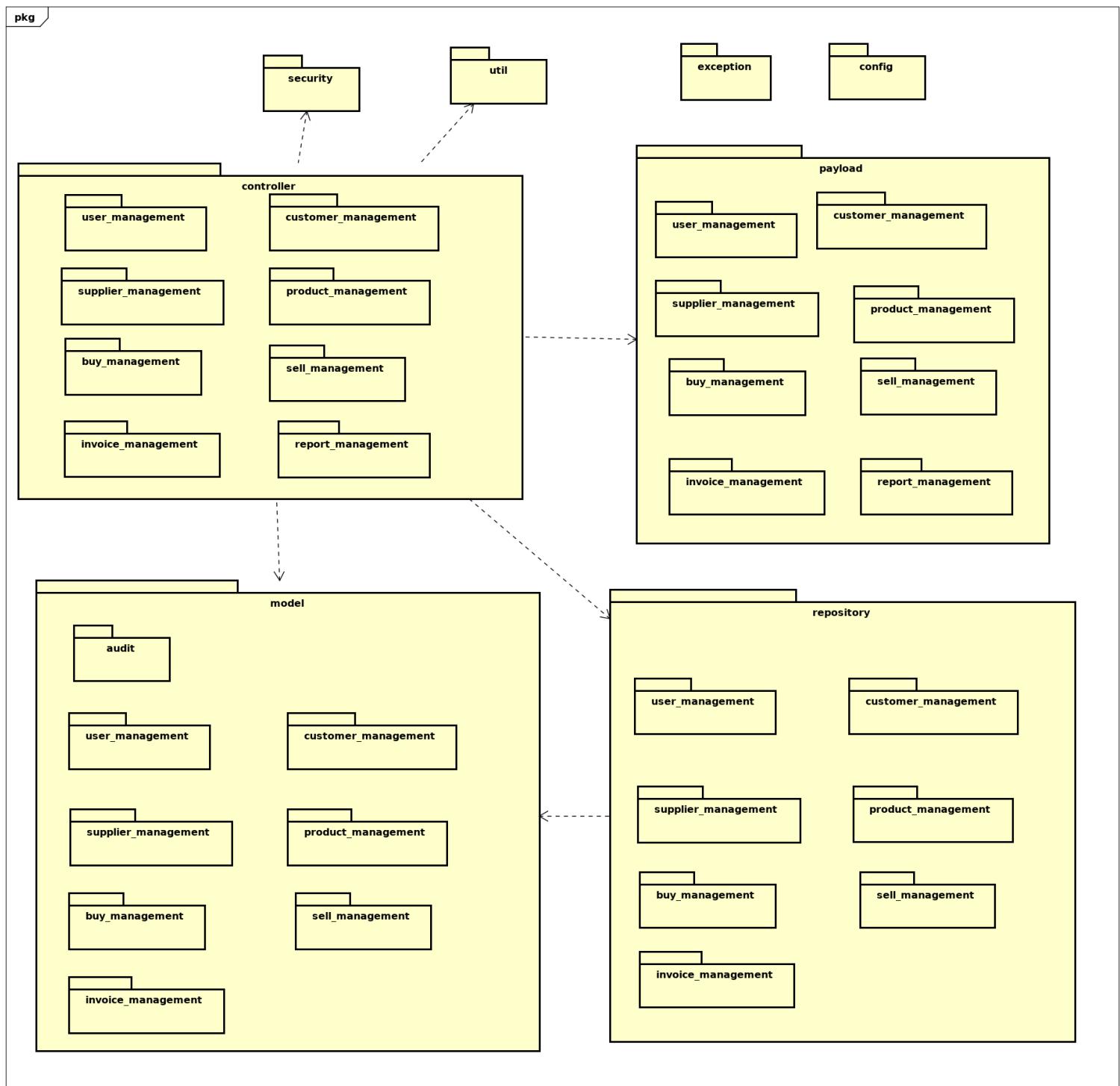
(7): Spring MVC converts the Resource object returned from REST API to JSON format message, by using `HttpMessageConverter`.

(8): Spring MVC sets JSON format message in response BODY and responds to client.

2.1.3 Client side



2.2 Overall Design



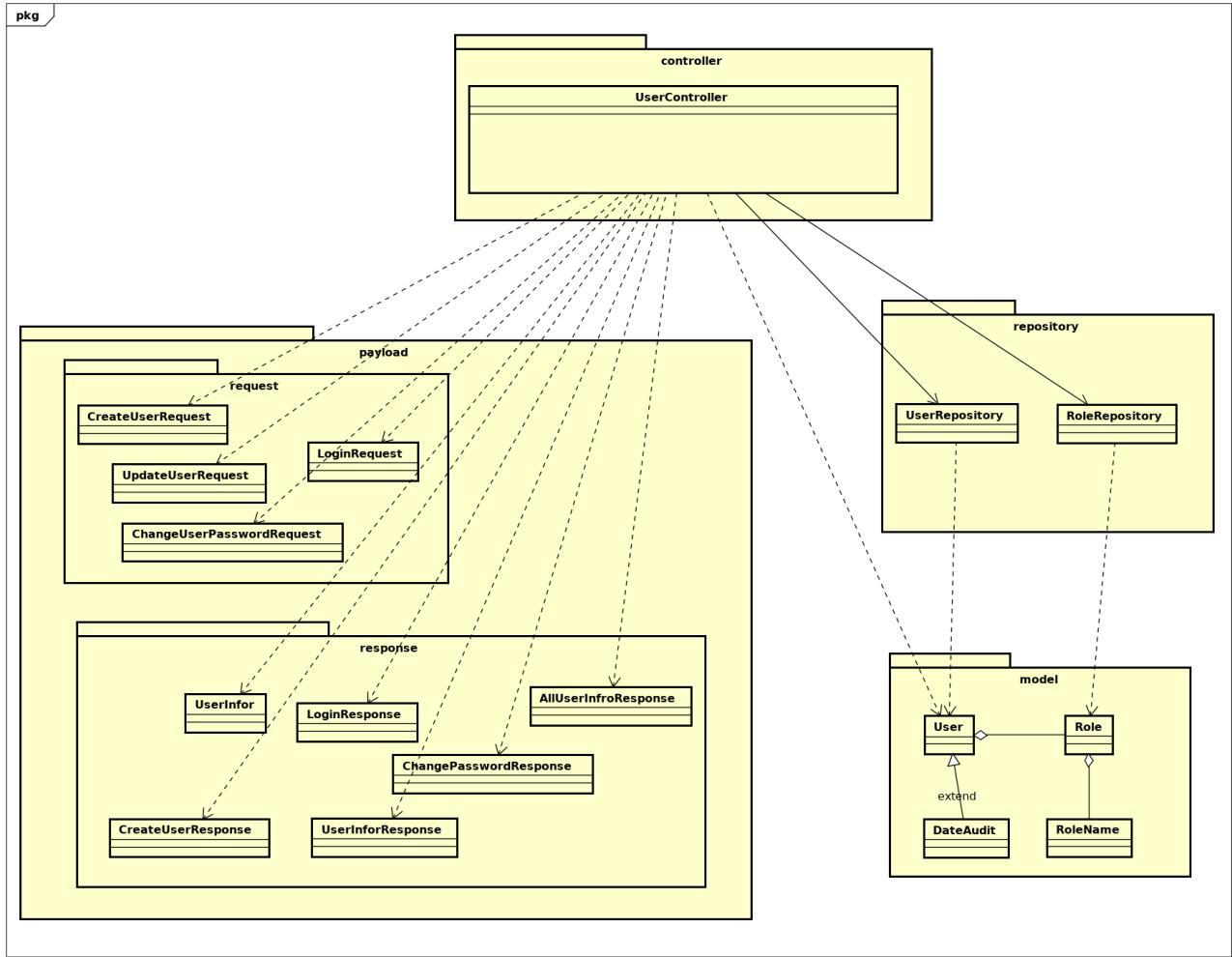
SERVER OVERALL PACKAGE DESIGN UML

Views	Controllers
<ul style="list-style-type: none"> - about <ul style="list-style-type: none"> + AboutView - barcode_scanner <ul style="list-style-type: none"> + BarcodeScannerView - cashier <ul style="list-style-type: none"> + CashierView - customers <ul style="list-style-type: none"> + CustomerView + AddCustomerView + EditCustomerView - employees <ul style="list-style-type: none"> + EmployeeView + AddEmployeeView + EditEmployeeView - import_products <ul style="list-style-type: none"> + ImportProductView + AddImportView + ViewImportView + AddImportItemView 	<ul style="list-style-type: none"> - shared <ul style="list-style-type: none"> + View - suppliers <ul style="list-style-type: none"> + SupplierView + AddSupplierView + EditSupplierView - login <ul style="list-style-type: none"> + LoginView - password_input <ul style="list-style-type: none"> + PasswordInputView - preferences <ul style="list-style-type: none"> + PreferenceView - products <ul style="list-style-type: none"> + ProductView + AddProductView + EditProductView - reports <ul style="list-style-type: none"> + ReportView - sell_bills <ul style="list-style-type: none"> + SellBillView + ViewSellBillView

CLIENT OVERALL PACKAGE DESIGN UML

- In the **Views** package, we have 16 subpackages, each package contains at least 1 view.
 - + **shared/View** is the class of a common view. Other views extend this View and add more logic handlers.
- In the **Controllers** package, we have some **Controllers** controlling the logic of the whole application and communicate with server via **Services/RestService**.
- In the **Services** package we have:
 - + **ConfigGetter** to save and provide some global configuration.
 - + **Dialog**: provide Dialog UI service
 - + **EventGetter**: provide the event by the event ID. Because this application is mainly based on events, this service plays an important role.
 - + **RestService**: provide service to communicate with the server.
 - + **TextGetter**: Provide the language service. This application uses this class to implement multiple language.

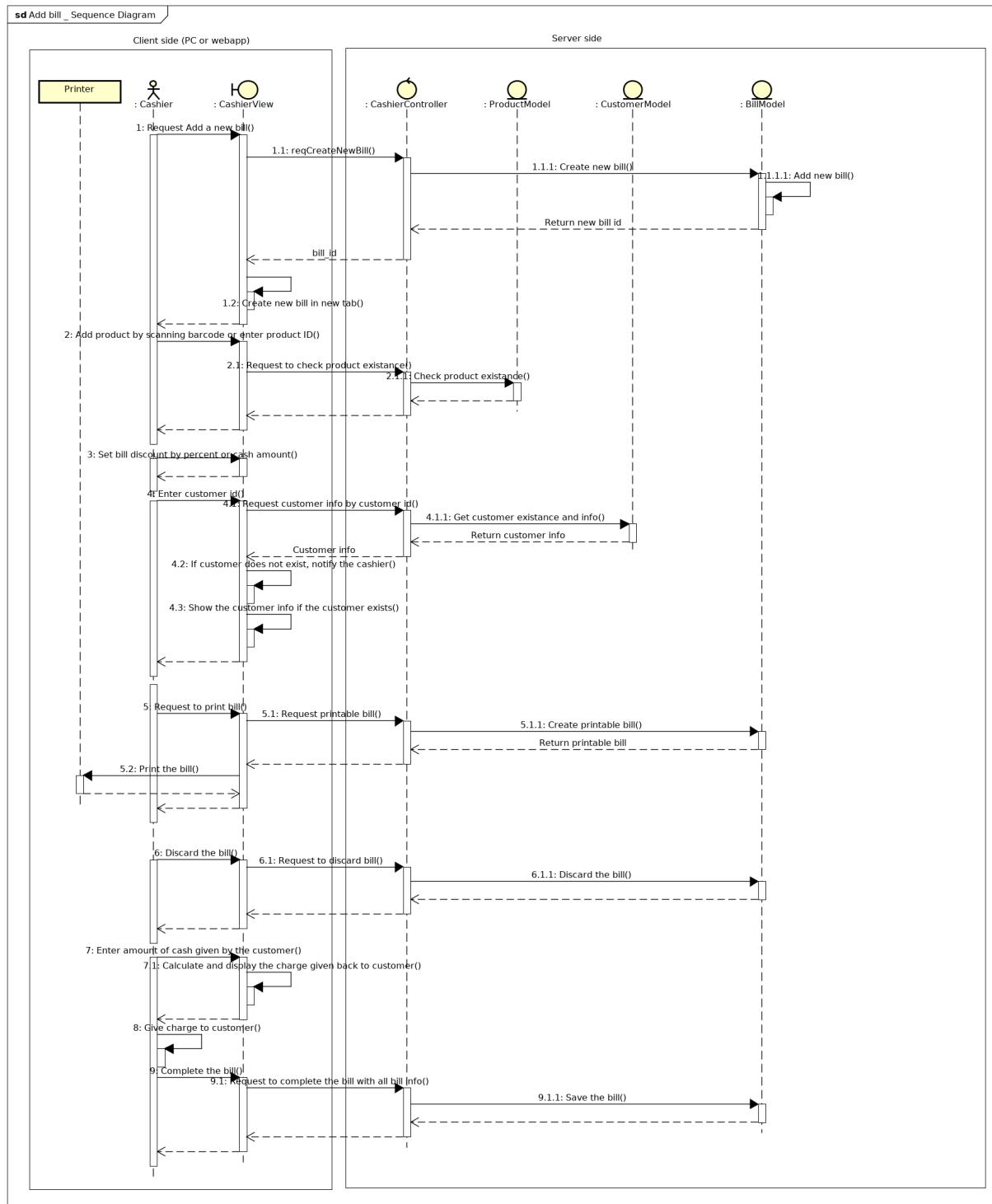
2.3 Package Detail Design



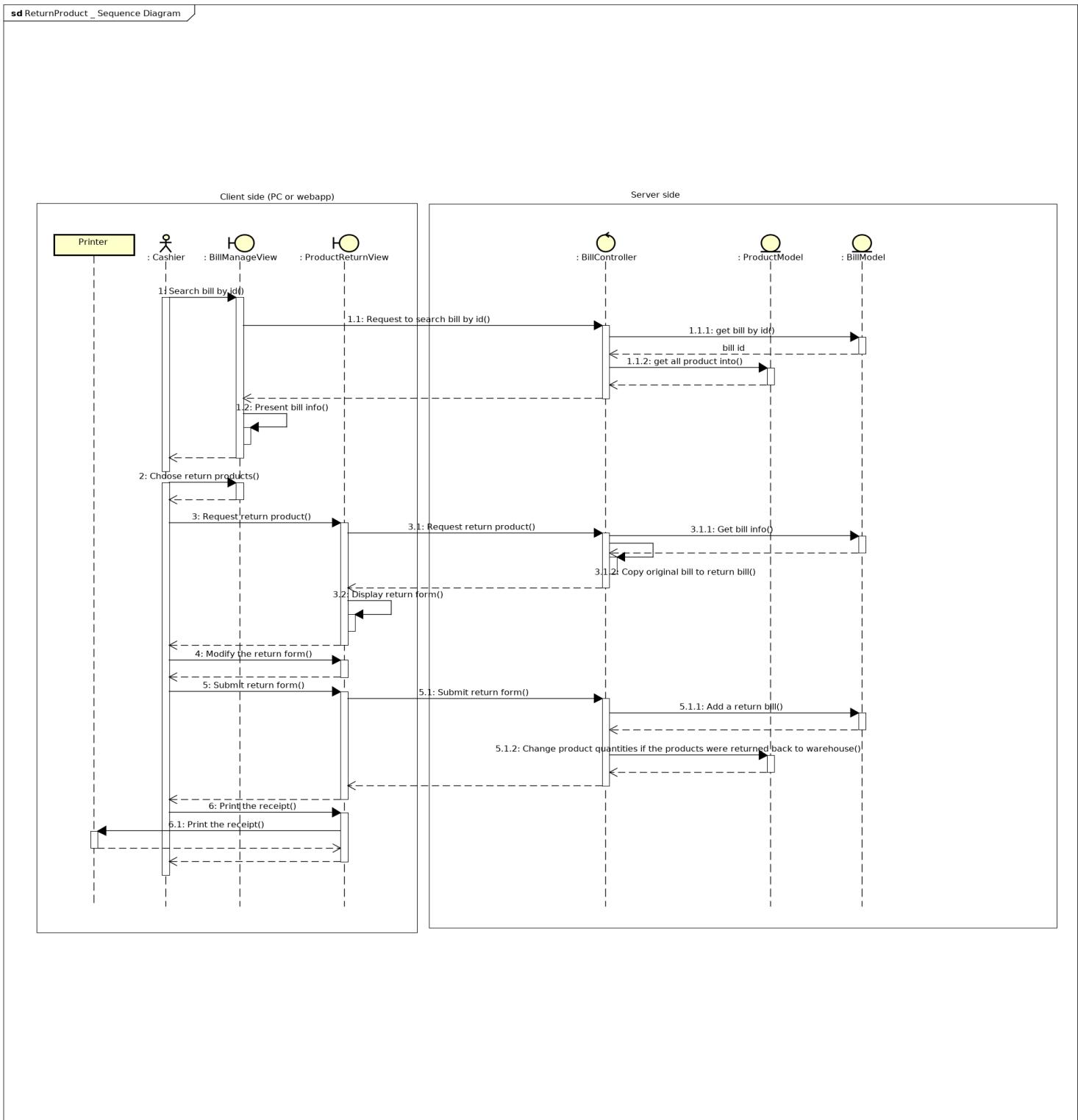
USER MANAGER PACKAGE DESIGN UML

2.4 Interaction Diagram

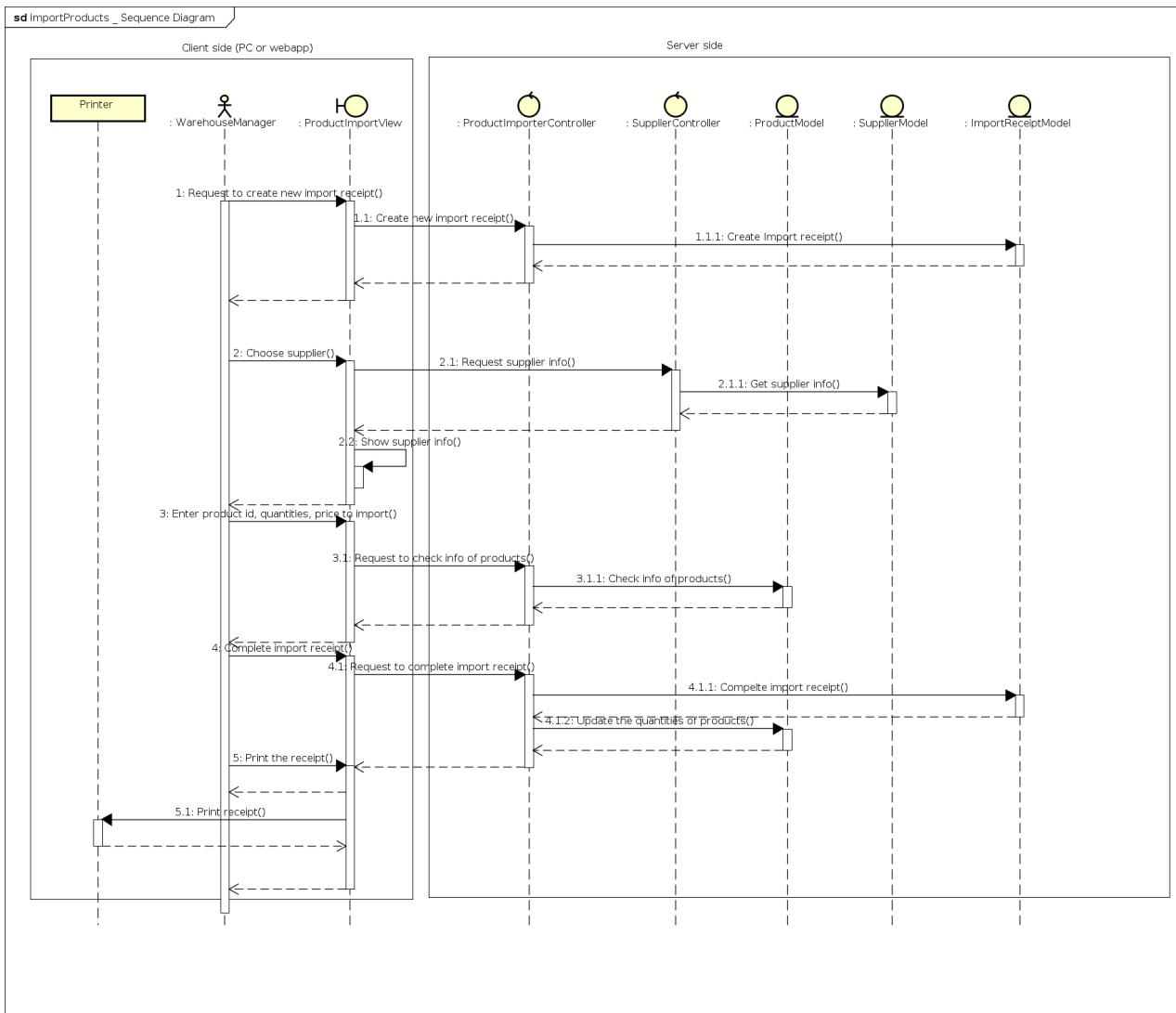
2.4.1 Interaction diagram – Add bill



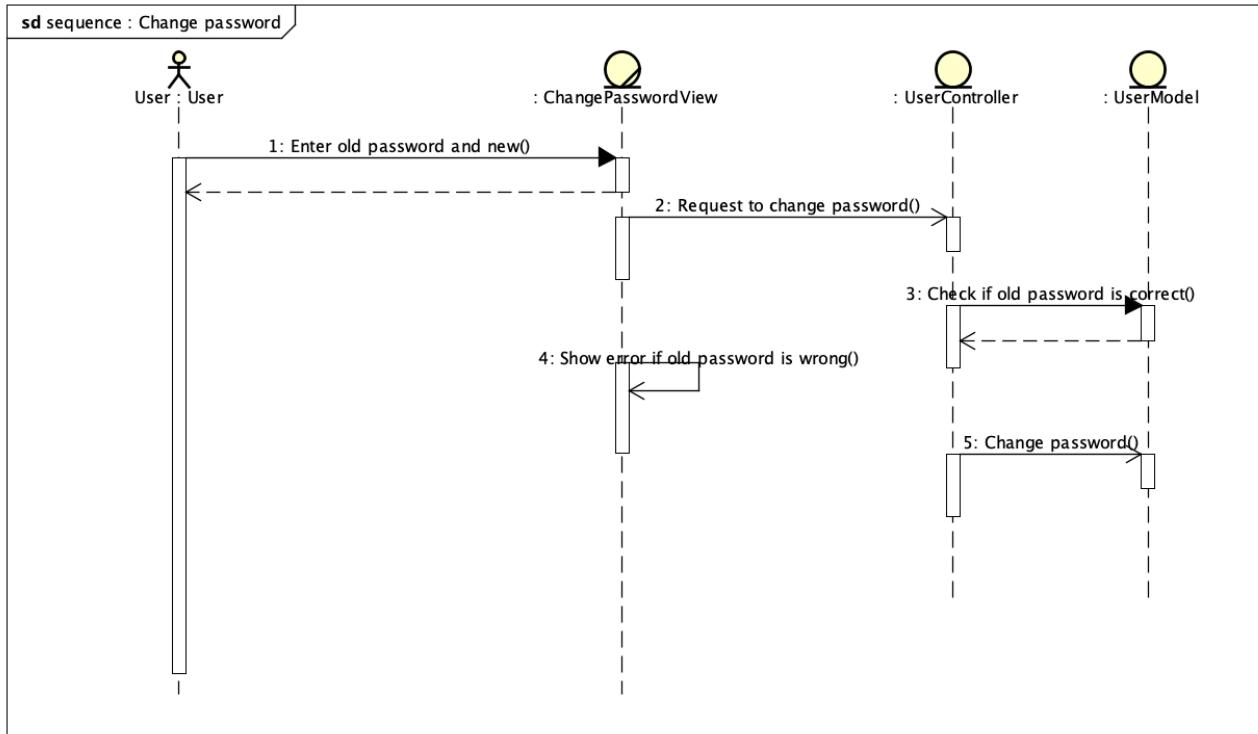
2.4.2 Interaction diagram – Return product



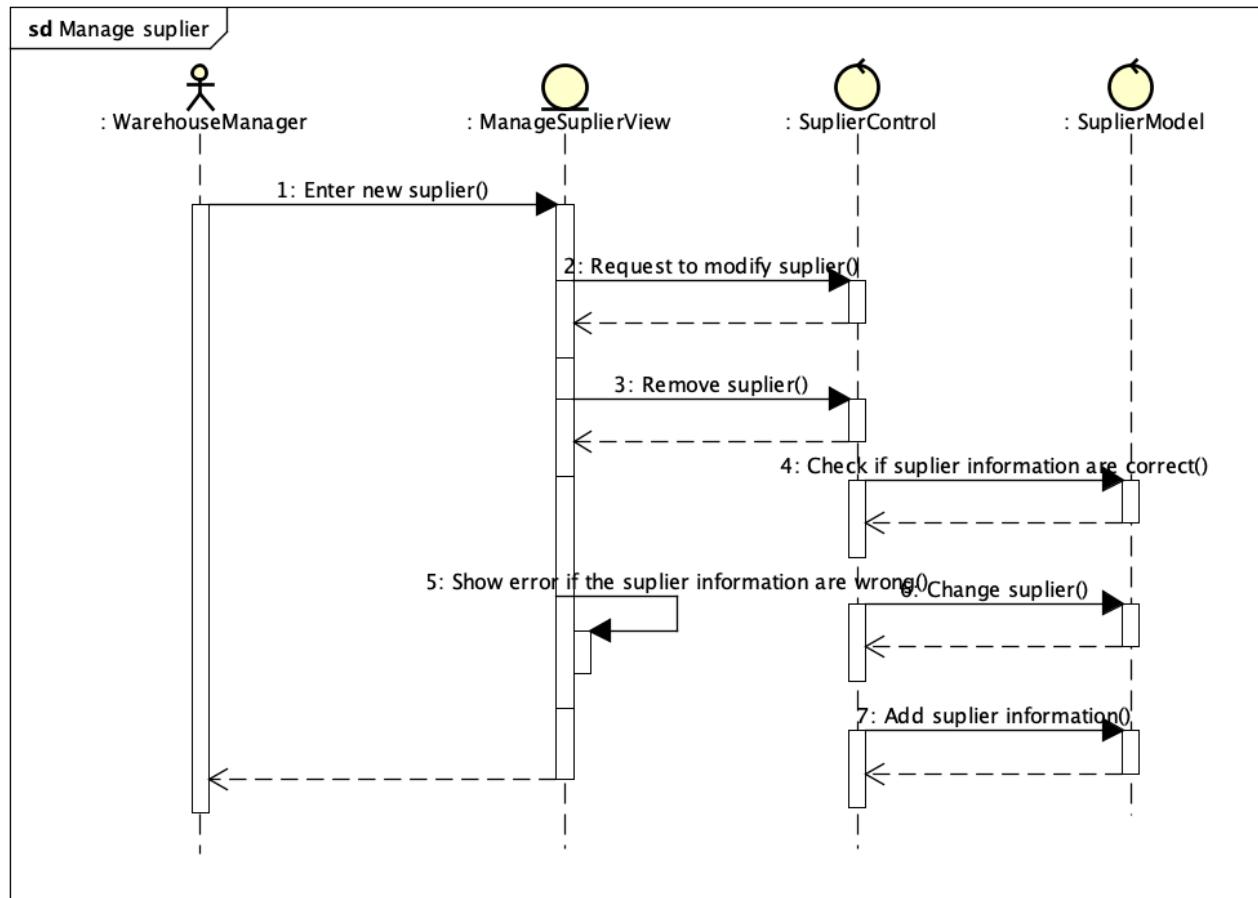
2.4.3 Interaction diagram – Import product



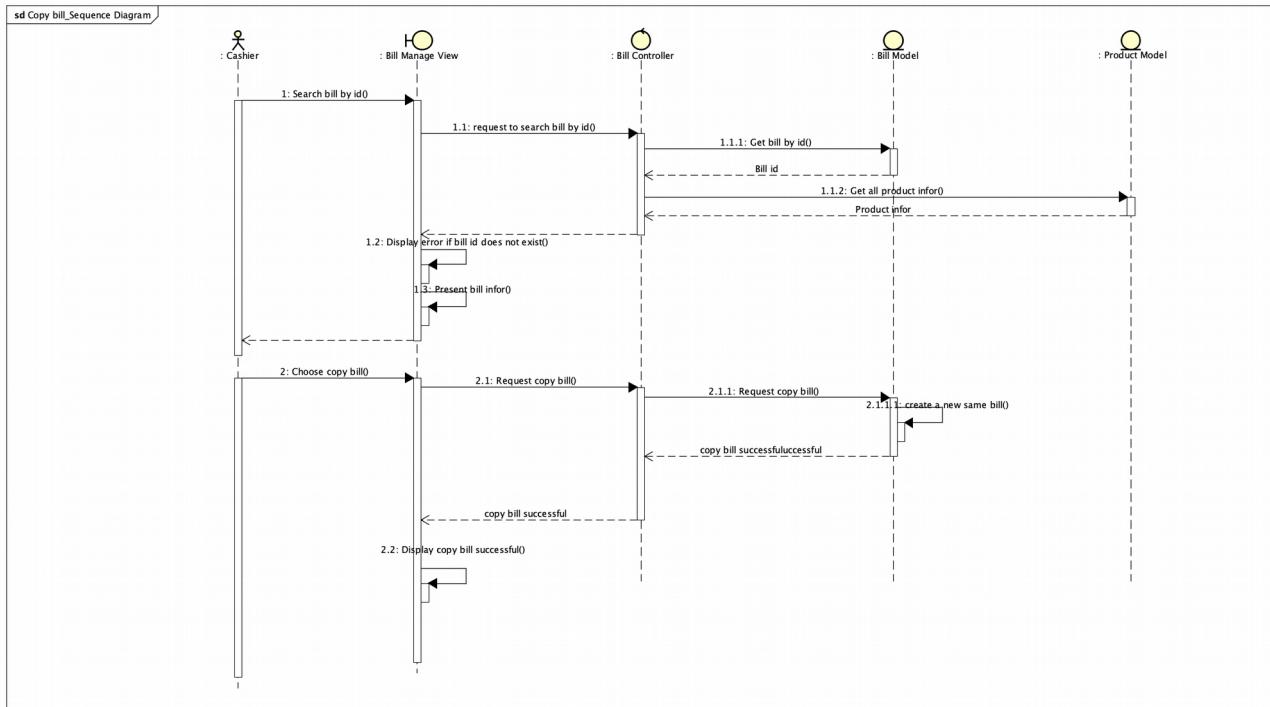
2.4.4 Interaction diagram – Change password



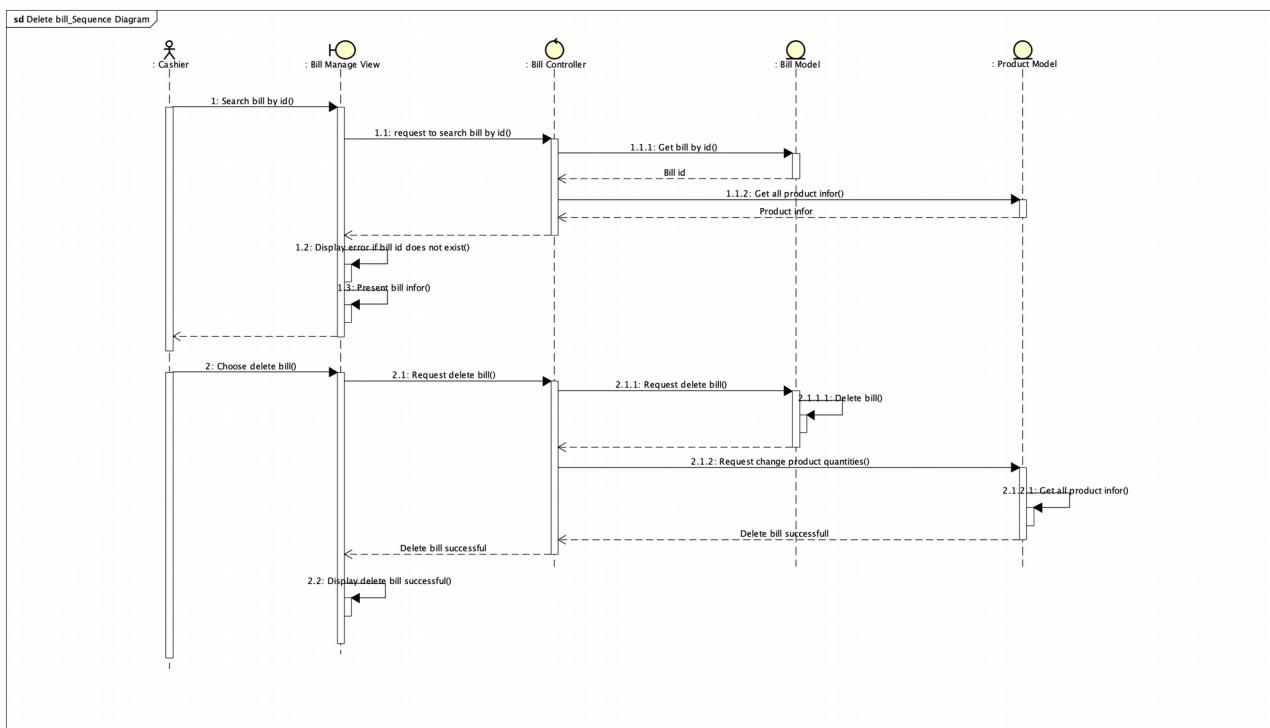
2.4.5 Interaction diagram – Manage Supplier



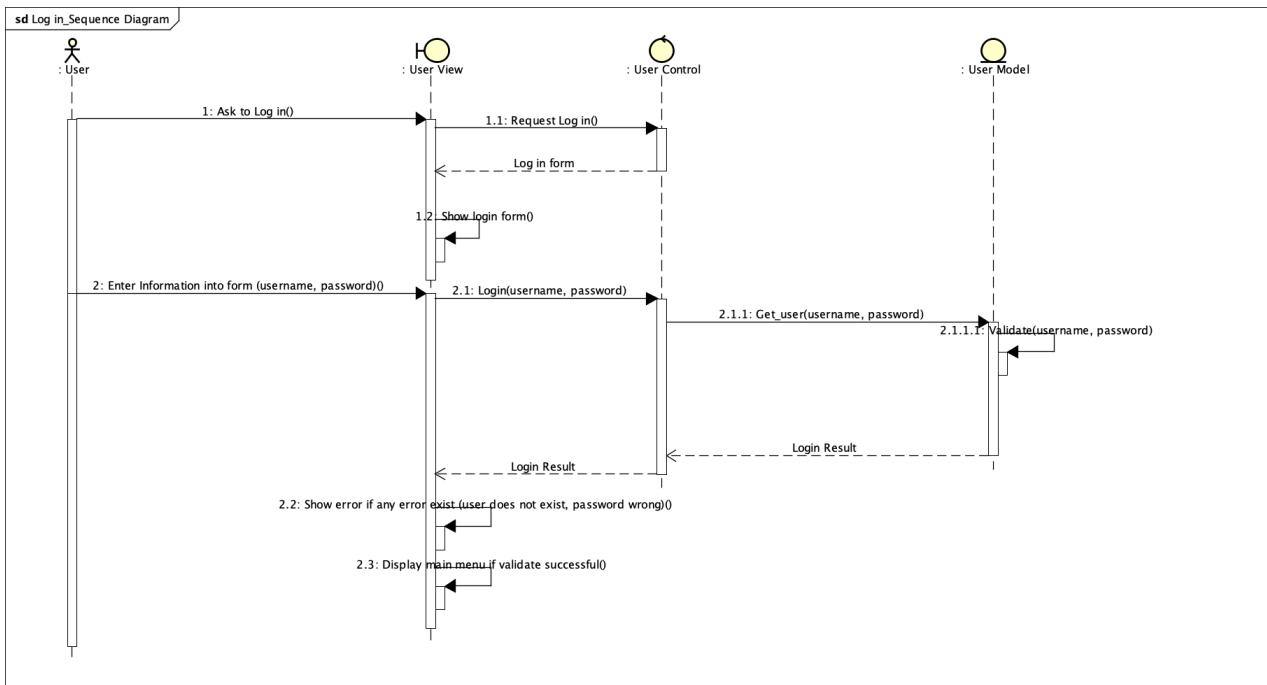
2.4.6 Interaction diagram – Copy Bill



2.4.7 Interaction diagram – Delete Bill



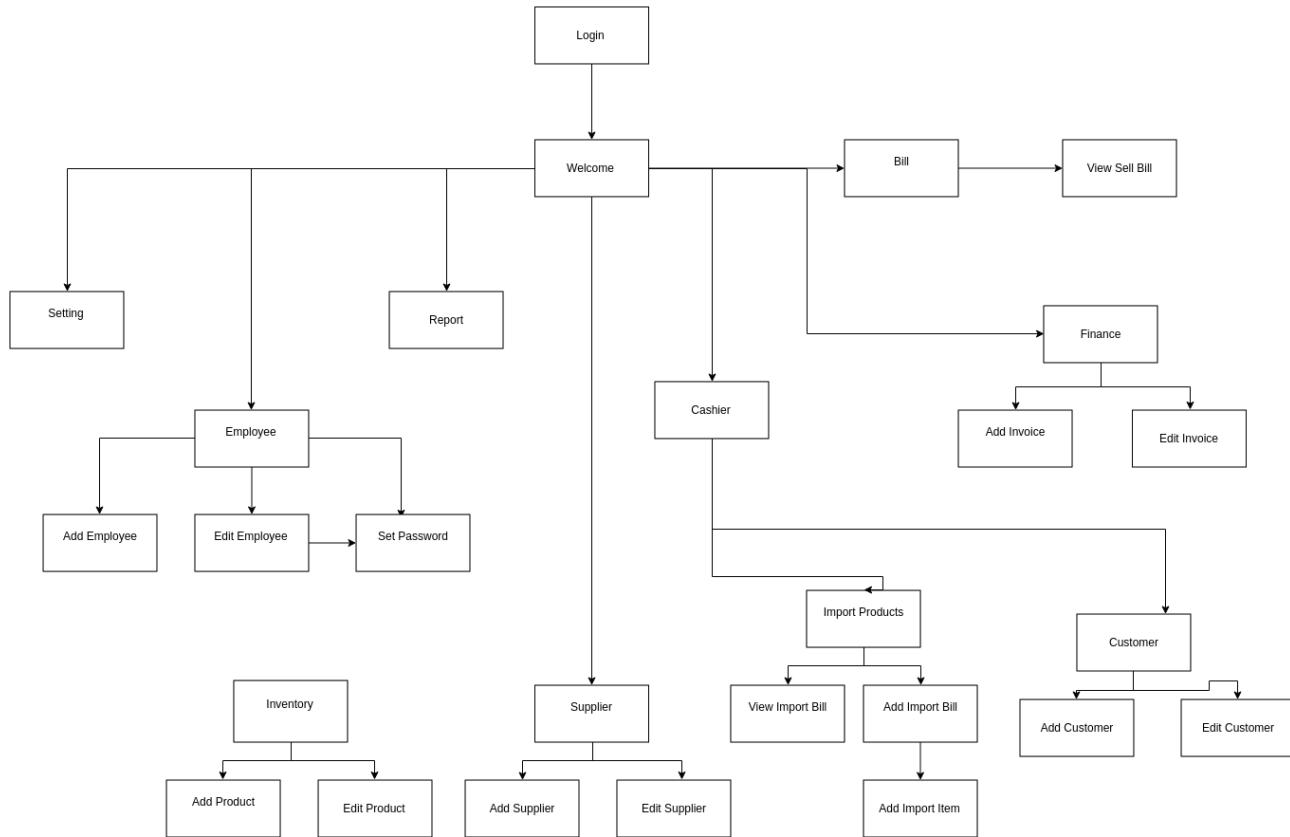
2.4.8 Interaction diagram – Log in



3 Interface Design

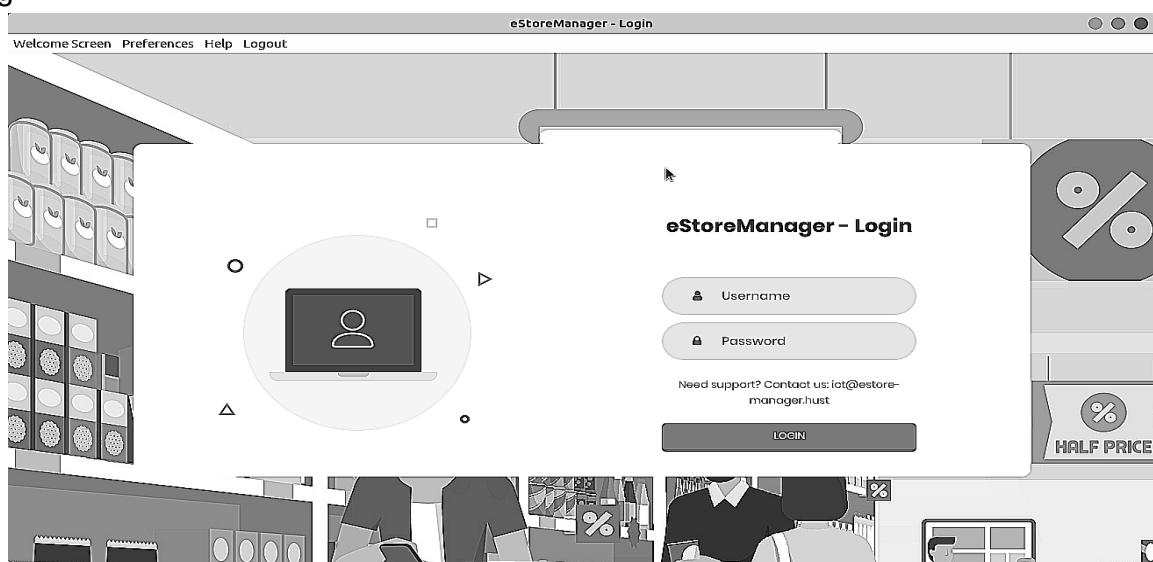
3.1 User Interface

3.1.1 Screen Transition



3.1.2 Screen design and screen specification

* Login



* Setting

eStoreManager - Preferences

The screenshot shows the 'eStoreManager - Preferences' window. At the top left is a gear icon. The title bar says 'eStoreManager - Preferences'. Below the title bar, the word 'Preferences' is displayed. A horizontal line separates this from the 'Server connection' section. In the 'Server connection' section, there are two radio buttons: 'http' (selected) and 'https'. To the right of these are two input fields: 'Hostname' containing 'localhost' and 'Port' containing '8080'. Another horizontal line follows. Below it is the 'Application language' section, which contains a dropdown menu set to 'English'. A third horizontal line is present. The next section is 'Store Information', which includes 'Store Name' (set to 'ICT Store') and 'Store Address' (set to '4th Floor D9 HUST'). A large grey button at the bottom right of the form area is labeled 'SAVE SETTINGS'.

Preferences

Server connection

Hostname: localhost

Port: 8080

Application language: English

Store Information

Store Name: ICT Store

Store Address: 4th Floor D9 HUST

SAVE SETTINGS

* About Us

eStoreManager - About Us

The screenshot shows the 'eStoreManager - About Us' window. At the top left is a storefront icon. The title bar says 'eStoreManager - About Us'. Below the title bar, the word 'About Us' is displayed. A horizontal line follows. The next section is 'Program: eStoreManager'. Below this is a paragraph stating: 'This project is for ITSS Software Development and Structure Programming course at HUST'. Another horizontal line follows. The next section is 'Members: An Nguyen, Viet-Anh, Theo'. Below this is a paragraph stating: 'For more information and support, please contact us at ICT K60 - Hanoi University of Science and Technology.'.

About Us

Program: eStoreManager

This project is for ITSS Software Development and Structure Programming course at HUST

Members: An Nguyen, Viet-Anh, Theo

For more information and support, please contact us at ICT K60 - Hanoi University of Science and Technology.

* Welcome



* Employee

eStoreManager - Employee Management																																																							
WELCOME SCREEN CASHIER BILLS CUSTOMERS INVENTORY SUPPLIERS IMPORT PRODUCTS FINANCE EMPLOYEES REPORTS																																																							
>> EMPLOYEE MANAGEMENT																																																							
<div style="display: flex; justify-content: space-between;"> Show <input type="button" value="5"/> entries Search: <input type="text"/> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>User ID</th> <th>Name</th> <th>Username</th> <th>Salary</th> <th>Email</th> <th>Address</th> <th>Mobile</th> <th>Position</th> <th>Manipulation</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Viet Anh</td> <td>admin</td> <td>2000000</td> <td>admin@gmail.com</td> <td>Ngõ Gốc Đè Minh Khai</td> <td>03482934892384</td> <td>ROLE_ADMIN</td> <td> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Set Password"/> </td> </tr> <tr> <td>2</td> <td>Sỹ An</td> <td>syan</td> <td>1000000</td> <td>an@vietanhdev.com</td> <td>Hòa Bình</td> <td>02384234023434</td> <td>ROLE_ADMIN</td> <td> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Set Password"/> </td> </tr> <tr> <td>3</td> <td>Theo</td> <td>theo</td> <td>500000</td> <td>theo@gmail.com</td> <td>Theo Street</td> <td>032492034838</td> <td>ROLE_CASHIER</td> <td> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Set Password"/> </td> </tr> <tr> <td>4</td> <td>Warehouse keeper</td> <td>keeper</td> <td>5000000</td> <td>keeper@gmail.com</td> <td>Tạ Quang Bửu</td> <td>0384032840823</td> <td>ROLE_MANAGER</td> <td> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Set Password"/> </td> </tr> </tbody> </table>											User ID	Name	Username	Salary	Email	Address	Mobile	Position	Manipulation	1	Viet Anh	admin	2000000	admin@gmail.com	Ngõ Gốc Đè Minh Khai	03482934892384	ROLE_ADMIN	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Set Password"/>	2	Sỹ An	syan	1000000	an@vietanhdev.com	Hòa Bình	02384234023434	ROLE_ADMIN	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Set Password"/>	3	Theo	theo	500000	theo@gmail.com	Theo Street	032492034838	ROLE_CASHIER	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Set Password"/>	4	Warehouse keeper	keeper	5000000	keeper@gmail.com	Tạ Quang Bửu	0384032840823	ROLE_MANAGER	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Set Password"/>
User ID	Name	Username	Salary	Email	Address	Mobile	Position	Manipulation																																															
1	Viet Anh	admin	2000000	admin@gmail.com	Ngõ Gốc Đè Minh Khai	03482934892384	ROLE_ADMIN	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Set Password"/>																																															
2	Sỹ An	syan	1000000	an@vietanhdev.com	Hòa Bình	02384234023434	ROLE_ADMIN	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Set Password"/>																																															
3	Theo	theo	500000	theo@gmail.com	Theo Street	032492034838	ROLE_CASHIER	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Set Password"/>																																															
4	Warehouse keeper	keeper	5000000	keeper@gmail.com	Tạ Quang Bửu	0384032840823	ROLE_MANAGER	<input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Set Password"/>																																															
Showing 1 to 4 of 4 entries Previous 1 Next																																																							

* Employee - Add employee

eStoreManager - Add Employee

Name	Nguyễn Sỹ An		
Username	syan	Email	an@gmail.com
Password	*****	SET PASSWORD	
Address	Hòa Bình	Mobile	02389439040
Salary	15000		
Position	Admin		
SAVE			

* Employee – Set password

eStoreManager - Password Input

Input password

dcb1gzh6h	Eye icon

OK	GENERATE SECURE PASSWORD

* Employee – Edit employee

eStoreManager - Edit Employee

Edit Employee

Name		
<input type="text" value="1"/>	<input type="text" value="Việt Anh"/>	
Username	Email	
<input type="text" value="admin"/>	<input type="text" value="admin@gmail.com"/>	
Password	<input type="text" value="Password"/>	SET PASSWORD
Address	Mobile	
<input type="text" value="Ngõ Gốc Đề Minh Khai"/>	<input type="text" value="03482934892384"/>	
Salary		
<input type="text" value="2000000"/>		
Position		
<input type="text" value="Admin"/>		
SAVE		

* Customer

eStoreManager - Customer Management

Welcome Screen Preferences Help Logout

>> CUSTOMER MANAGEMENT

ADD CUSTOMER						
Show 8 entries		Search: <input type="text"/>				
ID	Customer Name	Email	Address	Mobile	Manipulation	
12	VietAnhDev	vietanh@vietanhdev.com	vietanhdev.com	023849023498		
11	Quang Trung Company	tds@ddd.com	Quang Trung Street	034584938545		
10	Việt Anh ĐT	vietanh@vietanhdev.com	Ngõ Gốc Đề Minh Khai	0348923840982		
8	Quang QQ	q@xx.com	Paris City	0348234354545		
7	Hải Hồng Hách	hhh@gmail.com	Hồ Tiền Street	0234832840823904		
6	Tuấn Tiên Tỉ	ttt@gmail.com	Bách Khoa KTX	034234234545		
5	Dũng Vui vẻ	d@xx.com	Dũng Street	023489234234		
4	Vương Vật Vờ	v@gg.com	Hello Street	04382934892343		

Showing 1 to 8 of 11 entries

Previous 1 2 Next

* Customer – Add customer

eStoreManager - Add Customer

Add Customer

Name

Email

Address

Mobile

SAVE

* Customer – Edit customer

eStoreManager - Edit Customer

Edit Customer

ID <input type="text" value="12"/>	Name <input type="text" value="VietAnhDev"/>
Email <input type="text" value="vietanh@vietanhdev.com"/>	Address <input type="text" value="vietanhdev.com"/>
Mobile <input type="text" value="023849023498"/>	

SAVE

* Inventory – Add product

eStoreManager - Add Product

Add Product

Product Name

Price

Quantity Unit

Barcode

SAVE

* Inventory – Edit product

eStoreManager - Edit Product

Edit Product

ID

Product Name

Price

Quantity Unit

Barcode

SAVE

* Import products

eStoreManager - Import Products

Welcome Screen Preferences Help Logout

WELCOME SCREEN CASHIER BILLS CUSTOMERS INVENTORY SUPPLIERS IMPORT PRODUCTS FINANCE EMPLOYEES REPORTS

>> IMPORT PRODUCTS

Use Date Filter

ADD IMPORT BILL

Show 8 entries Search:

Date	Created By	User ID	Status	# Products	Manipulation
2018-12-05 21:14:37	Việt Anh	1	Discarded	2	VIEW DISCARD
2018-12-05 23:36:53	Việt Anh	1	Discarded	1	VIEW DISCARD
2018-12-06 01:14:43	Việt Anh	1	Completed	3	VIEW DISCARD
2018-12-09 01:15:18	Việt Anh	1	Completed	3	VIEW DISCARD

Showing 1 to 4 of 4 entries Previous 1 Next

* Import products – Add import bill

eStoreManager - Add Import Bill

>> ADD IMPORT BILL

ADD IMPORT ITEM

Show 8 entries Search:

Product Name	Price	Quantity	Unit	Total	Supplier	Manipulation
Bàn Phím Dell 1202	34000	2	Piece	68000	Apache	
Chuột Dell 1201	12000	34	Piece	408000	Hồng Hà Co	
Nước Lavie 1L	12000	34	Bottle	408000	Microsoft	

Showing 1 to 3 of 3 entries Previous 1 Next

IMPORT

* Import products – View import bill

The screenshot shows a completed import bill with the following details:

Product Name	Price	Quantity	Unit	Total	Supplier
Nước Lavie 1L	12000	3	Bottle	36000	Hunadas Co
Nước Lavie 1L	1000	12	Bottle	12000	Bach Khoa ICT
Sữa Vinamilk 350ml	12000	2	Pack	24000	Dinh Tuan Co

Showing 1 to 3 of 3 entries

Created By: Việt Anh | Created At: 2018-12-06 01:14:43

Show 8 entries | Search: []

Previous | 1 | Next

CLOSE

* Import products – Add import bill – Add import item

The screenshot shows the 'Add Import Item' form with the following fields:

Product Name	New Product?	Product ID
Type here to search	<input type="button" value="ADD PRODUCT"/>	
Price		
Quantity	Unit	
Supplier	New Supplier?	Supplier ID
Type here to search	<input type="button" value="ADD SUPPLIER"/>	

SAVE

* Suppliers

eStoreManager - Supplier Management

Welcome Screen Preferences Help Logout

SUPPLIER MANAGEMENT

Show **5** entries **ADD SUPPLIER**

ID	Supplier Name	Email	Address	Mobile	Manipulation
12	Hồng Hà Co	hh@hongha.com	2 Đại Cồ Việt	0238432849	
11	Microsoft	ms@vietanh.com	Không Rõ Ở đâu	032840234324	
10	Intel Company	intel@gmail.com	Silicon Valley	04358435435	
9	Bach Khoa Computer	bk@gmail.com	Le Thanh Nghi Street	02348923434	
8	Dung Store	dung@gmail.com	Góc De ngo	0238472384	

Showing 1 to 5 of 12 entries **Previous** **1** **2** **3** **Next**

* Suppliers – Add supplier

eStoreManager - Add Supplier

Add Supplier

Name

Email

Address

Mobile

SAVE

* Suppliers – Edit supplier

eStoreManager - Edit Supplier

Edit Supplier

ID	Name
<input type="text" value="12"/>	<input type="text" value="Hồng Hà Co"/>
Email	Address
<input type="text" value="hh@hongha.com"/>	<input type="text" value="2 Đại Cồ Việt"/>
Mobile	
<input type="text" value="0238432849"/>	
<input type="button" value="SAVE"/>	

* Finance – Add invoice

eStoreManager - Add Invoice

Add Invoice

Amount
<input type="text" value="12000"/> <input type="button" value="▼"/>
Purpose
<input type="text" value="Buy dog"/>
Description
<input type="text" value="Buy dog for office"/>
<input type="button" value="SAVE"/>

* Finance – Edit invoice

eStoreManager - Edit Invoice

Edit Invoice

ID
1

Amount
50000

Purpose
Buy candy for dogs

Description
Buy 25k for bull dog and 25k for curry dog.

SAVE

* Report

eStoreManager - Reports

Welcome Screen Preferences Help Logout

WELCOME SCREEN CASHIER BILLS CUSTOMERS INVENTORY SUPPLIERS IMPORT PRODUCTS FINANCE EMPLOYEES REPORTS

>> REPORTS

Start Date: 12/01/2018 End Date: 12/31/2018 THIS MONTH UPDATE

TOTAL COST	TOTAL REVENUE	PROFIT
1112002	1800000	687998
Total cost is made up of buying goods, space cost, employee cost...	Total Revenue is made up of the grand total of all selling orders.	Profit is the difference of Total Cost and Total Revenue.

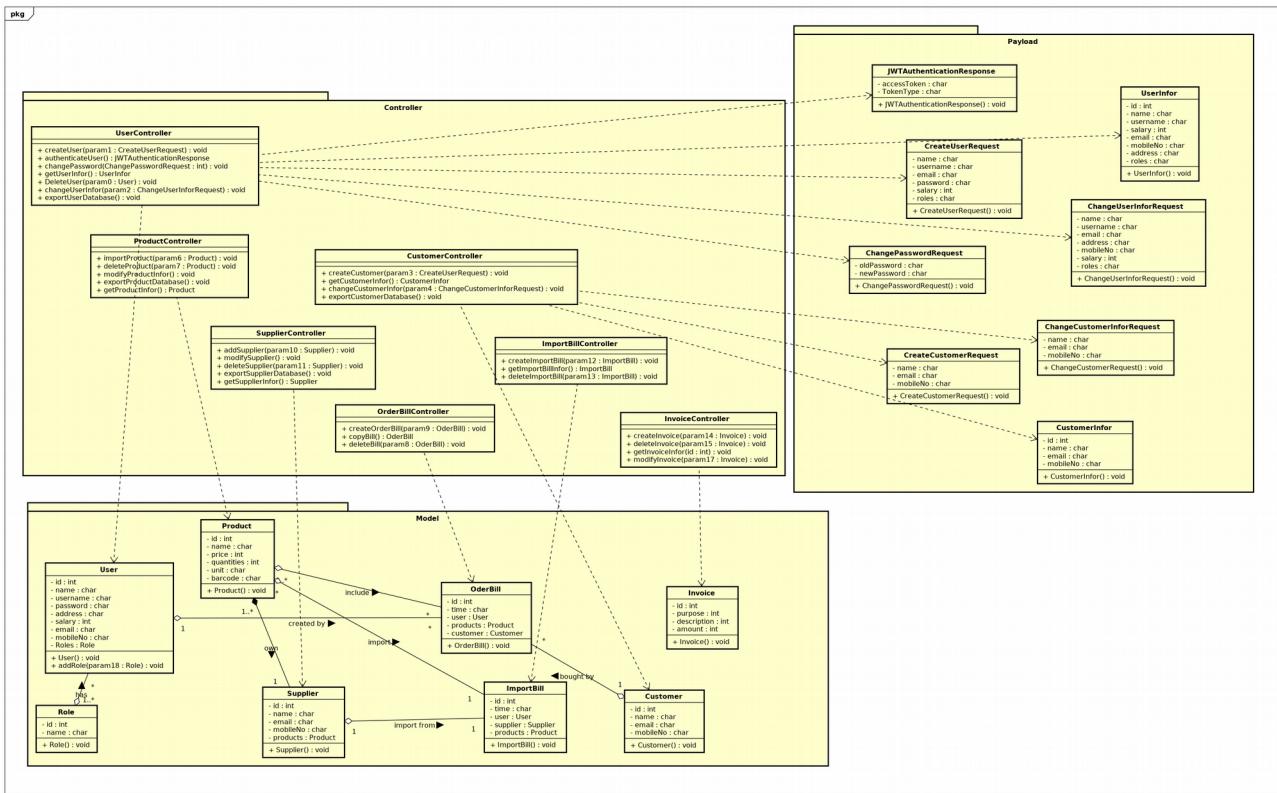
BEST SELLING

- CHUỘT DELL 1201 (SOLD: 6)
- BÀN PHÍM DELL 1202 (SOLD: 5)
- VINASOLD SỮA ĐẬU NÀNH (SOLD: 4)
- NUỐC LAVIE 1L (SOLD: 3)
- SỮA VINAMILK 350ML (SOLD: 3)

Best selling is the list of best seller product by the selling quantity.

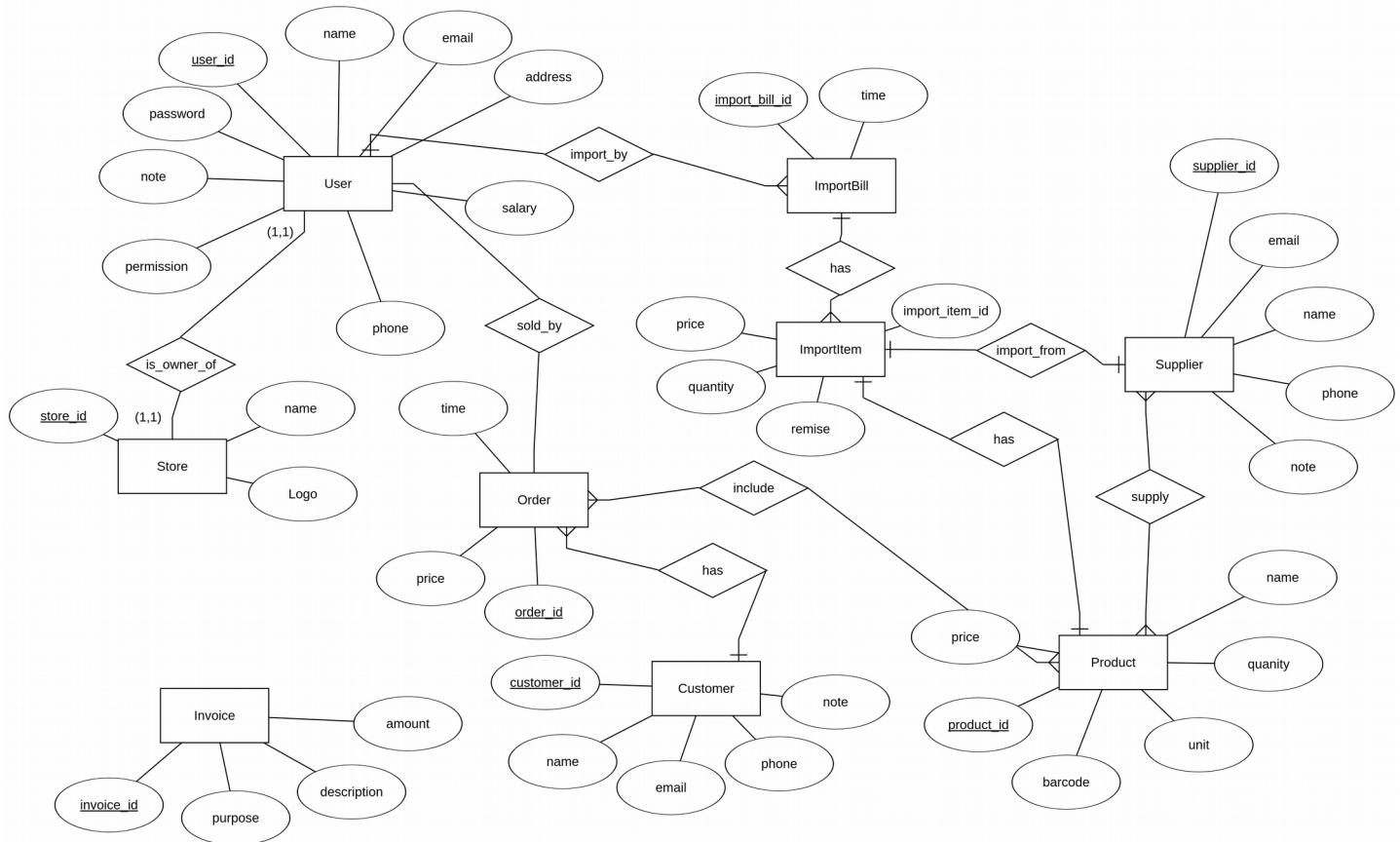
4 Class Design

4.1 Class Diagram

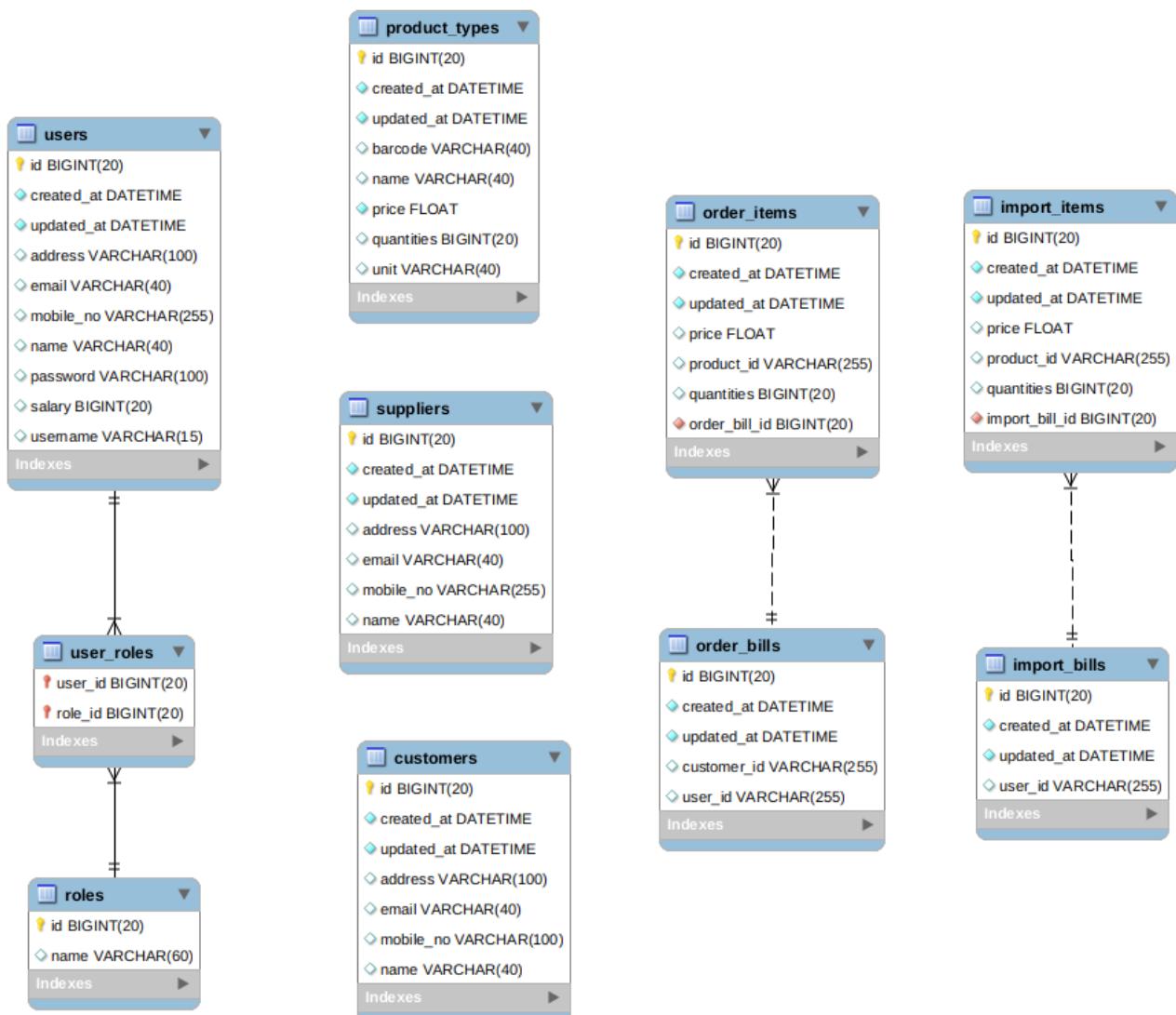


5 Data Model

5.1 Entity Relation Diagram



5.2 Logical Data Model



5.3 Detailed Design

5.3.1 Users

Field	Type	Null	Key	Default	Extra
id	bigint(20)	NO	PRI	NULL	auto_increment
created_at	datetime	NO		NULL	
updated_at	datetime	NO		NULL	
address	varchar(100)	YES		NULL	
email	varchar(40)	YES	UNI	NULL	
mobile_no	varchar(255)	YES		NULL	
name	varchar(40)	YES		NULL	
password	varchar(100)	YES		NULL	
salary	bigint(20)	YES		NULL	
username	varchar(15)	YES	UNI	NULL	

5.3.2 Roles

Field	Type	Null	Key	Default	Extra
id	bigint(20)	NO	PRI	NULL	auto_increment
name	varchar(60)	YES	UNI	NULL	

5.3.3 User_roles

Field	Type	Null	Key	Default	Extra
user_id	bigint(20)	NO	PRI	NULL	
role_id	bigint(20)	NO	PRI	NULL	

5.3.4 Customers

Field	Type	Null	Key	Default	Extra
id	bigint(20)	NO	PRI	NULL	auto_increment
created_at	datetime	NO		NULL	
updated_at	datetime	NO		NULL	
address	varchar(100)	YES		NULL	
email	varchar(40)	YES		NULL	
mobile_no	varchar(100)	YES		NULL	
name	varchar(40)	YES		NULL	

5.3.5 Products

Field	Type	Null	Key	Default	Extra
id	bigint(20)	NO	PRI	NULL	auto_increment
created_at	datetime	NO		NULL	
updated_at	datetime	NO		NULL	
barcode	varchar(40)	YES		NULL	
name	varchar(40)	YES		NULL	
price	float	NO		NULL	
quantities	float	YES		NULL	
unit	varchar(40)	YES		NULL	

5.3.6 Suppliers

Field	Type	Null	Key	Default	Extra
id	bigint(20)	NO	PRI	NULL	auto_increment
created_at	datetime	NO		NULL	
updated_at	datetime	NO		NULL	
address	varchar(100)	YES		NULL	
email	varchar(40)	YES		NULL	
mobile_no	varchar(255)	YES		NULL	
name	varchar(40)	YES		NULL	

5.3.7 Buys

Field	Type	Null	Key	Default	Extra
id	bigint(20)	NO	PRI	NULL	auto_increment
created_at	datetime	NO		NULL	
updated_at	datetime	NO		NULL	
active	bit(1)	YES		NULL	
user_id	bigint(20)	YES		NULL	

5.3.8 Buy_items

Field	Type	Null	Key	Default	Extra
id	bigint(20)	NO	PRI	NULL	auto_increment
created_at	datetime	NO		NULL	
updated_at	datetime	NO		NULL	
price	float	YES		NULL	
product_id	bigint(20)	YES		NULL	
quantities	float	YES		NULL	
supplier_id	bigint(20)	YES		NULL	
buy_id	bigint(20)	NO	MUL	NULL	

5.3.9 Sells

Field	Type	Null	Key	Default	Extra
id	bigint(20)	NO	PRI	NULL	auto_increment
created_at	datetime	NO		NULL	
updated_at	datetime	NO		NULL	
active	bit(1)	YES		NULL	
customer_id	bigint(20)	YES		NULL	
tax	float	YES		NULL	
user_id	bigint(20)	YES		NULL	
total	float	YES		NULL	

5.3.10 Sell_items

Field	Type	Null	Key	Default	Extra
id	bigint(20)	NO	PRI	NULL	auto_increment
created_at	datetime	NO		NULL	
updated_at	datetime	NO		NULL	
price	float	YES		NULL	
product_id	bigint(20)	YES		NULL	
quantities	float	YES		NULL	
sell_id	bigint(20)	NO	MUL	NULL	

5.3.11 Invoices

Field	Type	Null	Key	Default	Extra
id	bigint(20)	NO	PRI	NULL	auto_increment
created_at	datetime	NO		NULL	
updated_at	datetime	NO		NULL	
amount	float	YES		NULL	
description	varchar(255)	YES		NULL	
purpose	varchar(255)	YES		NULL	