

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

Software Requirement Specification

Version 2.0

eStoreManager

Store management solution

Subject: ITSS Software Development

Group 01

Nguyễn Sỹ An

Nguyễn Việt Anh

Theo Mercurio

Hanoi, 12/2018

1. Table of contents

1 Introduction.....	3
1.1 Objective.....	3
1.2 Scope.....	3
1.3 Glossary.....	3
1.4 References.....	4
2 Overall Description.....	5
2.1 Actors.....	5
2.2 Overall use case diagram.....	5
2.3 Decomposition of Use cases.....	6
2.3.1 Use case diagram for “Manage customer list”.....	6
2.3.2 Use case diagram for “Manage products”.....	6
2.3.3 Use case diagram for “Manage Employees”.....	7
2.3.4 Use case diagram for “Manage suppliers”.....	8
2.3.5 Use case diagram for “Manage finance”.....	9
2.3.6 Use case diagram for “Create reports”.....	9
2.4 Business processes.....	10
2.4.1 Process of add bill.....	10
2.4.2 Process of import products.....	11
2.4.3 Process of add employee.....	11
2.4.4 Process of add supplier.....	12
2.4.5 Process of add customer.....	12
3 Specific requirements.....	13
3.1 Specification of Use case UC001 - “Log in”.....	13
3.2 Specification of Use case UC002- “Change password”.....	14
3.3 Specification of Use case UC003 - “Add Employee”.....	15
3.4 Specification of Use case UC004- “Add Customer”.....	15

3.5 Specification of Use case UC005 - “Add supplier”	16
3.6 Specification of Use case UC006 - “Modify customer information”	17
3.7 Specification of Use case UC007 - “Add bill”	18
3.8 Specification of Use case UC008- “Import products”	19
3.9 Specification of Use case UC009 - “Return products”	20
4 Supplementary.....	21
Functionality.....	21
Usability.....	21
Reliability.....	21
Performance.....	21
Supportability.....	21
Other requirements.....	21

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 Overall Description

2.1 Actors

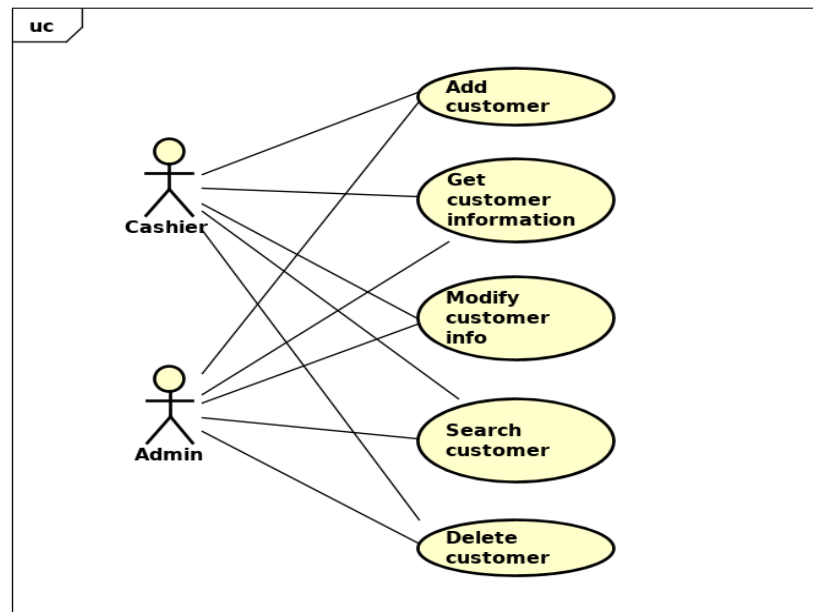
There are 3 main actors in the system: *Admin*, *Cashier*, *Warehouse manager*.

2.2 Overall use case diagram

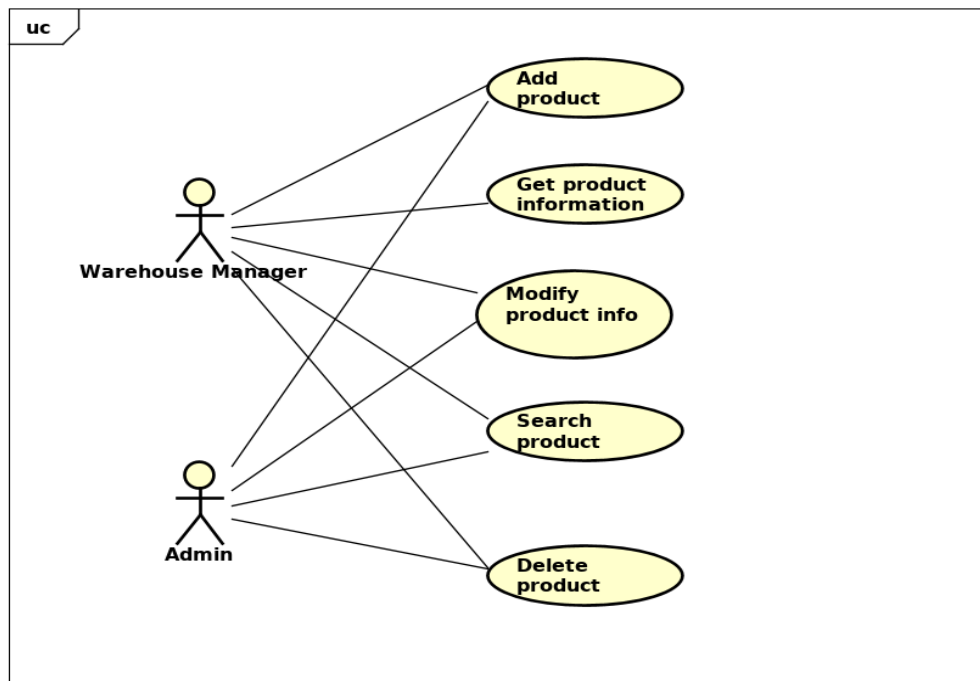


2.3 Decomposition of Use cases

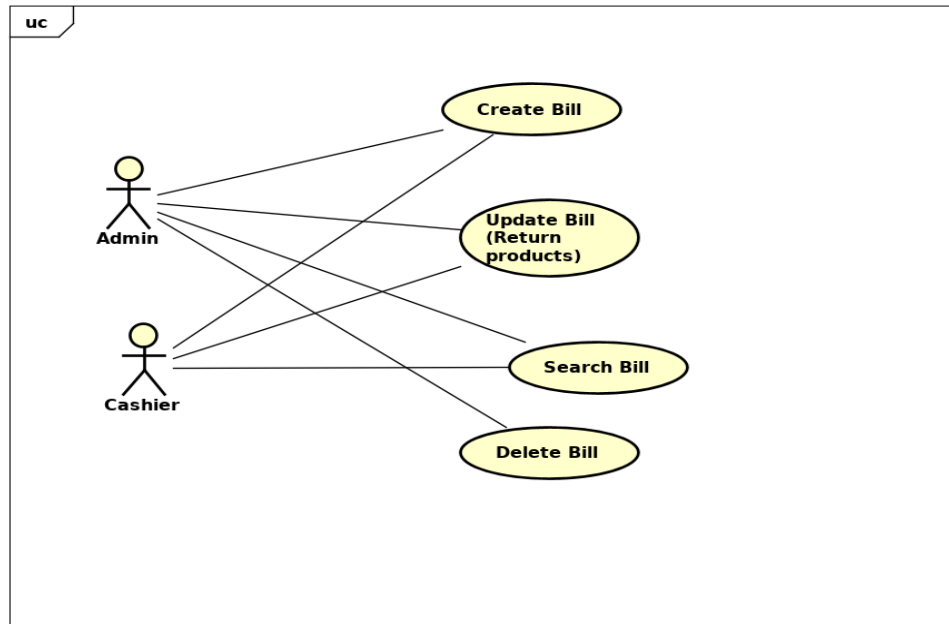
2.3.1 Use case diagram for “Manage customer list”



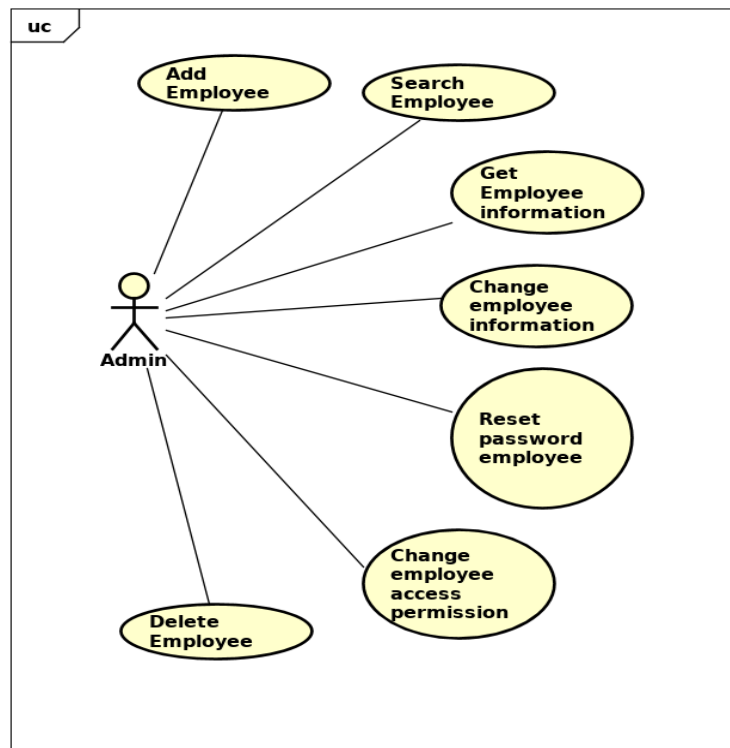
2.3.2 Use case diagram for “Manage products”



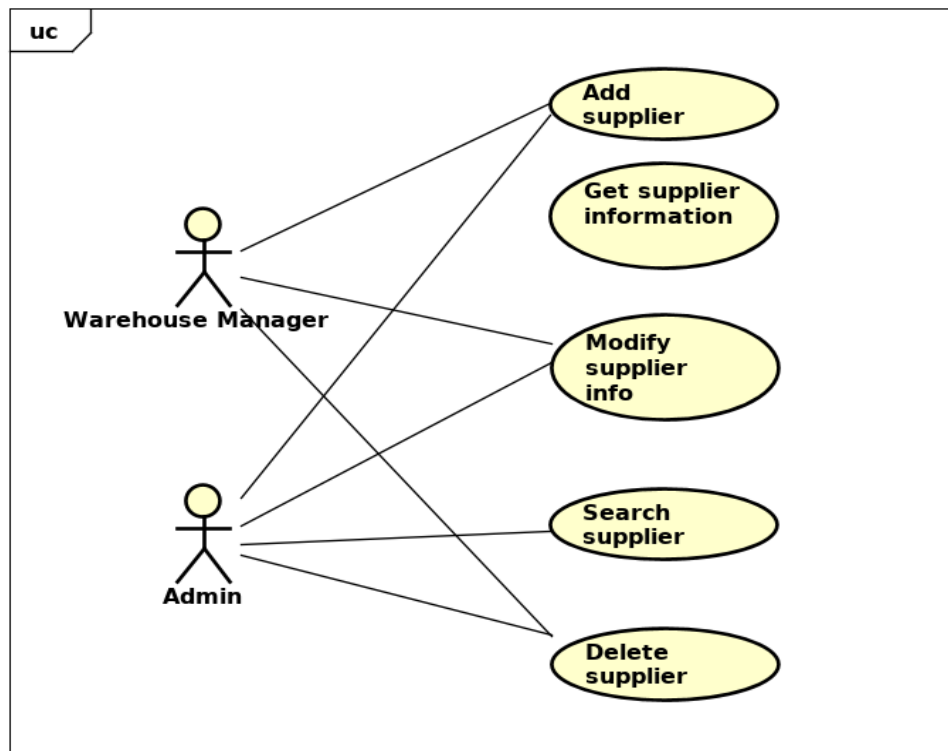
2.3.3 Use case diagram for “Manage Bills”



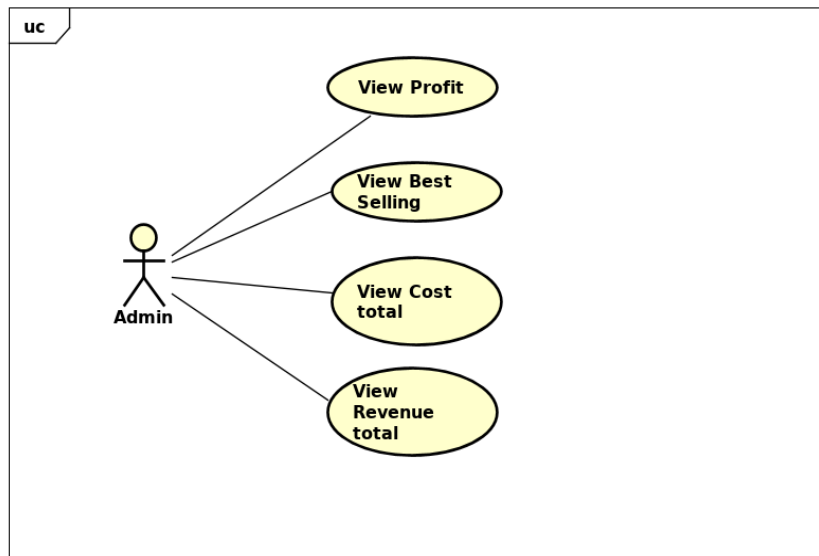
2.3.4 Use case diagram for “Manage Employees”



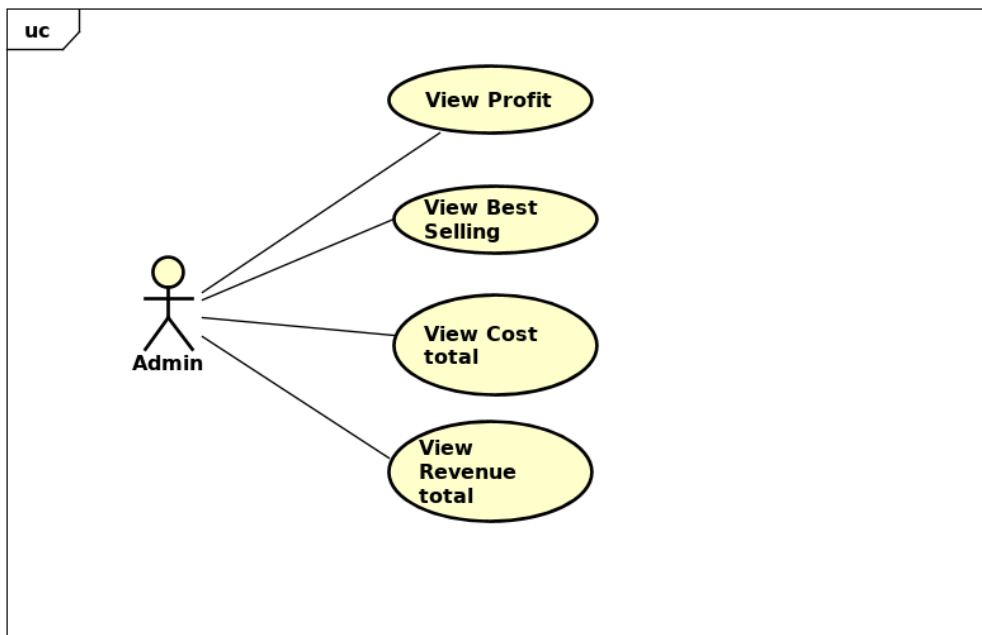
2.3.5 Use case diagram for “Manage suppliers”



2.3.6 Use case diagram for “Manage finance”

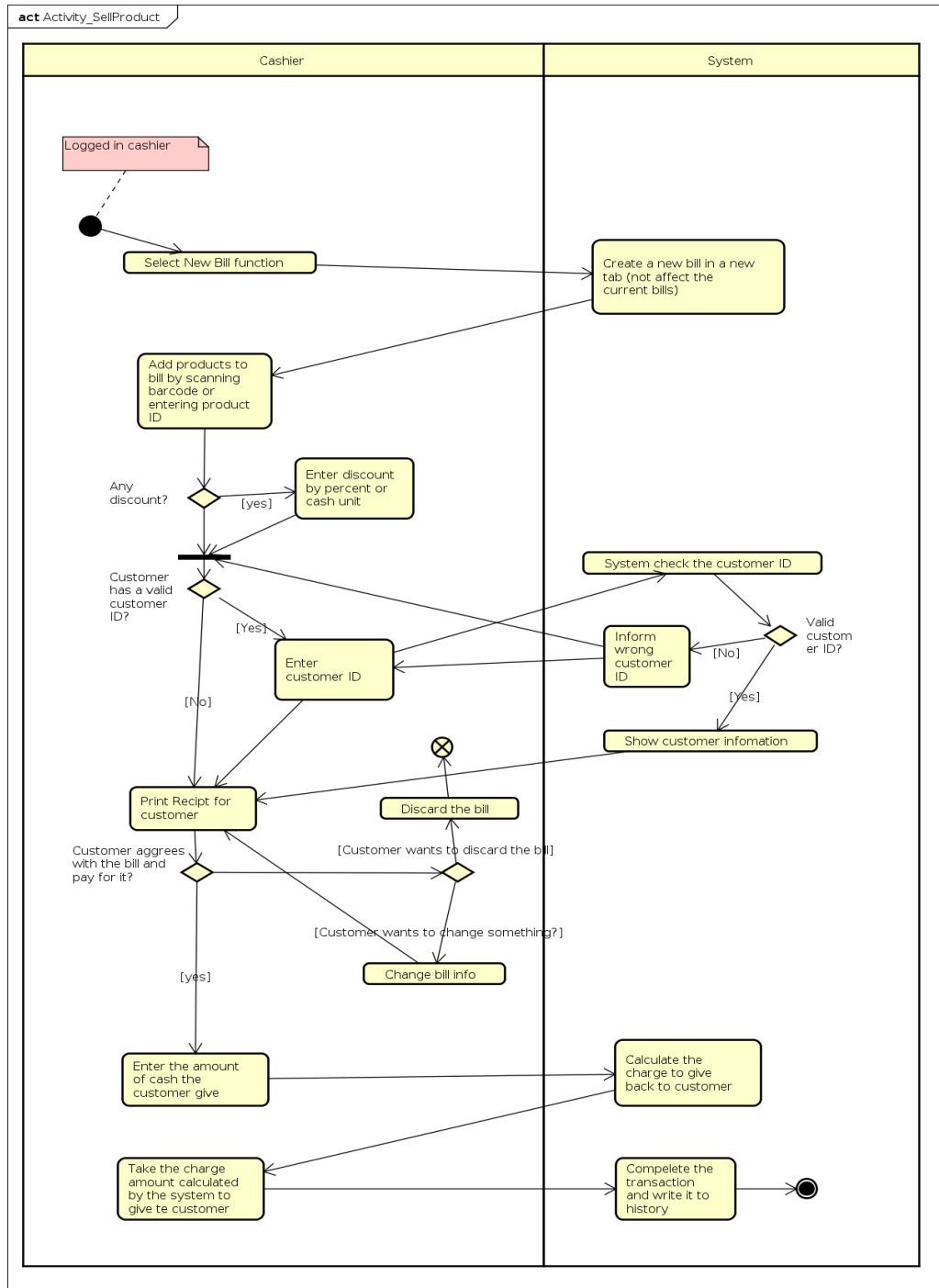


2.3.7 Use case diagram for “Create reports”



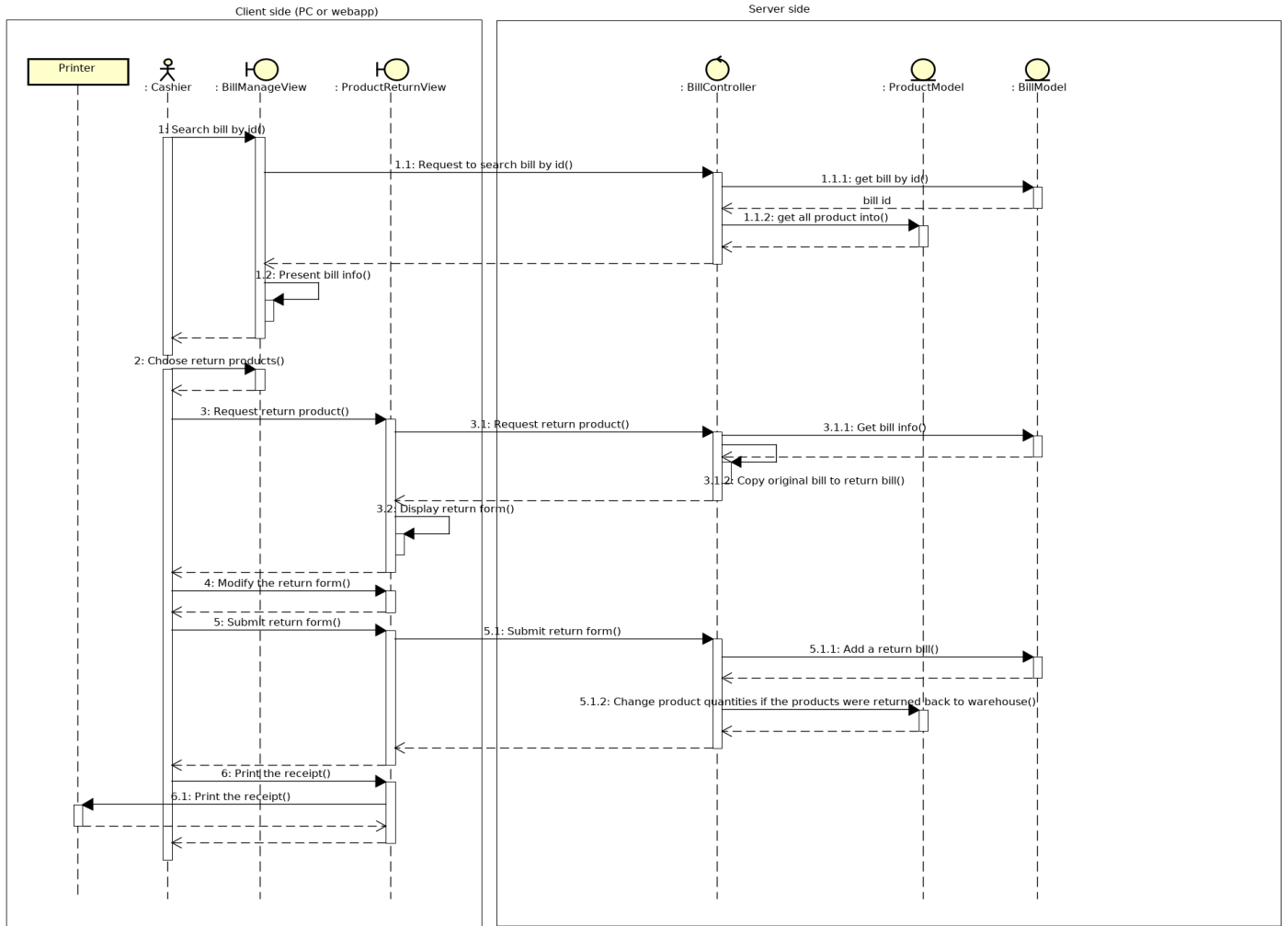
2.4 Business processes

2.4.1 Process of add bill

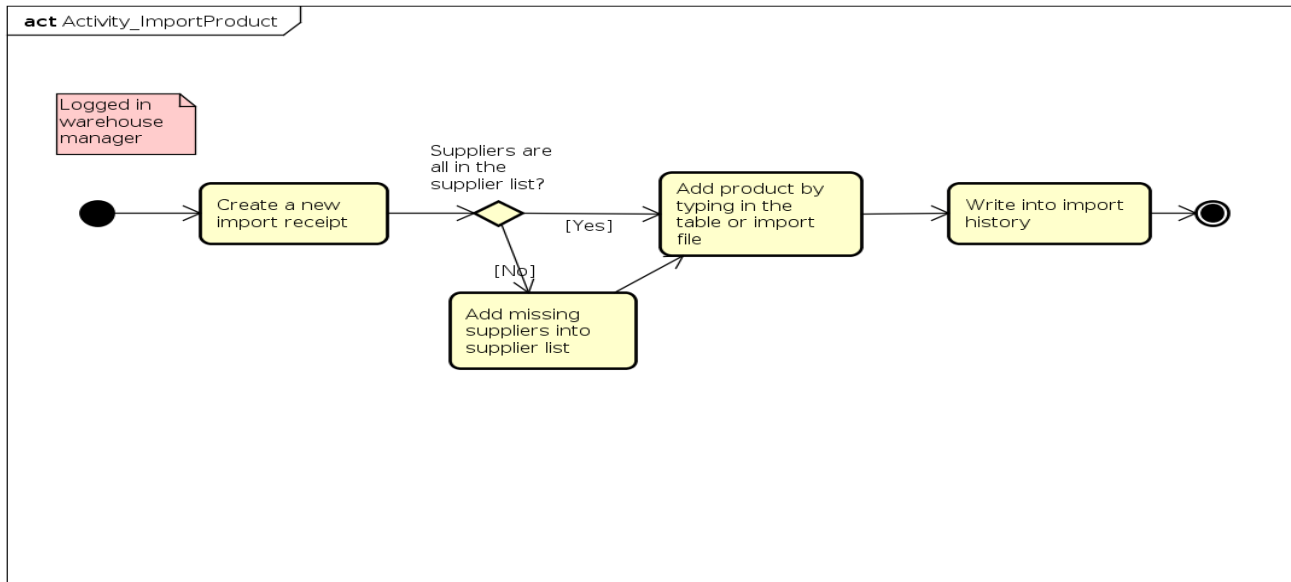


2.4.2 Process of return products

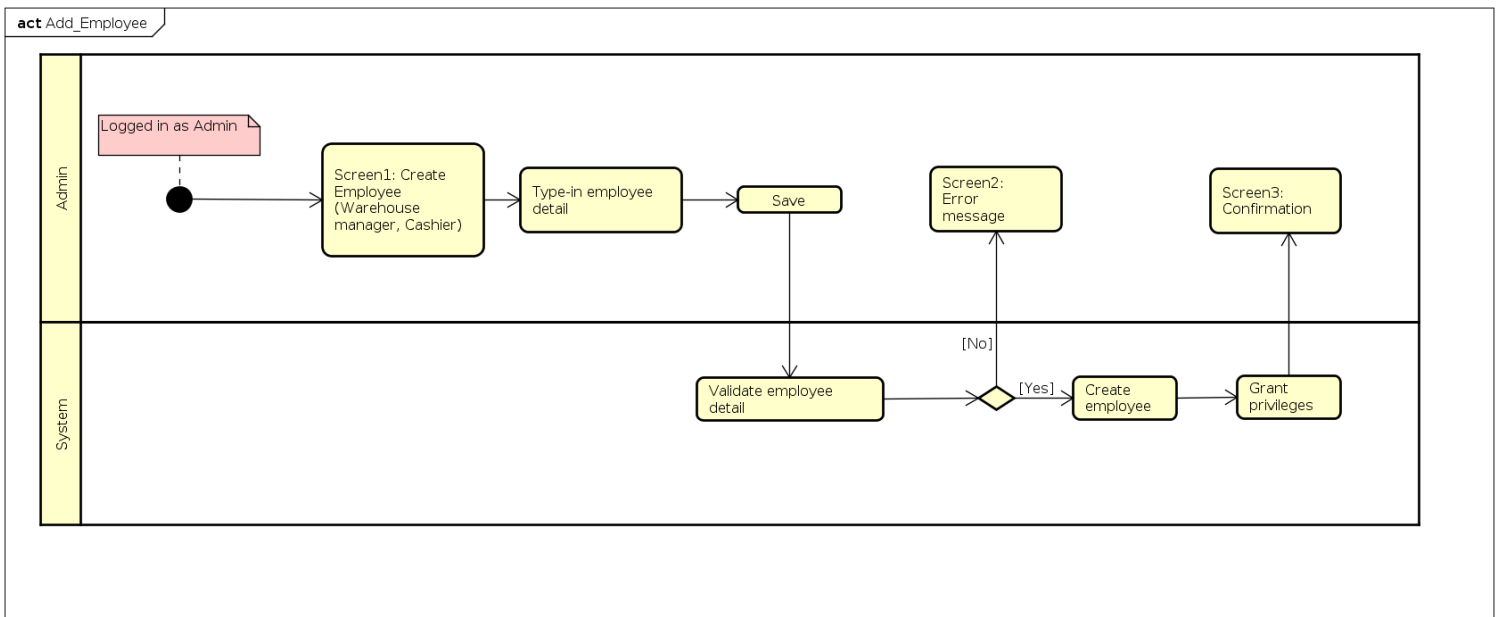
sd ReturnProduct _ Sequence Diagram



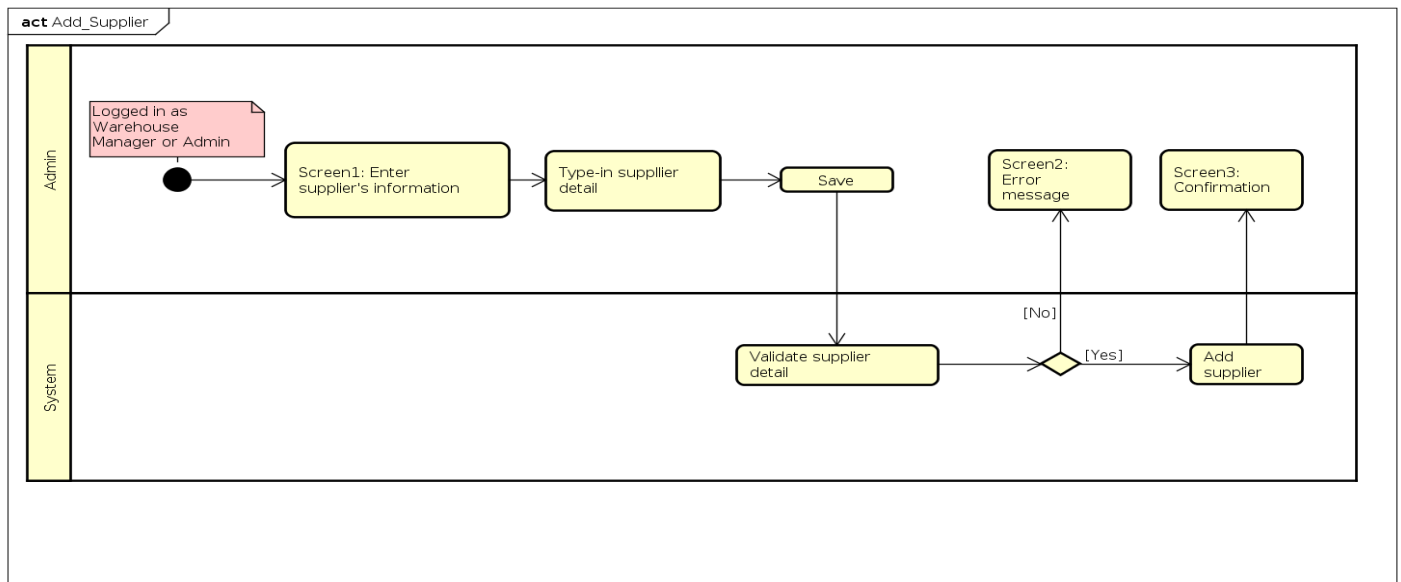
2.4.3 Process of import products



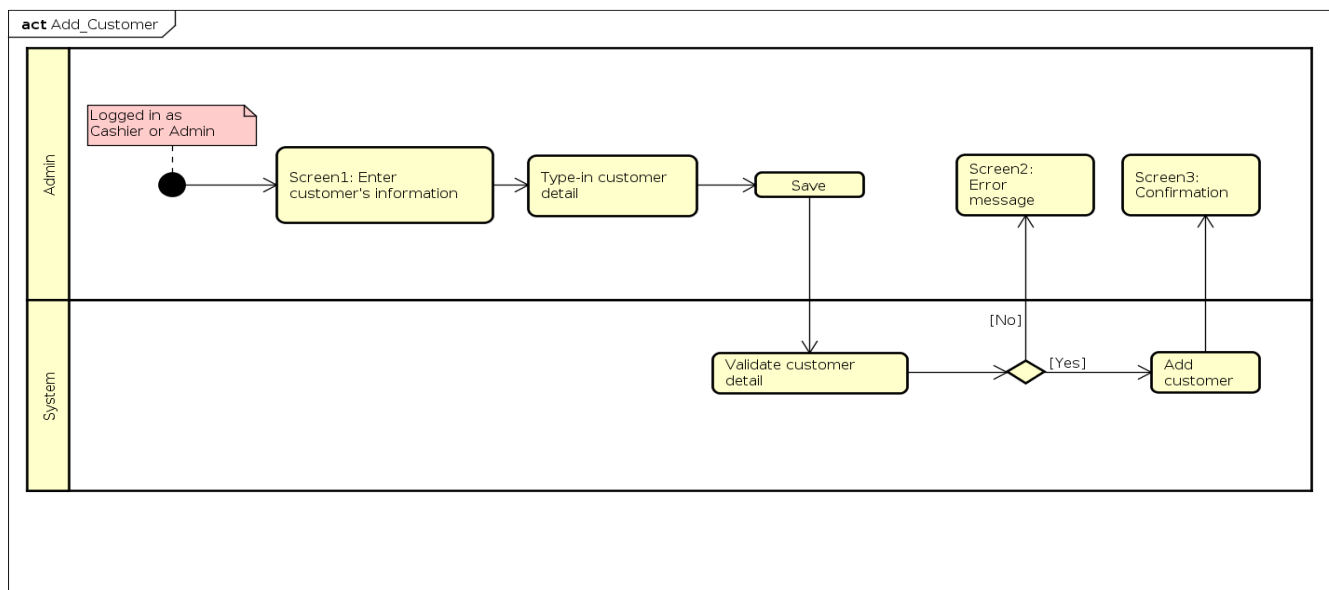
2.4.4 Process of add employee



2.4.5 Process of add supplier



2.4.6 Process of add customer



3 Specific requirements

Details of the use cases given in following sections are specified below.

3.1 Specification of Use case UC001 - “Log in”

Use case code	UC001	Use case name	Sell Product
Actor	User (Cashier, Warehouse Manager, Admin)		
Pre-condition	None		
Main flow of events (Success)	No	Actor	Action
	1	User	Select Log in function
	2	System	Display log in interface
	3	User	Input user name and passwords
	4	User	Request log in
	5	System	Check whether user had input the required field
	6	System	Validate the user name and password
	7	System	Log user into the system
	8	Sytem	Display the Main Form and the use case end
Alternative flow of events	No	Actor	Action
	6a	Cashier	Output error: need to put into the missing required field if had not done so
	7a	System	Output error: If the user name and password is invalid, require user enter user name and password again
Post-condition	None		

3.2 Specification of Use case UC002- “Change password”

Use case code	UC002	Use case name	Change password
Actor	User (Cashier, Warehouse Manager, Admin)		
Pre-condition	User logged in to system		
Main flow of events (Success)	No	Actor	Action
	1	User	Select change password function
	2	System	Display change password interface, require confirm old password and new password
	3	Cashier	Enter old password and new password
	4	Cashier	Check whether user had input the required field
	5	Cashier	Check the new password is secure enough or not
	6	Cashier	Check the old password is right or not
	7	System	Display change password successfully, save new password
Alternative flow of events	No	Actor	Action
	5a.	Cashier	Output error: need to put into the missing required field if had not done so
	6a.	System	Output error: the new password is not secure enough, back to step confirm password
	7a.	System	Output error: the old password is not right, back to step confirm password
Post-condition	None		

3.3 Specification of Use case UC003 - “Add Employee”

Use case code	UC003	Use case name	Add Employee
Actor	Admin		
Pre-condition	Admin logged in to system		
Main flow of events (Success)	No	Actor	Action
	1	Admin	Select add employee function
	2	System	Display form to enter employee information
	3	Admin	Enter employee information then choose save
	4	System	Validate employee information
	5	System	Create new employee in system
	6	System	Grant privileges for employee
	7	System	Confirm save new employee successfully
Alternative flow of events	No	Actor	Action
	5a	System	Display error in validating employee information
Post-condition	None		

3.4 Specification of Use case UC004- “Add Customer”

Use case code	UC004	Use case name	Add Customer
Actor	Cashier/Admin		
Pre-condition	Cashier/Admin logged in to system		

Main flow of events (Success)	No	Actor	Action
	1	Cashier/ Admin	Select add customer function
	2	System	Display form to enter customer information
	3	Cashier/ Admin	Enter customer information then choose save
	4	System	Validate customer information
	5	System	Create new customer in system
	6	System	Confirm save new employee successfully
Alternative flow of events	No	Actor	Action
	5a	System	Display error in validating employee information
Post-condition	None		

3.5 Specification of Use case UC005 - “Add supplier”

Use case code	UC005	Use case name	Add Supplier
Actor	Warehouse Manager/Admin		
Pre-condition	Warehouse Manager/Admin logged in to system		
Main flow of events (Success)	No	Actor	Action
	1	Warehouse Manager/Admin	Select add supplier function
	2	System	Display form to enter supplier information
	3	Warehouse Manager/Admin	Enter supplier information then choose save
	4	System	Validate supplier information

	5	System	Create new supplier in system
	6	System	Confirm save new supplier successfully
Alternative flow of events	No	Actor	Action
	5a	System	Display error in validating supplier information
Post-condition	None		

3.6 Specification of Use case UC006 - "Modify customer information"

Use case code	UC006	Use case name	Modify customer information
Actor	Cashier/Admin		
Pre-condition	Cashier/Admin logged in to system		
Main flow of events (Success)	No	Actor	Action
	1	Admin	Select modify customer information function
	2	System	Display form to enter customer information (with copy of old information)
	3	Admin	Enter new customer information then choose save
	4	System	Validate customer information
	6	System	Save change
	7	System	Confirm save change successfully
Alternative flow of events	No	Actor	Action
	5a	System	Display error in validating customer information
Post-condition	None		

3.7 Specification of Use case UC007 - "Add bill"

Use case code	UC007	Use case name	Add bill
Actor	Cashier/Admin		
Pre-condition	Cashier/Admin has logged in		
Main flow of events (Success)	No	Actor	Action
	1	Cashier/ Admin	Select New Bill function
	2	System	Create a new bill in a new tab (not affect the current bills)
	3	Cashier/ Admin	Add products to bill by scanning barcode or entering product ID
	4	Cashier/ Admin	Enter discount by percent or cash unit if there is any discount
	5	CCashier/ Admin	Print receipt for customer
	6	Cashier/ Admin	Receive cash from customer and Enter the amount of cash the customer give
	7	System	Calculate the charge to give back to customer
	8	Cashier/ Admin	Take the charge amount calculated by the system to give te customer
	9	System	Compelete the transaction and write it to history
Alternative flow of events	No	Actor	Action
	5a1	Cashier/ Admin	Enter customer ID if customer has a valid id (may be from customer card)
	5a2	System	Check the customer id
	5a3	System	Display the customer information

	7a	Cashier/ Admin	If customer does not agree with the bill, discard it
	7b	Cashier/ Admin	If customer wants to correct something, change the bill and go back to step 5.
Post-condition	None		

3.8 Specification of Use case UC008- "Import products"

Use case Code	UC008		Use case Name	Import products
Actor	Warehouse manager			
Pre-condition	Warehouse manager has logged in			
Main flow of events (Success)	No	Actor	Action	
	1	Warehouse manager	Create a new import receipt	
	2	Warehouse manager	Add product by typing in the table or import file	
	3	System	Write into import history	
Alternative flow of events	No	Actor	Action	
	2a	Warehouse manager	If goods are importing from unlisted supplier, the warehouse manager go to supplier list to add.	
Post-conditions	None			

3.9 Specification of Use case UC009 - “Return products”

Use case code	UC009	Use case name	Return products
Actor	Cashier/Admin		
Pre-condition	Cashier/Admin has logged in		
Main flow of events (Success)	No	Actor	Action
	1	Cashier/Admin	Entering bill number or search for bill
	2	System	Find bill match to searching then display to screen
	3	Cashier/Admin	Choose bill to edit, return product
	4	System	Copy “sale receipt” to “return receipt”
	5	Cashier/Admin	Correct the product quantities and price to return
	6	Cashier/Admin	Choose the reason to return
	7	Cashier/Admin	Choose the process to do with return goods (return money to customer or not)
	8	System	Save the new receipt
	9	System	Print the receipt
Alternative flow of events	No	Actor	Action
	8	System	Write to finance history in case returning money to customer
Post-condition	None		

4 Supplementary

- **Functionality**

Multi machine – one database. You can use this app by client application and connect to server.

- **Usability**

Train the employees and store owner to use the system

- **Reliability**

Automatically restart the server after a system failure.

- **Performance**

Acceptable performance. No lag when using.

- **Supportability**

Host and maintain the system for the stores.

- **Other requirements**

Nice and modern user interface.