GENERAL CONFEDERATION OF LABOR OF VIETNAM **TON DUC THANG UNIVERSITY FACULTY OF INFORMATION TECHNOLOGY**



**SOFTWARE ENGINEERING**

**COMPUTER SHOP MANAGEMENT WEBSITE**

*Instructing Lecturer*:**Mr. ĐẶNG MINH THẮNG**

*Student’s name*: **THI NGỌC PHÚ– 518H0044**

**NGUYỄN KHÁNH VINH-518H0076**

Class**:** **18H50201**

Course**:** **22**

**HO CHI MINH CITY, May 2020**

i

# **ACKNOWLEDGEMENT**

In order to make this report complete and achieve good results, we have received the support and assistance of many teachers and classmates.

With deep affection, sincerity, we express deep gratitude to all individuals and agencies who have helped us in our study and research.

First of all, I would like to express a special appreciation to Ton Duc Thang.

University’s teachers for their conscientious guidance and advices throughout the last semester by gave me their modern outlook and meticulous supervision to carry out the job perfectly.

Especially we would like to send our sincere thanks to Mr. Dang Minh Thang has paid attention, help, guide us to complete the report well over the past time.

We would like to express our sincere gratitude to the leadership of Ton Duc Thang University for supporting, helping and facilitating us to complete the report well during the study period.

With limited time and experience, this report cannot avoid mistakes. We are looking forward to receiving advice and comments from teachers so that we can improve our awareness, better serve the practical work later.

We sincerely thank you!

ii

# **THE PROJECT WAS COMPLETED AT TON DUC THANG UNVERSITY**

I pledge that this is a product of our own project and is under the guidance of Mr. Nguyen Khanh Tung. The content of research, results in this subject is honest and not published in any form before. The data in the tables used for the analysis, comment, and evaluation were collected by the authors themselves from various sources indicated in the reference section. In addition, many comments and assessments as well as data from other authors and organizations have been used in the project, with references and annotations.

**If any fraud is found, I am fully responsible for the content of my project.**

Ton Duc Thang University is not involved in any copyright infringement or copyright infringement in the course of implementation (if any).

*Ho Chi Minh City, May 24th 2020*

*Author*

*(Sign full name)*

*Signed*

*Thi Ngọc Phú*

iii

# **EVALUATION OF INSTRUCTING LECTURER**

**Confirmation of the instructor**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Ho Chi Minh City, May 24th 2020*

(sign full name)

**The assessment of the teacher marked**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Ho Chi Minh City, May 24th 2020*

(sign full name)

**TABLE OF CONTENTS**

[**ACKNOWLEDGEMENT** 2](#_Toc41256458)

[**THE PROJECT WAS COMPLETED AT TON DUC THANG UNVERSITY** 3](#_Toc41256459)

[**EVALUATION OF INSTRUCTING LECTURER** 4](#_Toc41256460)

[1.1 Business Introduction 2](#_Toc41256461)

[1.2 Problems of the shop 3](#_Toc41256462)

[1.3 Q&A table 3](#_Toc41256463)

[1.4 Functional and non-functional requirements 5](#_Toc41256464)

[II. System requirements analysis 6](#_Toc41256465)

[2.1 Use case diagrams 6](#_Toc41256466)

[2.2 Use case specification 8](#_Toc41256467)

[2.2.1 Login 8](#_Toc41256468)

[2.2.2 Log out 8](#_Toc41256469)

[2.2.3 View account information 9](#_Toc41256470)

[2.2.4 Update account information 10](#_Toc41256471)

[2.2.5 Add product 11](#_Toc41256472)

[2.2.6 Delete product 12](#_Toc41256473)

[2.2.7 Edit product 13](#_Toc41256474)

[2.2.8 Add order 14](#_Toc41256475)

[2.2.9 Delete order 15](#_Toc41256476)

[2.2.10 Edit order 16](#_Toc41256477)

[2.2.11 Add employee 17](#_Toc41256478)

[2.2.12 Delete employee 18](#_Toc41256479)

[2.2.13 Edit employee salary 19](#_Toc41256480)

[2.2.14 Add customer 20](#_Toc41256481)

[2.2.15 Delete customer 21](#_Toc41256482)

[2.2.16 Edit customer 21](#_Toc41256483)

[2.3 Activity diagrams 23](#_Toc41256484)

[2.4 Sequence diagrams 24](#_Toc41256485)

[III. System requirements desgin 25](#_Toc41256486)

[3.1 Class diagram 25](#_Toc41256487)

[3.2 Entity relationship diagram 26](#_Toc41256488)

[3.3 Relational data model 27](#_Toc41256489)

[3.4 Database design 28](#_Toc41256490)

[IV. System UI 30](#_Toc41256491)

[V. Github 33](#_Toc41256492)

**TABLE OF FIRGURES**

**FIRGURE**

[Figure 2. 1 Activity diagram of Computer shop management with admin privellege 23](file:///C:\Users\virgo\Desktop\Computer_Management\Report\518H0044_ThiNgocPhu_CNPM_RP.docx#_Toc41256493)

[Figure 2. 2 Sequence diagram of admin interaction with functions process 24](file:///C:\Users\virgo\Desktop\Computer_Management\Report\518H0044_ThiNgocPhu_CNPM_RP.docx#_Toc41256494)

[Figure 2. 3 Sequence diagram of login process 24](file:///C:\Users\virgo\Desktop\Computer_Management\Report\518H0044_ThiNgocPhu_CNPM_RP.docx#_Toc41256495)

[Figure 3. 1 Class diagram 25](file:///C:\Users\virgo\Desktop\Computer_Management\Report\518H0044_ThiNgocPhu_CNPM_RP.docx#_Toc41256560)

[Figure 3. 2 Entity relationship diagram 26](file:///C:\Users\virgo\Desktop\Computer_Management\Report\518H0044_ThiNgocPhu_CNPM_RP.docx#_Toc41256561)

[Figure 3. 3 Relational model diagram 27](file:///C:\Users\virgo\Desktop\Computer_Management\Report\518H0044_ThiNgocPhu_CNPM_RP.docx#_Toc41256562)

[Figure 3. 4 Account database 28](file:///C:\Users\virgo\Desktop\Computer_Management\Report\518H0044_ThiNgocPhu_CNPM_RP.docx#_Toc41256563)

[Figure 3. 5 Employee database 28](file:///C:\Users\virgo\Desktop\Computer_Management\Report\518H0044_ThiNgocPhu_CNPM_RP.docx#_Toc41256564)

[Figure 3. 6 Customer database 28](file:///C:\Users\virgo\Desktop\Computer_Management\Report\518H0044_ThiNgocPhu_CNPM_RP.docx#_Toc41256565)

[Figure 3. 7 OrderOriginal database 29](file:///C:\Users\virgo\Desktop\Computer_Management\Report\518H0044_ThiNgocPhu_CNPM_RP.docx#_Toc41256566)

[Figure 3. 8 Order database 29](file:///C:\Users\virgo\Desktop\Computer_Management\Report\518H0044_ThiNgocPhu_CNPM_RP.docx#_Toc41256567)

[Figure 3. 9 Product database 29](file:///C:\Users\virgo\Desktop\Computer_Management\Report\518H0044_ThiNgocPhu_CNPM_RP.docx#_Toc41256568)

[Figure 4. 1 Login Interface 30](#_Toc41256569)

[Figure 4. 2 Main Page 31](#_Toc41256570)

[Figure 4. 3 Customer Management function 31](#_Toc41256571)

[Figure 4. 4 User Management function 32](#_Toc41256572)

[Figure 4. 5 Order function 32](#_Toc41256573)

[Figure 4. 6 Product function 32](#_Toc41256574)

1. Requirements engneering
   1. Business Introduction

The shop we chose to survey is a medium computer shop located on Hung Vuong street, district 5. The business scale is not too big but it has enough functions of a computer shop to operate. The products that this shop sells including:

* Hardware : Mainboard,CPU,graphic card, RAM, HDD, SSD, PSU, computer case, air cooler fan, internet card.
* Peripheral device : Monitor, keyboard, mouse, headphone, speaker.
* Software : Windows 7/8/10, anti-virus software,…

The shop also has employees to reach the best efficiency. These including :

* Salesman : Sell the products, give advice to customer.
* Acounting : Maintain financial records, calculate tax.
* Storage manager : Manage items in storage, re-stock.

Busy as it sound, but the shop doesn’t have a software to manage these operation.

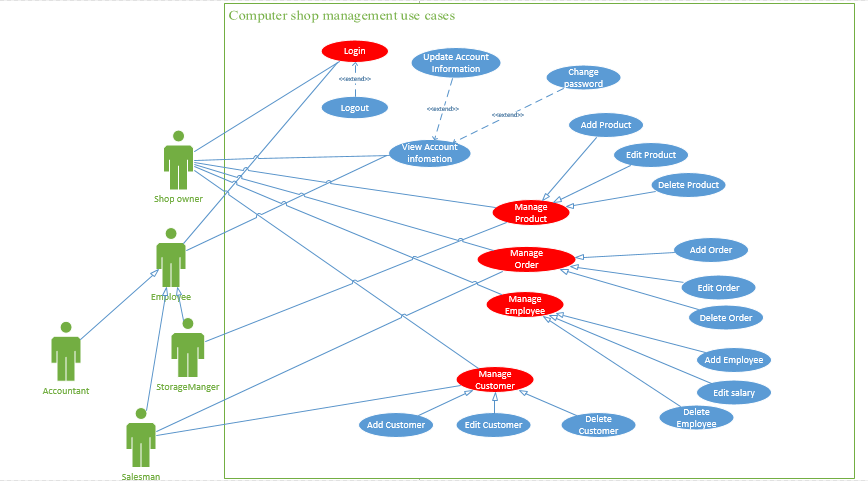
* 1. Problems of the shop
* Storage manager stil uses excel to import/export data.
* The management is lack of profesional.
* The salesman still uses notebook to store customer-order detail.
* Doesn’t have a tool to manage employee.
  1. Q&A table

|  |  |  |
| --- | --- | --- |
| **Section** | **Comment/Question** | **Answer** |
| Login form | Question : About system login, we don’t have a clear view about system authorization. How many role did the system include and how is the authorization ?  Suggest : We will have many roles, such as n different roles similar to the actor role. The system will interfere the functions access of actor depend on their privellege. So we need to know how many roles did you need so that we can create the database. | After the discussion, both sides agree to have two main roles like the following :   * Admin : With the admin privellege, user can use all the functions in the system. This include : manage employee,product,customer,order * Storage manager: Manage product * Salesman : Manage customer and order list |
| Employee view form | Question : About employee view screen, which information will be displayed and how can user interact with information in the view screen ?  Suggest : We need to discuss to the shop owner about the information of the shop employees he want to display. We suggest to have a action bar where the delete and view function option is displayed. | After the discussion with the shop owner, we agreed to have the employee view screen showed the basic information of the employee in the main view screen. To view the ‘discrete’ information like salary, he/she will need to click the view button in the action bar to navigate to full employe view screen where he can edit personal information. Only the admin can edit salary. |
| Function in main page | Question: How to desgin the main screen that user can use their function(s) convieniently.  Suggest : We suggest that each employee will has an account created by admin to log into system and use his/her main function | We will limit the employee to use his/her function base on their role (E.g: Salesman uses manage order and customer function). When a user logged in the system,the system will only show functions depend on his/her role. Except user with admin privellege will see all functions. |
| Order view screen | Question: About the order screen, how it will be designed ?  Suggest : We will design a multi-pages order view screen so the user will have a better look on the order list. | The order view screen will have 2 parts. First part will include main information of the order ( OrderID, Employee created,customer name, purchase method…). The second part is order detail To view order detail which is the second part, user can click on view detail option to view the detail of the order, |

* 1. Functional and non-functional requirements

|  |  |  |
| --- | --- | --- |
| Requirements | Functional | Non-functional |
| The system must limit user to use their approriate function | x |  |
| The login button must be on the top right of the main page |  | x |
| The system must allow admin to create employee account | x |  |
| The system must allow admin to add/delete employee | x |  |
| The system must allow user to edit their account information. | x |  |
| The system must not allow user without admin authorization to change the salary information | x |  |
| The system must allow user to add/delete product | x |  |
| The system must allow user to add/delete order | x |  |
| The system must allow user to add/delete customer | x |  |
| The system must allow user to view their account information | x |  |
| The system must allow user to log out | x |  |
| In employee view screen, employees must be sorted to their approriate role | x |  |
| Transition between tabs must not longer than 5 seconds |  | x |
| After logged in, username of user positioned next to login button |  | x |

1. System requirements analysis
   1. Use case diagrams



* 1. Use case specification

### 2.2.1 Login

|  |  |  |
| --- | --- | --- |
| Use case name | Login | |
| Description | This use case allow user to login account | |
| Trigger | User click Login | |
| Actors | Admin, Storage manager, salesman, accountant | |
| Preconditions | Account must exist in system | |
| Postconditions | User logged in system | |
| Flow of activities | Actor | System |
| 1. User click Login button 2. User enter username and password | * 1. System check if user account is already stored or have yet   2. System show up notification login success after user login succes |
| Exception conditions | 1.1 Inserted username and password wrong will return to login interface  1.2 Display error, forgot password. | |

### 2.2.2 Log out

|  |  |  |
| --- | --- | --- |
| Use case name | Logout | |
| Description | This use case allow user to logout account | |
| Trigger | User click logout button | |
| Actors | Admin, Storage manager, salesman, accountant | |
| Preconditions | User must logged in account | |
| Postconditions | User logout their account from system | |
| Flow of activities | Actor | System |
| 1. User click on the logout button 2. User return to mainpage | * 1. System provide logout button   2. System receive request Logout account from user   3. User account logged out |
| Exception conditions | 1.1 No showing Logout button if user not logged in | |

### 2.2.3 View account information

|  |  |  |
| --- | --- | --- |
| Use case name | View account information | |
| Description | This use case allow user to view account information | |
| Trigger | User click my account | |
| Actors | Admin, Storage manager, salesman, accountant | |
| Preconditions | Account must exist in system  User must logged in | |
| Postconditions | User view account information | |
| Flow of activities | Actor | System |
| 1. User click username 2. User click my account 3. User view account information | * 1. System navigate to account option screen   2. System navigate to account information screen |
| Exception conditions |  | |

### 2.2.4 Update account information

|  |  |  |
| --- | --- | --- |
| Use case name | Update account information | |
| Description | This use case allow user to update information | |
| Trigger | User click edit info button | |
| Actors | Admin, Storage manager, salesman, accountant | |
| Preconditions | Account must exist in system  User must logged in | |
| Postconditions | User update account information | |
| Flow of activities | Actor | System |
| 1. User click username 2. User click my account 3. User click edit info button 4. User update account infomation | * 1. System navigate to account option screen   2. System navigate to account information screen   3. System navigate to account edit screen   4. System stored new information in database |
| Exception conditions | * 1. Account information is not valid   2. Added information is duplicate | |

### 2.2.5 Add product

|  |  |  |
| --- | --- | --- |
| Use case name | Add product | |
| Description | This use case allow user to add products | |
| Trigger | User click add product in product tab | |
| Actors | Admin,storage manager | |
| Preconditions | User must log in to system | |
| Postconditions | Product added to system database | |
| Flow of activities | Actor | System |
| 1. User click add product 2. User enter product information | * 1. System navigate to add product screen   2. System store product in database |
| Exception conditions | 1.1 Product information not valid  1.2 Added product is duplicate | |

### 2.2.6 Delete product

|  |  |  |
| --- | --- | --- |
| Use case name | Delete product | |
| Description | This use case allow user to delete existing products | |
| Trigger | User click delete product in view product screen | |
| Actors | Admin,storage manager | |
| Preconditions | User must log in to system  There must be a product(s) in product list | |
| Postconditions | Product deleted in system database | |
| Flow of activities | Actor | System |
| 1. User click view product tab 2. User click delete in action bar | * 1. System navigate to view product screen   2. System delete product in database |
| Exception conditions | 1.1 | |

### 2.2.7 Edit product

|  |  |  |
| --- | --- | --- |
| Use case name | Edit product | |
| Description | This use case allow user to update product information | |
| Trigger | User click edit product in in view product screen | |
| Actors | Admin,storage manager | |
| Preconditions | User must log in to system  There must be a product(s) in product list | |
| Postconditions | Product information updated and stored in product database | |
| Flow of activities | Actor | System |
| 1. User click edit in view product screen 2. User change product information | * 1. System navigate to view product screen   2. System store product information in database |
| Exception conditions | 1.1 Product information not valid  1.2 Product information is duplicate | |

### 2.2.8 Add order

|  |  |  |
| --- | --- | --- |
| Use case name | Add order | |
| Description | This use case allow user to add order | |
| Trigger | User click add order in order tab | |
| Actors | Admin,salesman | |
| Preconditions | User must log in to system  Customer must exist in customer database  Product must exist in product database | |
| Postconditions | Order added in system database | |
| Flow of activities | Actor | System |
| 1. User click add order 2. User enter order information | * 1. System navigate to add order screen   2. System store order in database |
| Exception conditions | 1.1 Order information not valid  1.2 Added order is duplicate  1.3 Product out of stock  1.4 Customer not exist in database | |

### 2.2.9 Delete order

|  |  |  |
| --- | --- | --- |
| Use case name | Delete order | |
| Description | This use case allow user to delete existing order | |
| Trigger | User click delete order in order view screen | |
| Actors | Admin,salesman | |
| Preconditions | User must log in to system  There must be a order(s) in order list | |
| Postconditions | Order deleted in system database | |
| Flow of activities | Actor | System |
| 1. User click order view screen 2. User click delete in action bar | * 1. System navigate to order view screen   2. System delete order in database |
| Exception conditions | 1.1 | |

### 2.2.10 Edit order

|  |  |  |
| --- | --- | --- |
| Use case name | Edit order | |
| Description | This use case allow user to update existing orders | |
| Trigger | User click edit order in view order screen | |
| Actors | Admin,salesman | |
| Preconditions | User must log in to system  There must be a order(s) in order list | |
| Postconditions | Order information updated and stored in system database | |
| Flow of activities | Actor | System |
| 1. User click edit order in order view screen 2. User change order information | * 1. System navigate to add product screen   2. System store order information in database |
| Exception conditions | 1.1 Order information not valid  1.2 Order information is duplicate | |

### 2.2.11 Add employee

|  |  |  |
| --- | --- | --- |
| Use case name | Add employee | |
| Description | This use case allow user add employees | |
| Trigger | User click add employee in employee tab | |
| Actors | Admin | |
| Preconditions | User must log in to system  User privellege must be admin | |
| Postconditions | Employee added in system database | |
| Flow of activities | Actor | System |
| 1. User click add employee 2. User enter employee information | * 1. System navigate to add employee screen   2. System store product in database |
| Exception conditions | 1.1 Employee information not valid  1.2 Added employee is duplicate | |

### 2.2.12 Delete employee

|  |  |  |
| --- | --- | --- |
| Use case name | Delete employee | |
| Description | This use case allow user to delete existing products | |
| Trigger | User click delete product in product tab | |
| Actors | Admin | |
| Preconditions | User must log in to system  User privellege must be admin  There must be an employee(s) in employee list | |
| Postconditions | Employee deleted from system database | |
| Flow of activities | Actor | System |
| 1. User click delete product 2. User enter product information | * 1. System navigate to add product screen   2. System store product in database |
| Exception conditions | 1.1 Product information not valid, return error  1.2 Product added is duplicate, return error screen | |

### 2.2.13 Edit employee salary

|  |  |  |
| --- | --- | --- |
| Use case name | Edit employee salary | |
| Description | This use case allow user to update employee salary | |
| Trigger | User click edit information in employee view screen | |
| Actors | Admin | |
| Preconditions | User must log in to system  User must have admin privellege | |
| Postconditions | Employee salary updated in system database | |
| Flow of activities | Actor | System |
| 1. User click edit information in employee view screen 2. User enter new salary | * 1. System navigate to employee edit information screen   2. System store new salary data in database |
| Exception conditions | 1.1 Salary datatype not valid | |

### 2.2.14 Add customer

|  |  |  |
| --- | --- | --- |
| Use case name | Add customer | |
| Description | This use case allow user to add customer | |
| Trigger | User click add customer in customer tab | |
| Actors | Admin,salesman | |
| Preconditions | User must log in to system | |
| Postconditions | Customer added to system database | |
| Flow of activities | Actor | System |
| 1. User click add customer 2. User enter customer information | * 1. System navigate to add customer screen   2. System store customer in database |
| Exception conditions | 1.1 Customer information not valid  1.2 Added customer is duplicate | |

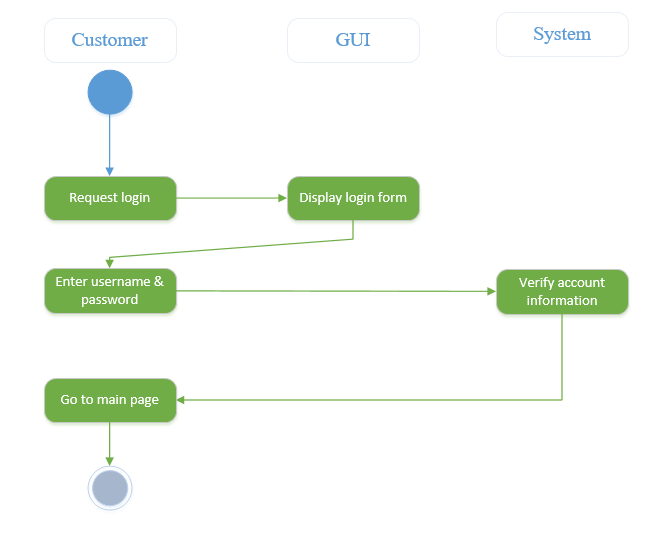
### 2.2.15 Delete customer

|  |  |  |
| --- | --- | --- |
| Use case name | Delete customer | |
| Description | This use case allow user to delete existing customer | |
| Trigger | User click delete customer in customer view screen | |
| Actors | Admin,salesman | |
| Preconditions | User must log in to system  There must be a customer (s) in product list | |
| Postconditions | Customer deleted from system database | |
| Flow of activities | Actor | System |
| 1. User click customer view screen 2. User click delete in action bar | * 1. System navigate to customer view screen   2. System delete customer in database |
| Exception conditions | 1.1 | |

### 2.2.16 Edit customer

|  |  |  |
| --- | --- | --- |
| Use case name | Edit customer | |
| Description | This use case allow user to update existing customer | |
| Trigger | User click edit customer in customer view screen | |
| Actors | Admin,salesman | |
| Preconditions | User must log in to system  There must be a customer(s) in customer list | |
| Postconditions | Customer information updated | |
| Flow of activities | Actor | System |
| 1. User click edit 2. User enter product information | * 1. System navigate to add product screen   2. System store product in database |
| Exception conditions | 1.1 Product information not valid, return error  1.2 Product added is duplicate, return error screen | |

* 1. Activity diagrams



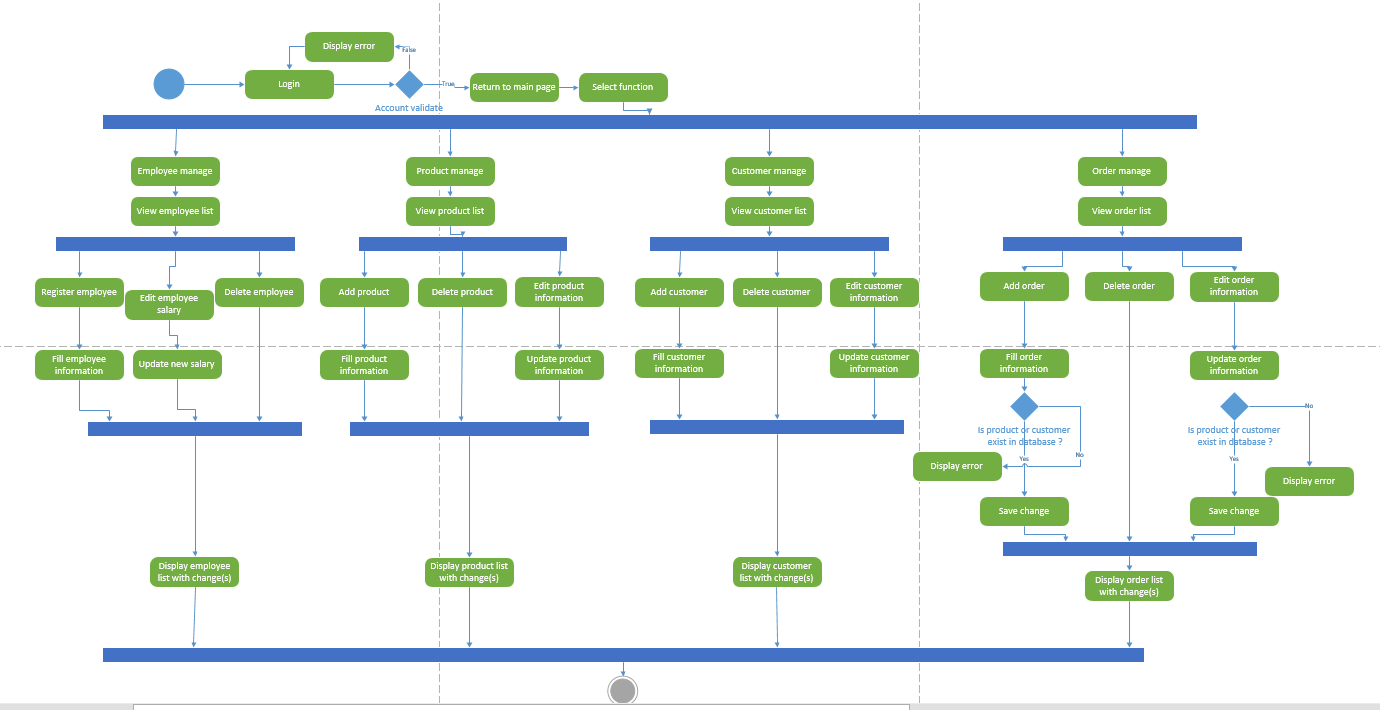


Figure 2. 1 Activity diagram of Computer shop management with admin privellege

* 1. Sequence diagrams

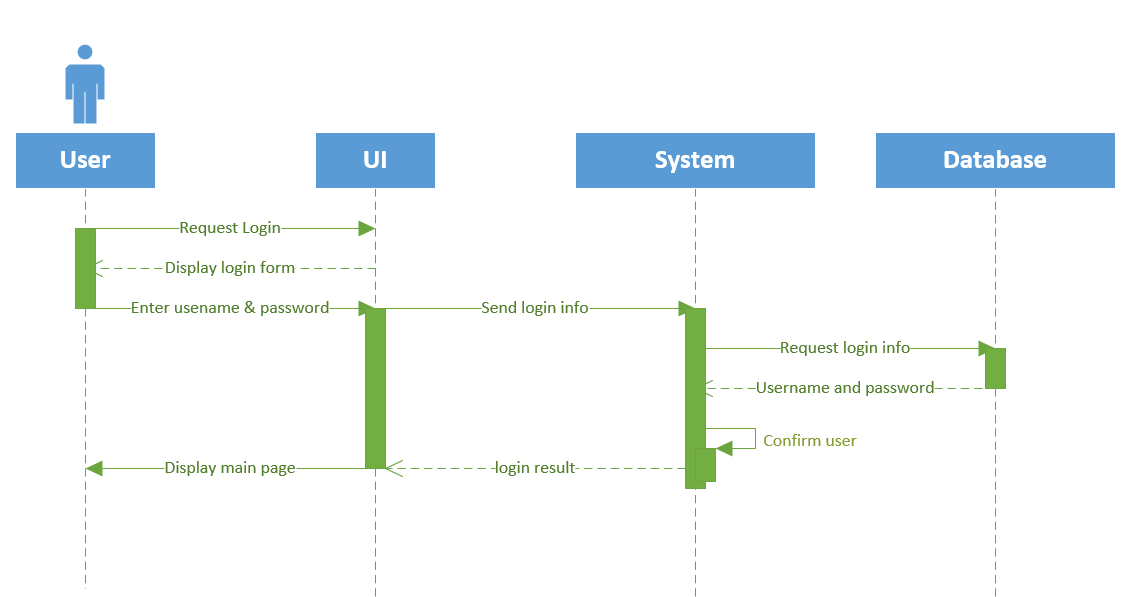
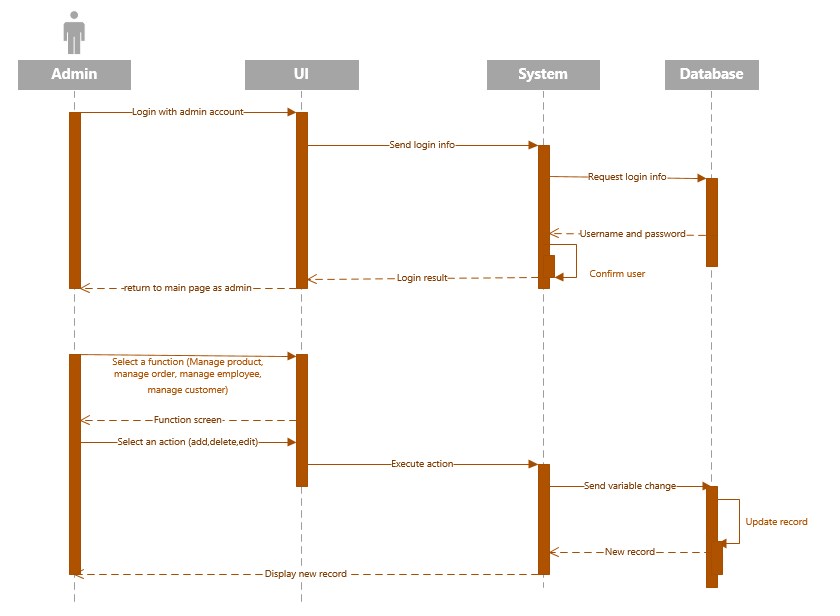


Figure 2. 2 Sequence diagram of admin interaction with functions process

Figure 2. 3 Sequence diagram of login process

1. System requirements desgin
   1. Class diagram

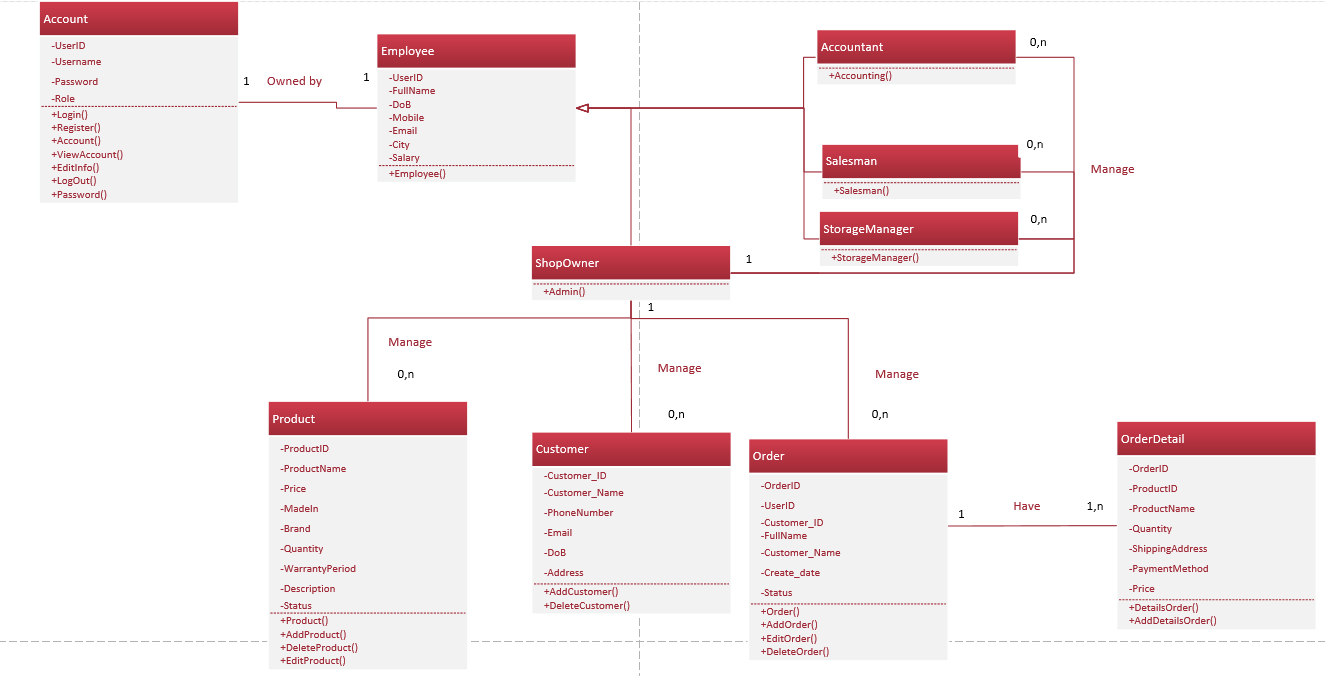


Figure 3. 1 Class diagram

* 1. Entity relationship diagram

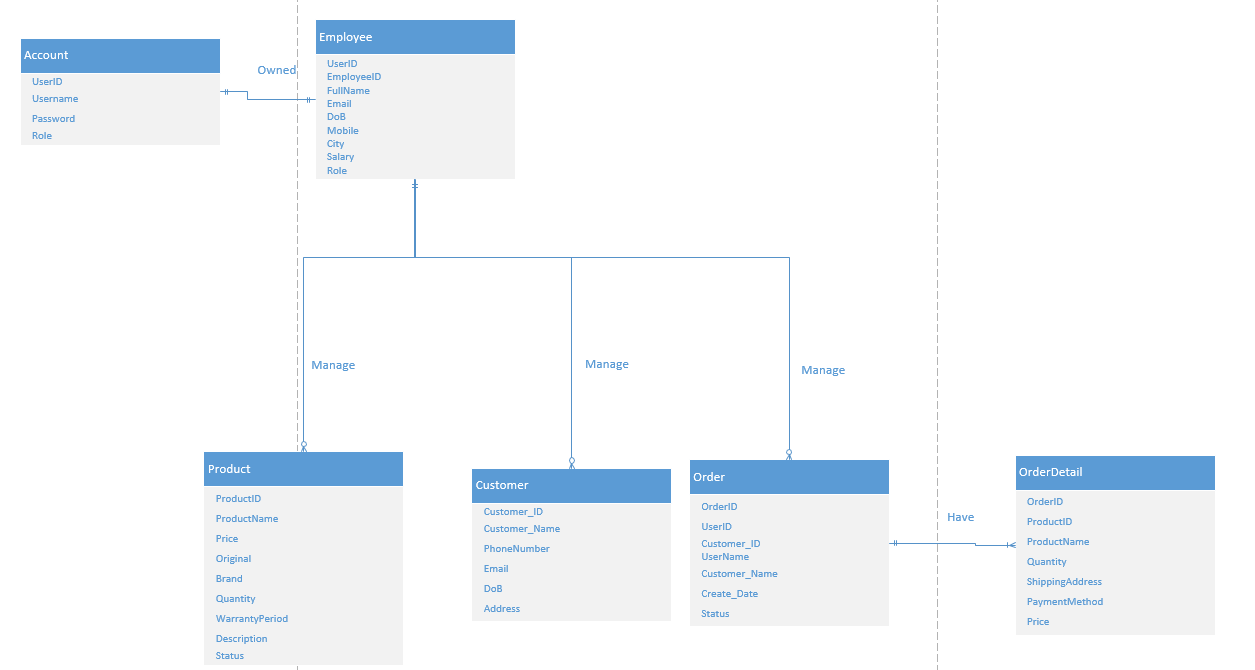


Figure 3. 2 Entity relationship diagram

* 1. Relational data model

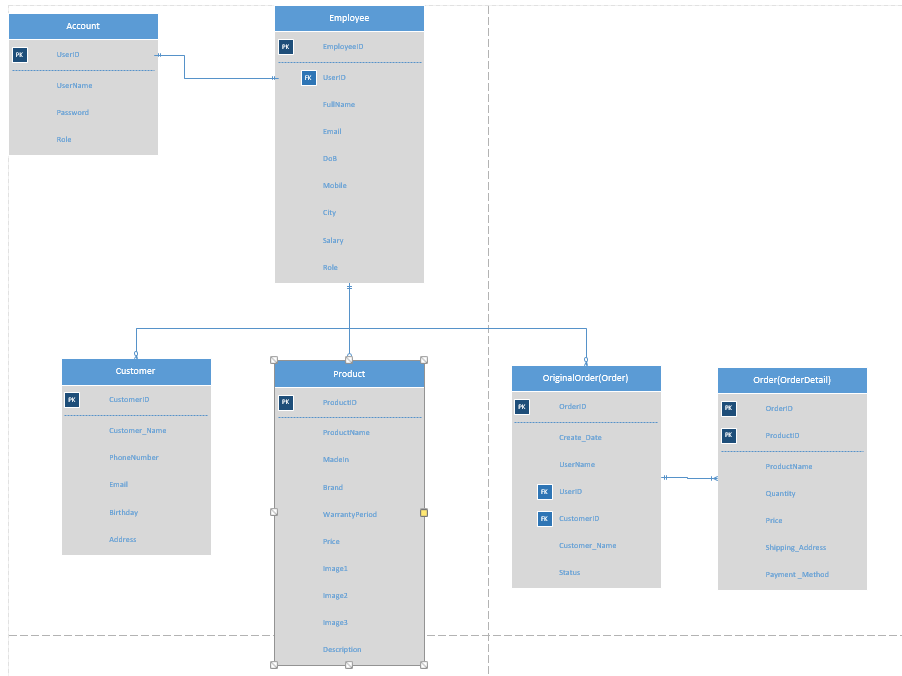


Figure 3. 3 Relational model diagram

3.4 Database design

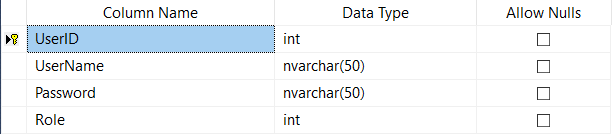


Figure 3. 4 Account database

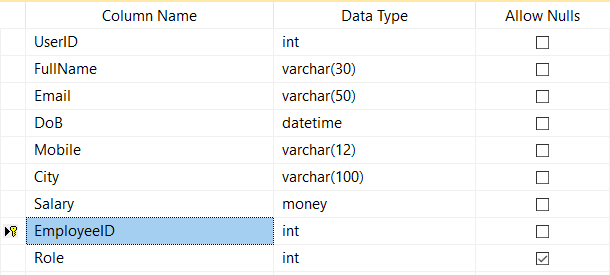


Figure 3. 5 Employee database

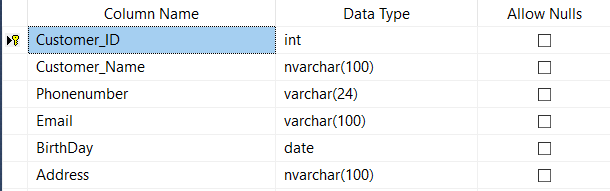


Figure 3. 6 Customer database

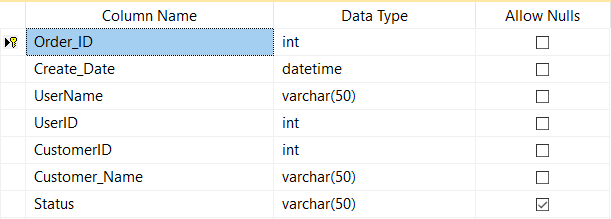


Figure 3. 7 OrderOriginal database

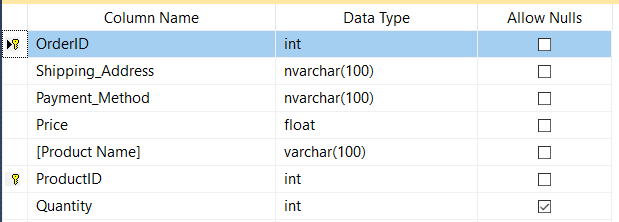


Figure 3. 8 Order database

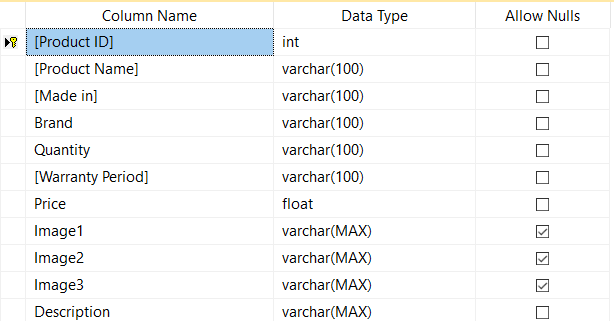


Figure 3. 9 Product database

1. System UI

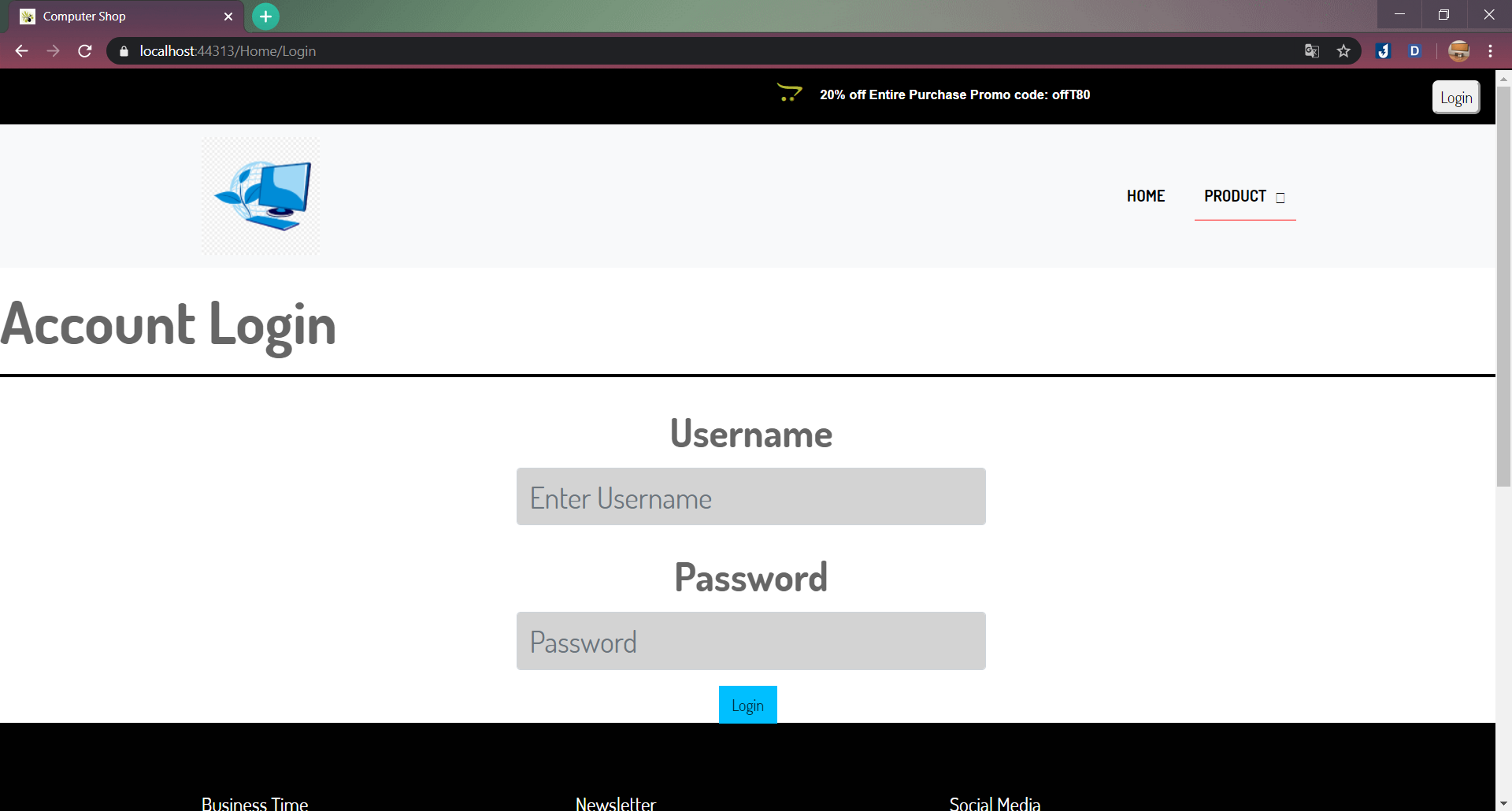
****

Figure 4. 1 Login Interface

* The picture above show the login page of the website, its require user login to interact with the page. There are 4 types of account on this website :
* Shop Owner (Admin)
  + Username login for Shop Owner :
* Username : vinhnk3100
* Password : 123456
* Storage Manager
* Salesman
* Accountant
* After login the website will show like this in the picture below :

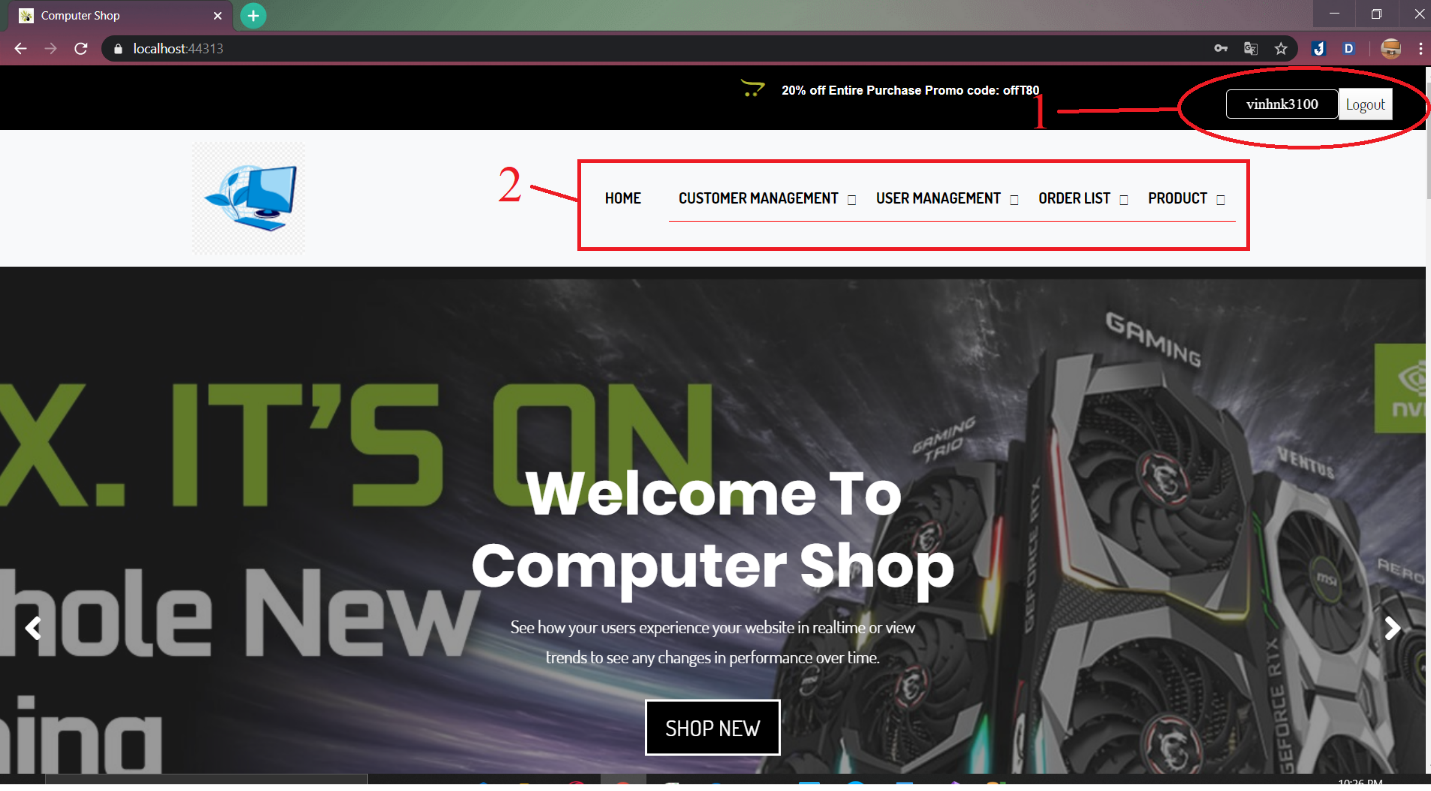


Figure 4. 2 Main Page

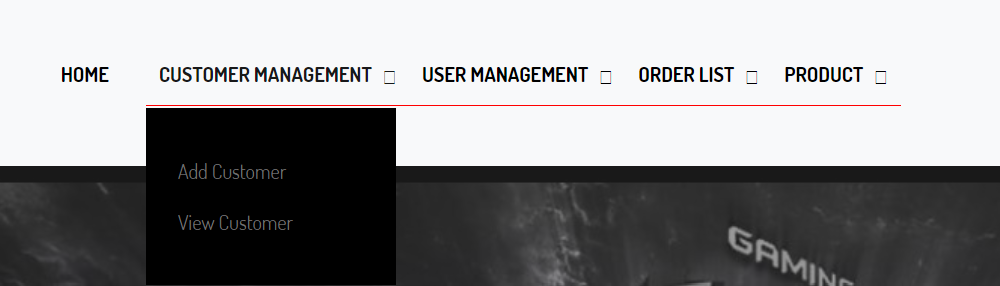
* 1 : This show our username
* 2 : This show each function that user can interact and manage in the website.
* This account user is an Admin so its can interact and manage each function show in “2” by clicking it will redirect to another page. Example from the picture below : 

Figure 4. 3 Customer Management function

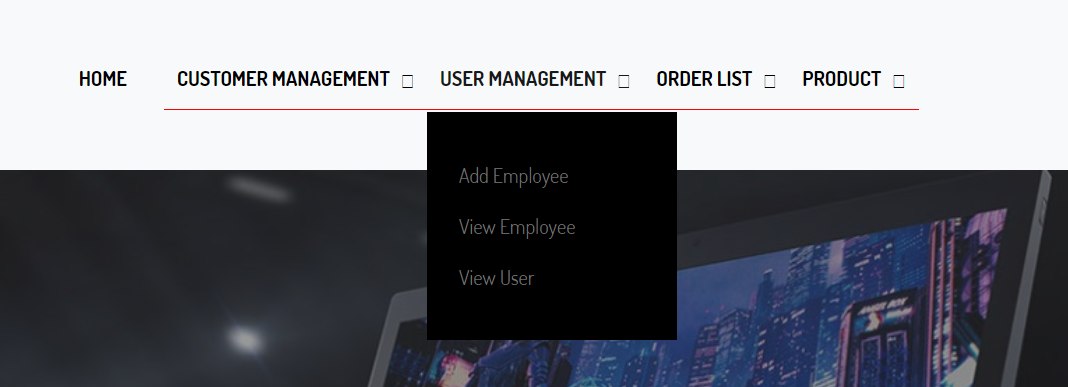


Figure 4. 4 User Management function

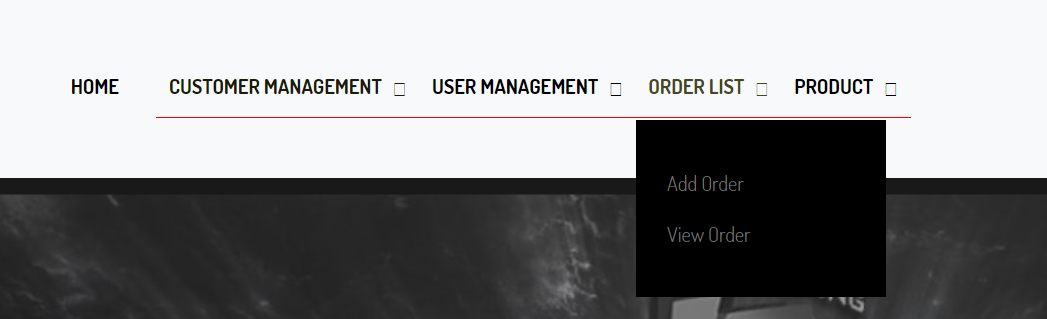


Figure 4. 5 Order function

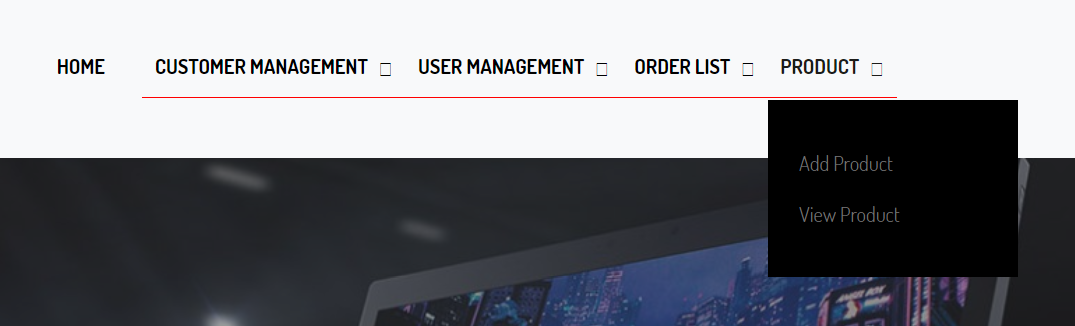


Figure 4. 6 Product function

* By clicking on each function it will dropdown more function for the user to interact it. However, each account types will have limit of function, because this account type is admin so it can interact with many function on the website to manage the website.

1. Github

Github link : https://github.com/btlCNPM1920/Assignment1920.git