

**WebSite Watch Sales**



Members

1. Trương Thị Thanh Xuân – CE140149
2. Trương Nhật Nam – CE140370
3. Nguyễn Quốc An – CE140024

Mentor Luong Hoang Huong

Table of Contents

[I. PROBLEM DENIFITION 4](#_Toc46235596)

[II. USER REQUIREMENTS 5](#_Toc46235597)

[*1.* *Input:* 5](#_Toc46235598)

[*2.* *Output:* 5](#_Toc46235599)

[*3.* *List of function:* 5](#_Toc46235600)

[III. SYSTEM REQUIREMENTS 6](#_Toc46235601)

[1. *System requirement*: 6](#_Toc46235602)

[*2.* *Required Software:* 7](#_Toc46235603)

[IV. JSP 9](#_Toc46235604)

[*1.* *What is JSP?* 9](#_Toc46235605)

[*2.* *Advantages of JSP over Servlet* 10](#_Toc46235606)

[*3.* *The Lifecycle of a JSP Page* 10](#_Toc46235607)

[V. MVC 11](#_Toc46235608)

[*1.* *What is MVC?* 11](#_Toc46235609)

[*2.* *The advantages of MVC* 13](#_Toc46235610)

[VI. DETAIL 14](#_Toc46235611)

[*1.* *Date of project plan* 14](#_Toc46235612)

[*2.* *Project’s Vision/Objective* 14](#_Toc46235613)

[*3.* *Project Initiation/Requirement Document* 14](#_Toc46235614)

[*4.* *Plan* 14](#_Toc46235615)

[VII. ROLES 15](#_Toc46235616)

[*1.* *Table roles* 15](#_Toc46235617)

[*2.* *Gantt* 16](#_Toc46235618)

[VIII. SCHEDULE 18](#_Toc46235619)

[IX. DESIGN PATTERN 19](#_Toc46235620)

[*1.* *Data base Diagram* 19](#_Toc46235621)

[*2.* *Class Diagram* 20](#_Toc46235622)

[21](#_Toc46235623)

[*3.* *Activity diagram* 21](#_Toc46235624)

[*a.* *User* 22](#_Toc46235625)

[*b.* *Admin* 22](#_Toc46235626)

[*4.* *DFD* 24](#_Toc46235627)

[*a.* *Lv0* 24](#_Toc46235628)

[b. Lv1 25](#_Toc46235629)

[*c.* *Lv2* 27](#_Toc46235630)

[*d.* *Lv3* 30](#_Toc46235631)

[*e.* *Lv4* 32](#_Toc46235632)

[*5.* *Usercase diagram* 33](#_Toc46235633)

[X. PROGRAM INTEFACE 35](#_Toc46235634)

# PROBLEM DENIFITION

Watches are an indispensable piece of jewelry for every person. People wearing beautiful watches show their class elegance and elegance. Catching this trend, the group 8 decided to create a website specializing in the sale of men's and women's watches. My group chose the topic of selling watches because:

* The clock has many diverse models
* Many brands are easy to choose,
* When it comes to making it easy for people to like and look to the website to see.
* The website that sells watches has functions that make it easy to find and order products.

*Disadvantage:*

* The group is not very knowledgeable about watches in the world.
* Update prices will not be accurate.
* Not much functional design for the site.
* There are not too many watches so there will be some watches that people find without.

# USER REQUIREMENTS

# *Input:*

* A website that sells watches should have a product listing, product images, product brand names, product prices, login and a shopping cart and payment function to purchase products.
* In addition, it is necessary to have the login function, add new products and update , search function products.

# *Output:*

* Information will be saved on the database when the product is completed.

# *List of function:*

* The user accesses the site
* View and search for product available on the site
* Select product and add to cart
* Login to pay for products
* Pay

# SYSTEM REQUIREMENTS

1. *System requirement*:

|  |
| --- |
| Minimum Hardware Configurations |
| Microsoft Windows Vista SP1/Windows 7 Professional:  Processor: 800MHz Intel Pentium III or equivalent  Memory: 512 MB  Disk space: 750 MB of free disk space  Ubuntu 9.10:  Processor: 800MHz Intel Pentium III or equivalent  Memory: 512 MB  Disk space: 650 MB of free disk space  Macintosh OS X 10.7 Intel:  Processor: Dual-Core Intel  Memory: 2 GB  Disk space: 650 MB of free disk space |

|  |
| --- |
| Recommended Hardware Configurations |
| Microsoft Windows 7 Professional/Windows 8/Windows 8.2:  Processor: Intel Core i5 or equivalent  Memory: 2 GB (32-bit), 4 GB (64-bit)  Disk space: 1.5 GB of free disk space  Ubuntu 15.04:  Processor: Intel Core i5 or equivalent  Memory: 2 GB (32-bit), 4 GB (64-bit)  Disk space: 1.5 GB of free disk space  OS X 10.10 Intel:  Processor: Dual-Core Intel  Memory: 4 GB  Disk space: 1.5 GB of free disk space |

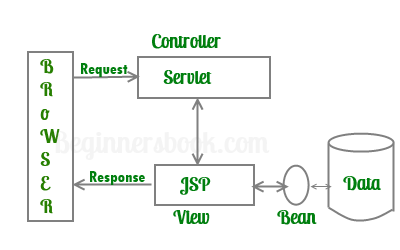
1. *Required Software:*

* NetBeans IDE runs on the Java SE Development Kit (JDK) which consists of the Java Runtime Environment and developer tools for compiling, debugging, and running applications written in the Java language.
* The tested JDK for this release is JDK 8u101 for Windows, Linux, and OS X. The 8.2 version of the IDE cannot be installed or run on the JDK older than JDK 8.
* Note:
* The PHP and C/C++ NetBeans bundles only require the Java Runtime Environment (JRE) 8 to be installed and run.
* Java features in the IDE and JavaFX 8 features require JDK 8.
* Download Tomcat for webserver, XAMPP for database.

# JSP

1. *What is JSP?*

* JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc.
* A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.



*----- 4. Introduce JSP -----*

1. *Advantages of JSP over Servlet*

* Extension to Servlet: JSP technology is the extension to Servlet technology. We can use all the features of the Servlet in JSP. In addition to, we can use implicit objects, predefined tags, expression language and Custom tags in JSP, that makes JSP development easy.
* Easy to maintain: JSP can be easily managed because we can easily separate our business logic with presentation logic. In Servlet technology, we mix our business logic with the presentation logic.
* Fast Development: No need to recompile and redeploy. If JSP page is modified, we don't need to recompile and redeploy the project. The Servlet code needs to be updated and recompiled if we have to change the look and feel of the application.
* Less code than Servlet: In JSP, we can use many tags such as action tags, JSTL, custom tags, etc. that reduces the code. Moreover, we can use EL, implicit objects, etc.

1. *The Lifecycle of a JSP Page*

* Translation of JSP Page
* Compilation of JSP Page
* Classloading (the classloader loads class file)
* Instantiation (Object of the Generated Servlet is created).
* Initialization (the container invokes jspInit() method).
* Request processing (the container invokes \_jspService() method).
* Destroy (the container invokes jspDestroy() method).
* As depicted in the above diagram, JSP page is translated into Servlet by the help of JSP translator. The JSP translator is a part of the web server which is responsible for translating the JSP page into Servlet. After that, Servlet page is compiled by the compiler and gets converted into the class file. Moreover, all the processes that happen in Servlet are performed on JSP later like initialization, committing response to the browser and destroy.

# MVC

1. *What is MVC?*

* MVC is an architecture that separates business logic, presentation and data. In MVC, M stands for Model, V stands for View, C stands for controller.
* MVC is a systematic way to use the application where the flow starts from the view layer, where the request is raised and processed in controller layer and sent to model layer to insert data and get back the success or failure message.

**Model Layer:**

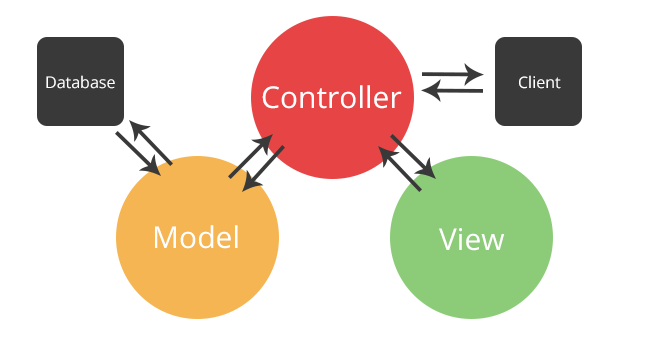
* This is the data layer which consists of the business logic of the system.
* It consists of all the data of the application
* It also represents the state of the application.
* It consists of classes which have the connection to the database.
* The controller connects with model and fetches the data and sends to the view layer.
* The model connects with the database as well and stores the data into a database which is connected to it.

**View Layer:**

* This is a presentation layer.
* It consists of HTML, JSP, etc. into it.
* It normally presents the UI of the application.
* It is used to display the data which is fetched from the controller which in turn fetching data from model layer classes.
* This view layer shows the data on UI of the application.

**Controller Layer:**

* It acts as an interface between View and Model.
* It intercepts all the requests which are coming from the view layer.
* It receives the requests from the view layer and processes the requests and does the necessary validation for the request.



*5. MVC Model*

1. *The advantages of MVC*

* Easy to maintain
* Easy to extend
* Easy to test
* Navigation control is centralized

# DETAIL

# *Date of project plan*

* Start: 02/06/2020
* Finish: 25/07/2020

# *Project’s Vision/Objective*

* Create a website that sells watches so people can view and buy products online.
* Products and prices will be updated regularly so people can visit the website to search and reference products new.

# *Project Initiation/Requirement Document*

* There are features like login, update, add, delete, add to cart and check out products.
* Detailed information about each product.
* There are comments and reviews from customers.
* Pay for the product that has been purchased and delivered to the user

# *Plan*

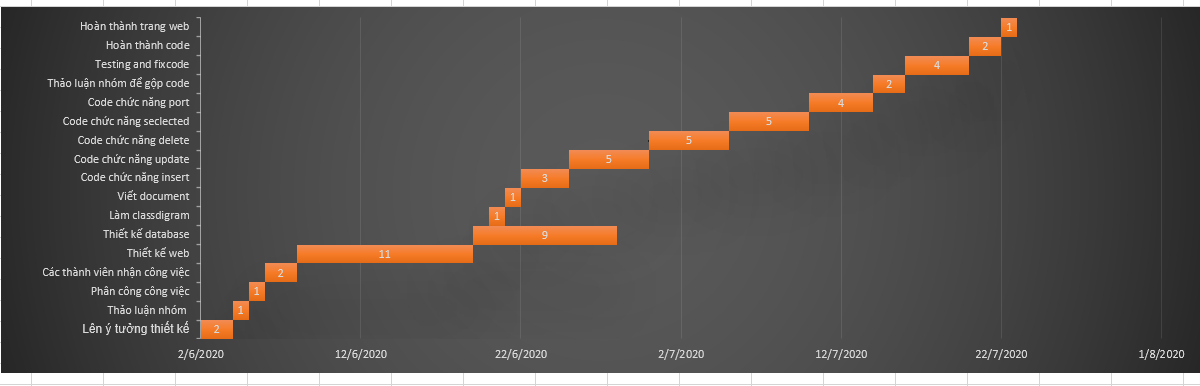
* Check activity: Test code, test program after done.
* Demo activity: Demo every day and at least 1 time a day.
* Review activity: review 4 times in class.
* Backup and recovery: 2 times a week.

# ROLES

1. *Table roles*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task No | Task Description | Expected Completion Date | Expected Time Needed(hrs) | Members in charge | Check |
| 1 | Lên ý tưởng thiết kế web | 2/06/2020 | 1 | Xuân, Nam , An | Done |
| 2 | Thảo luận nhóm chốt ý tưởng | 4/06/2020 | 1 | Xuân, Nam ,An | Done |
| 3 | Phân công công việc cho các thành viên | 5/06/2020 | 1 | Xuân | Done |
| 4 | Các thành viên trong nhóm nhận công việc | 6/06/2020 | 1 | Xuân, Nam , An | Done |
| 5 | Thiết kế web | 8/06/2020 | 148 | Xuân, Nam , An | Done |
| 6 | Thiết kế database | 19/06/2020 | 3 | Xuân, Nam ,An | Done |
| 7 | Làm classdiagram | 20/06/2020 | 4 | Xuân | Done |
| 8 | Viết document | 21/06/2020 | 4 | Xuân, Nam ,An | Done |
| 9 | Code chức năng insert product | 22/06/2020 | 3 | Nam | Done |
| 10 | Code chức năng insert user | 22/06/2020 |  | An | Done |
| 11 | Code chức năng insert bill | 22/06/2020 | 3 | Xuân | Done |
| 12 | Code chức năng update product | 25/06/2020 | 3 | Nam | Done |
| 13 | Code chức năng update user | 25/06/2020 | 3 | An | Done |
| 14 | Code chức năng update bill | 25/06/2020 | 3 | Xuân | Done |
| 15 | Code chức năng delete product | 30/06/2020 | 3 | Nam | Done |
| 16 | Code chức năng delete user | 30/06/2020 | 3 | An | Done |
| 17 | Code chức năng delete bill | 30/06/2020 | 3 | Xuân | Done |
| 18 | Code chức năng seclected product | 5/07/2020 | 3 | Nam | Done |
| 19 | Code chức năng seclected user | 5/07/2020 | 3 | An | Done |
| 20 | Code chức năng selected bill | 5/07/2020 | 3 | Xuân | Done |
| 21 | Code chức năng port product | 10/07/2020 | 3 | Nam | Done |
| 20 | Code chức năng port user | 10/07/2020 | 3 | An | Done |
| 21 | Code chức năng port bill | 10/07/2020 | 3 | Xuân | Done |
| 22 | Thảo luận nhóm để gộp code | 14/07/2020 | 4 | Xuân, Nam ,An | Done |
| 23 | Testing and fixcode | 16/07/2020 | 5 | Xuân, Nam ,An | Done |
| 24 | Hoàn thành code | 20/07/2020 | 1 | Xuân, Nam, An | Done |
| 25 | Hoàn thành trang web | 22/07/2020 | 2 |  | Done |
| 26 | Deadline | 25/07/2020 | 1 |  | Done |

1. *Gantt*

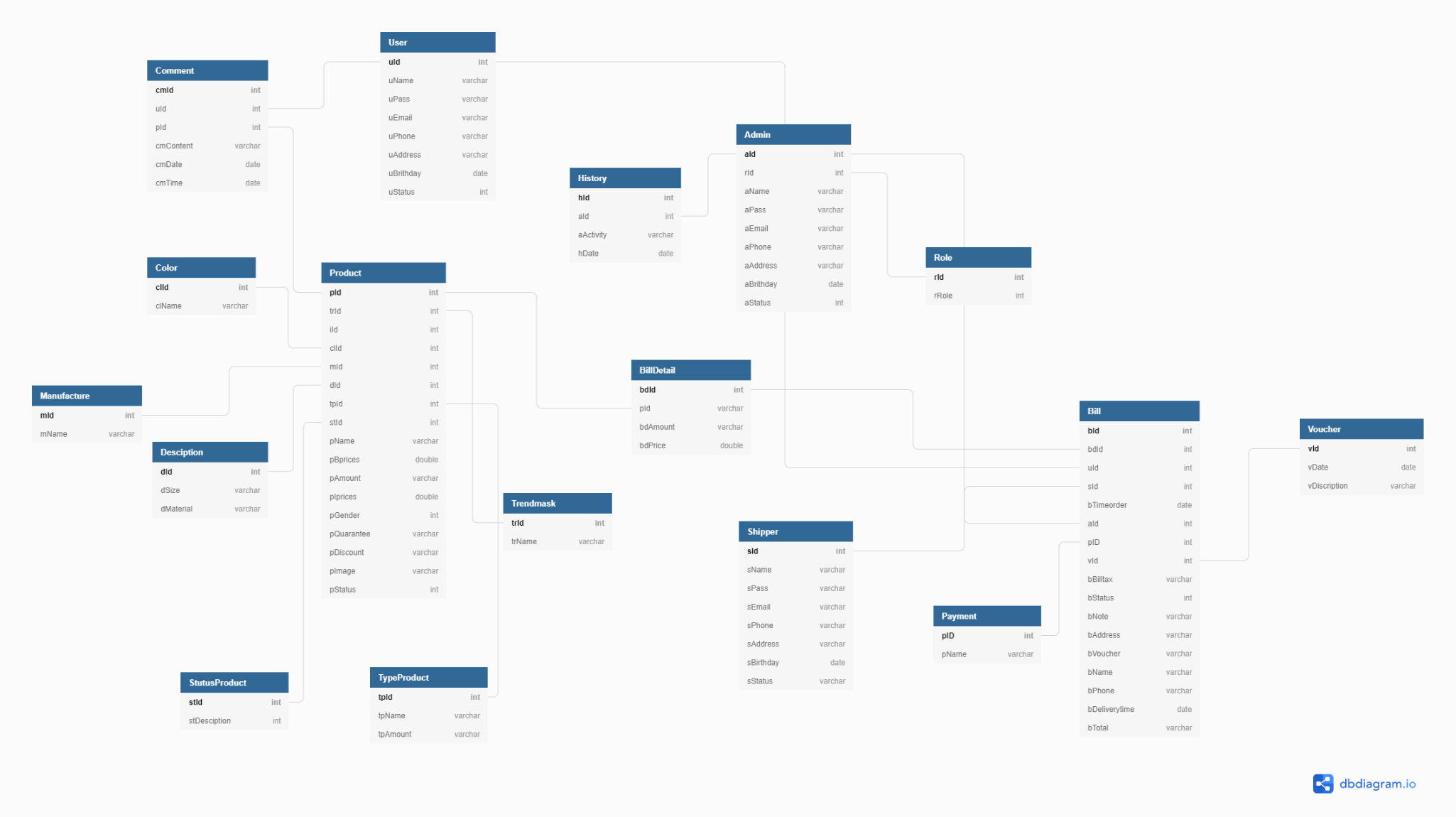


# SCHEDULE

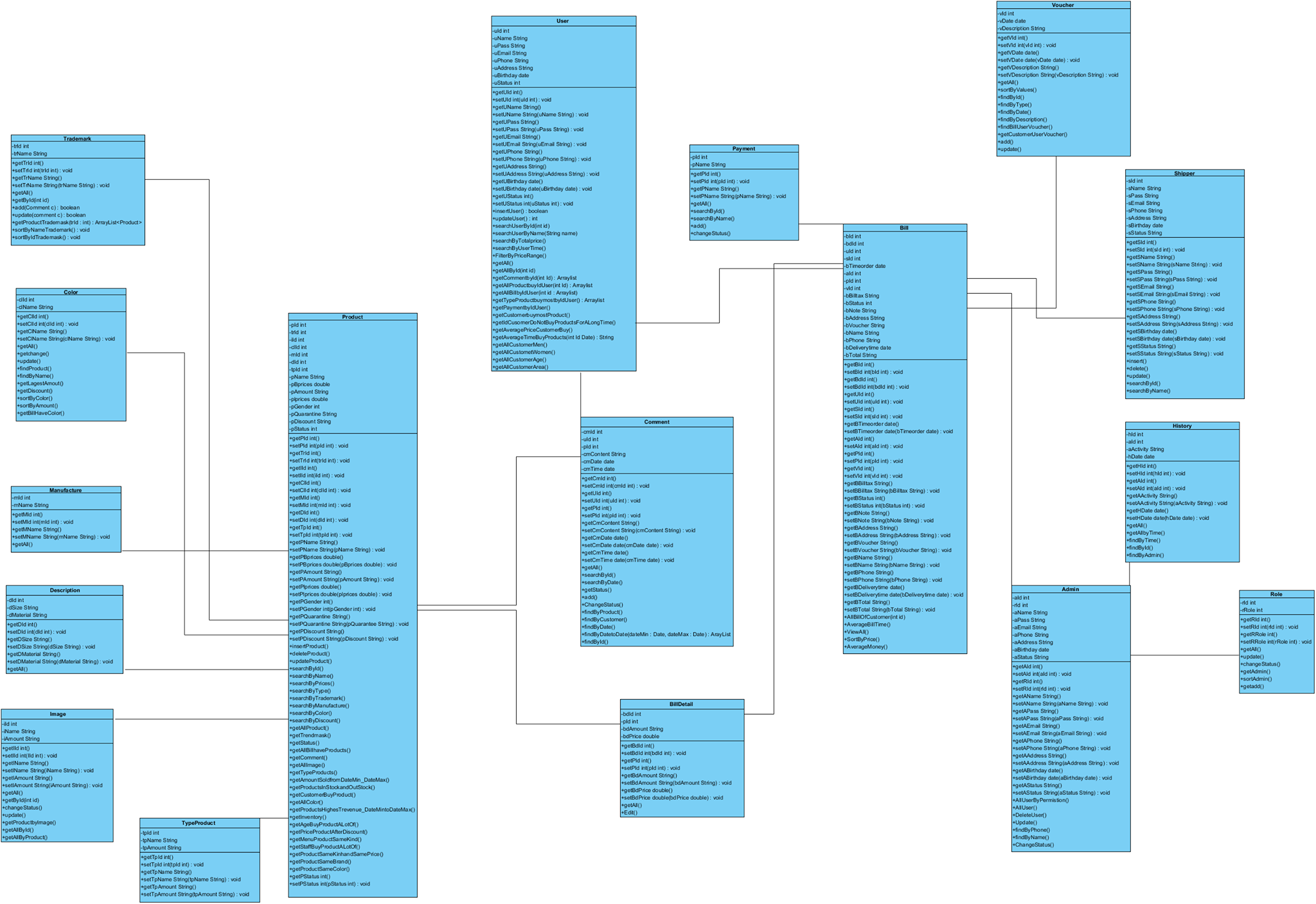
|  |  |  |
| --- | --- | --- |
| 1. Lịch họp dự kiến | | |
| Nội dung | Hình thức | Thời gian |
| Thảo luận về chức năng insert, update product, user, bill | Online | 20h |
| Thảo luận về delete và select product, user, bill, báo cáo tiến độ | Offline | 14h |
| Thảo luận về chức năng post select product, user, bill, báo cáo tiến độ | Offline | 14h |
| Thảo luận về thiết kế giao diện, chỉnh sửa lỗi | Offline | 14h |
| Chỉnh sửa document và làm powerpoint | Offline | 14h |

# DESIGN PATTERN

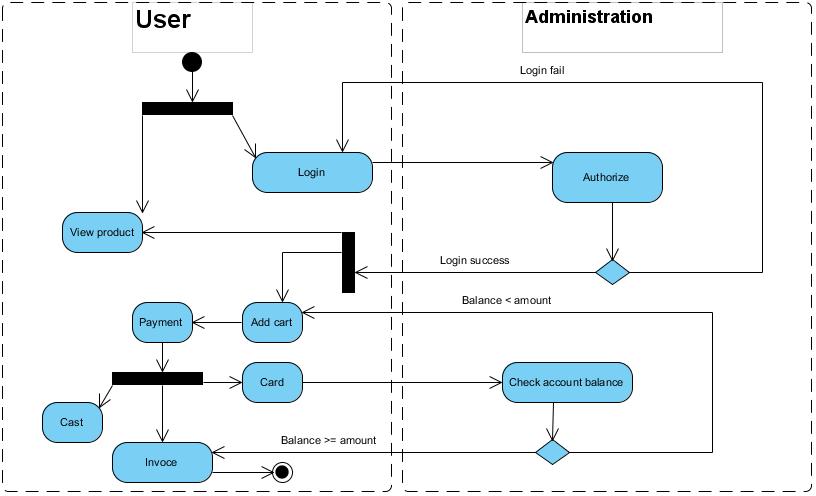
1. *Data base Diagram*

**

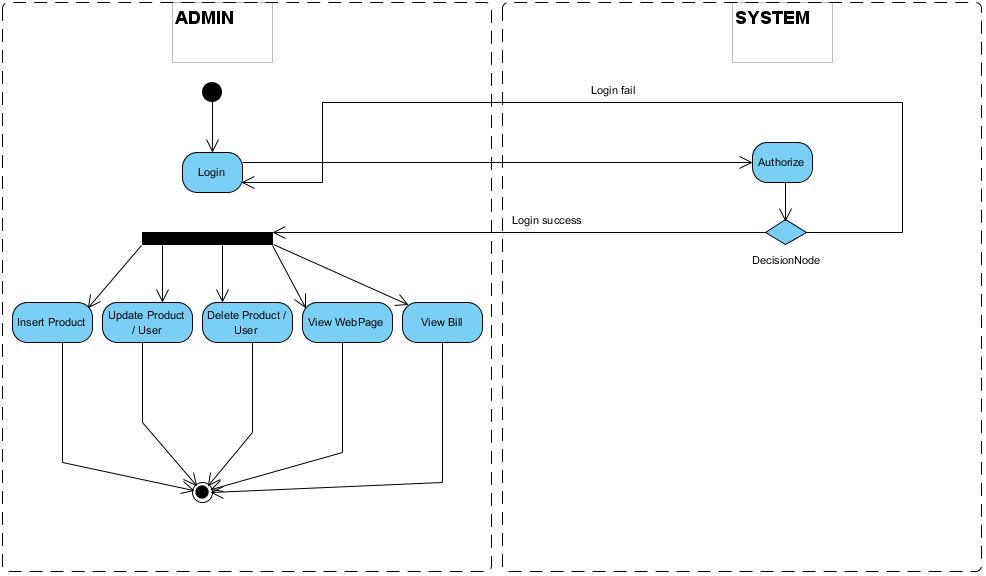
1. *Class Diagram*

**

1. *Activity diagram*
2. *User*



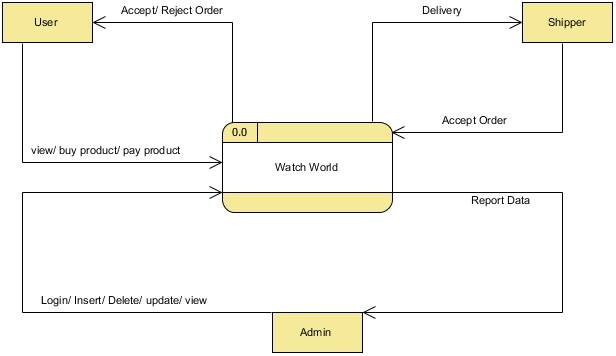
1. *Admin*



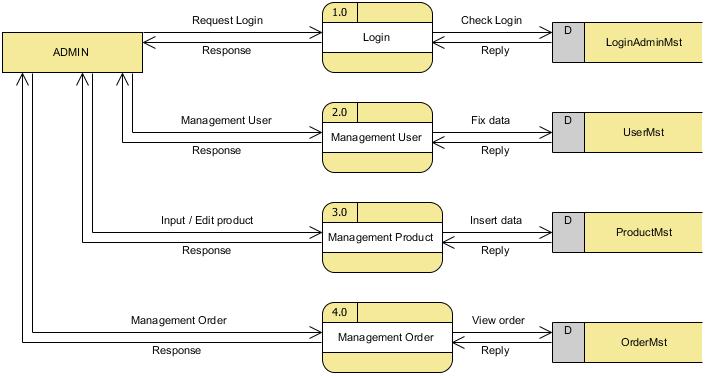
1. *DFD*

|  |  |
| --- | --- |
| Symbol | Description |
|  | **Data Flow:** Data flow are pipelines through the packets of information flow. |
|  | **Process:** A Process or task performed by the system. |
|  | **Entity:** Entity are object of the system. A source or destination data of a system. |
|  | **Data Store:** A place where data to be stored. |

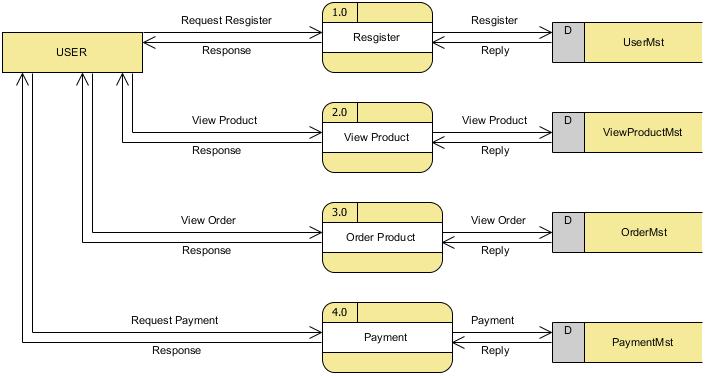
1. *Lv0*



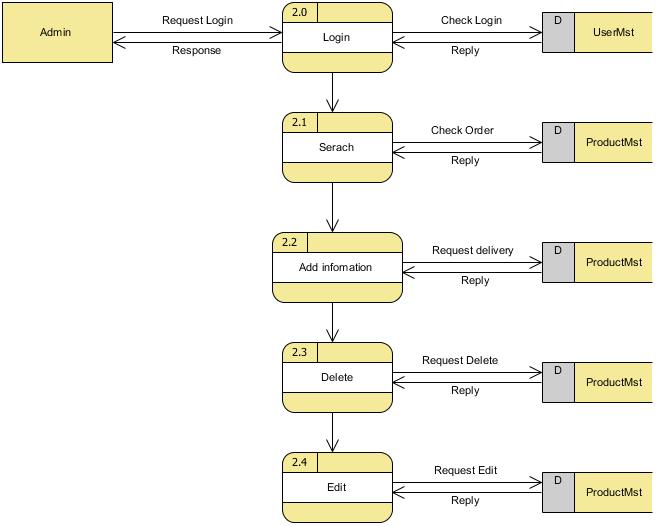
1. Lv1

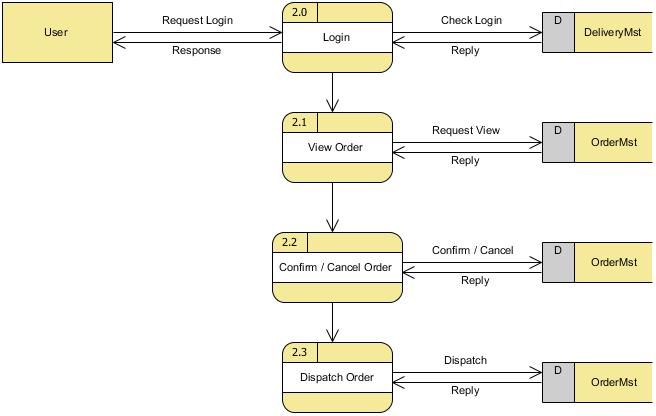


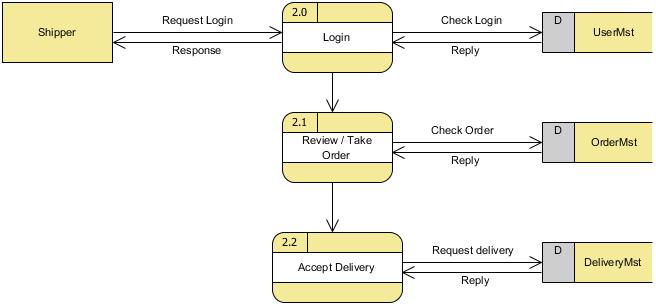




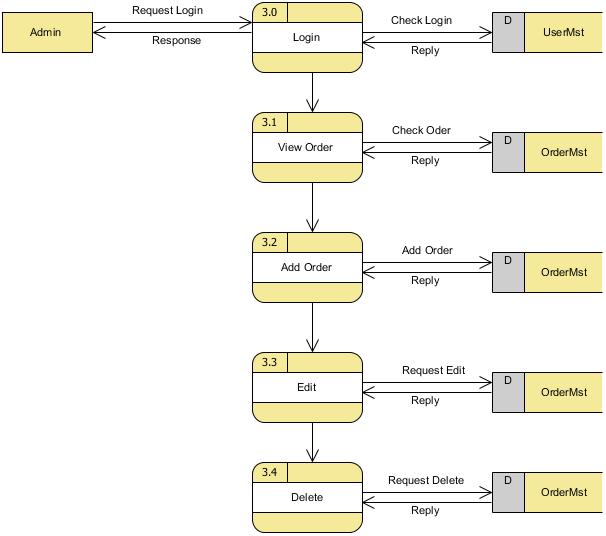
1. *Lv2*

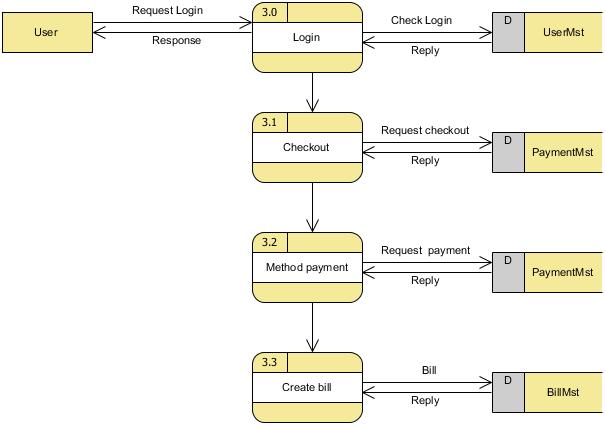




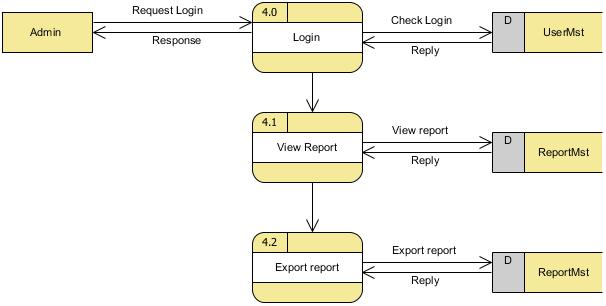


1. *Lv3*

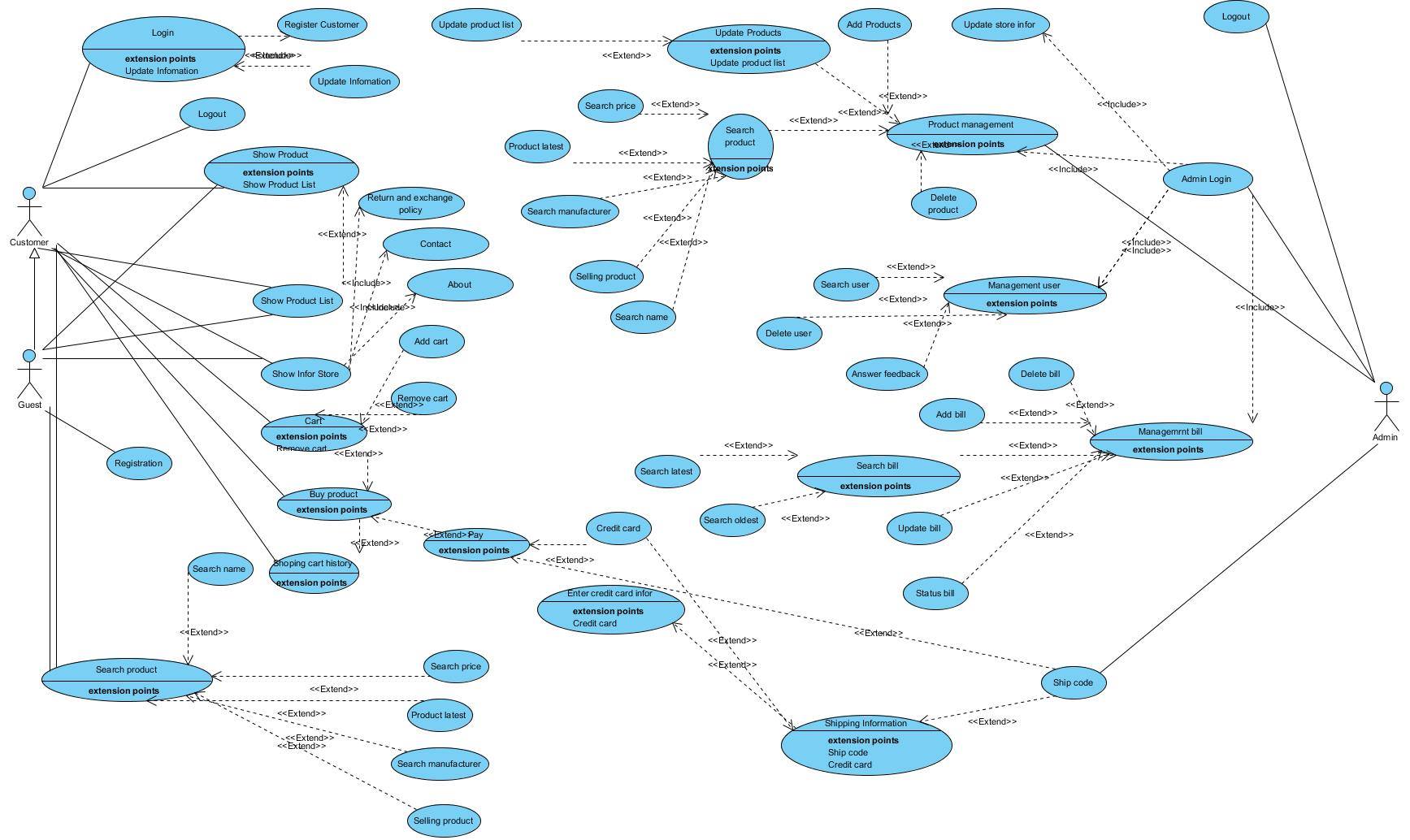




1. *Lv4*



1. *Usercase diagram*



# PROGRAM INTEFACE