***Software Requirements Specification***

for >**E-commerce Application** <

> Version 1.0 approved >

>Prepared by> Team Soft # 41

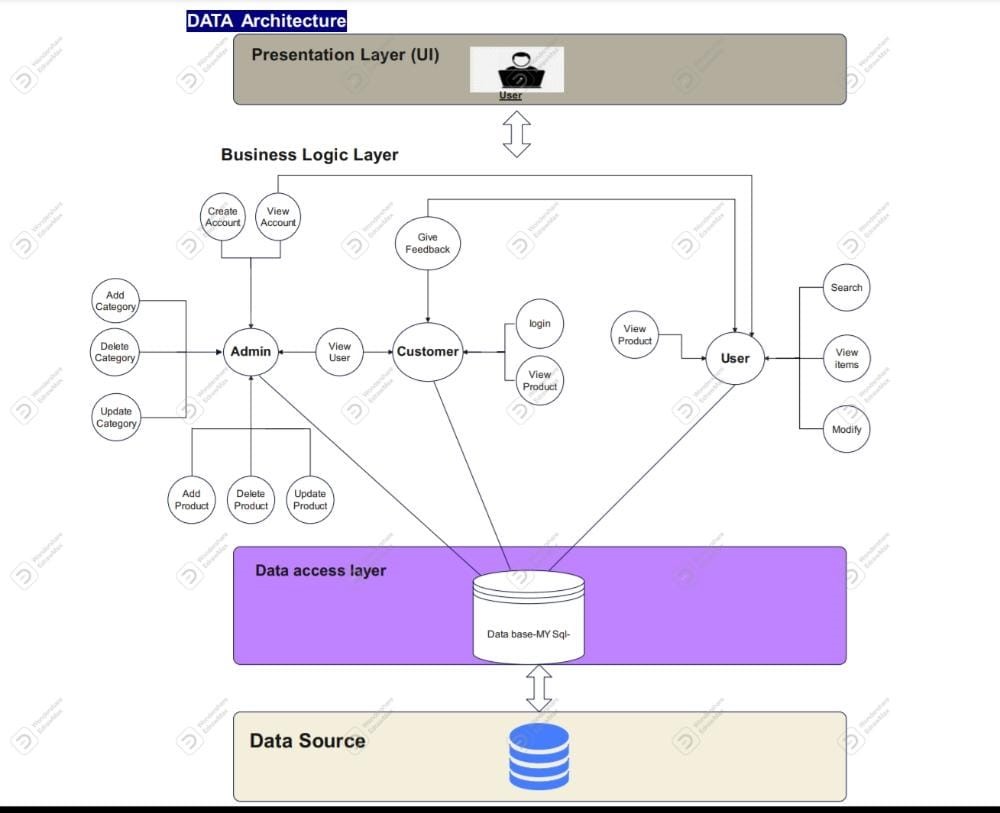
>8-05-2024<

**Scope**

E-commerce Application is software which helps people to buy things that they want .

Also it will help the users to know the paying details per day.

**System Architecture**



**Architecture Patterns: Layered Architecture**

|  |
| --- |
| **The presentation layer** : it have the user interface for user, admin and customer |
| **The business logic layer and Data access layer** : It contains function of each user in the system and handling function |
| **The database layer** : This is where all the data is stored. |

**Benefits of layered Architecture in the system:**

* Layers make reuse easier, dependencies between layers are minimized. A layer can easily be changed with a different implementation without a system destroy development and evolution can make easy and testing each component becomes more easy. It also increase reliability and availability
* ***Description***

E-commerce Application application helps people buy what they need: books, coffee cups, luggagetags, or mousepads. They can also pay and buy online and request orders. They can also log in and save all their information whenever they enter the site.

***System Requirement Specification***

**Functional Requirements**

**login**

|  |  |
| --- | --- |
| **1.login** | The system must allow the customer Sign in the system |
| **2.login** | The system must allow the customer change the password |
| **3.login** | The system must allow the customer log out in the system |
| **4.Login** | The system must allow the customer delete account from System |

**record**

|  |  |
| --- | --- |
| **1.record** | The system must records all details provided by user |
| **2.record** | The system must records a favorites products |

**Products**

|  |  |
| --- | --- |
| **1.products** | The system must allow the customer to **add** a new product in the basket |
| **2.products** | The system must allow the customer to **update** a new product in the basket |
| **3.products** | The system must allow the customer to **delete** a new product in the basket |
| **4.products** | The system must allow the customer to calculate a total of cast |

**Search**

|  |  |
| --- | --- |
| **1.Search** | The system must allow the customer to **Search** about product in the System |
| **1.Search** | The system must allow the customer to **Search** about product in google |

**Payment**

|  |  |
| --- | --- |
| **1.Payment** | The system must allow the customer to **pay** on line |
| **2.Payment** | The system must allow the customer to **pay** about use credit card |

**Help**

|  |  |
| --- | --- |
| **1.Help** | The System must allow the customer to asking a Help |
| **2.Help** | The System must allow the customer to Make a complaint |

**Assistant**

|  |  |
| --- | --- |
| **1.Assistant** | The system must allow the assistant help customers |
| **2.Assistant** | The system must allow the assistant replay to complaint of users |

**Confirm**

|  |  |
| --- | --- |
| **1.Confirm receipt** | the system must allow user confirm receipt |
| **1.Track delivery** | the system must allow user track delivery |

**Feedback**

|  |  |
| --- | --- |
| **1.Feedback** | The System must allow for user to sent a feedback about services |

**Category**

|  |  |
| --- | --- |
| **1.Category** | The System must allow for user to know Categories by **code** |
| **2.Category** | The System must allow for user to know Categories by **name** |
| **3.Category** | The System must allow for user to know that Categories is expired or not by **State** |

**3.2 Non-Functional Requirements**

* ***LOOK AND FEEL REQUIREMENTS***

* This system is user friendly.
* Approachable, so that people do not hesitate to use it.
* Authoritative, so that users feel they can rely on it and trust it.
* Attractive to students or some other specific group.
* Innovative and appearing to be state of the art.
* Cool.

**Performance**

|  |  |  |
| --- | --- | --- |
| **1.** **Performance** |  | The system must respond the business operation in less than 3 seconds for user |
| **2.** **Performance** |  | The system should be reliable. |
| **3. Performance** |  | The system must mange many users at the same time |

**Simple**

|  |  |
| --- | --- |
| **1. Simple** | Shall at maximum three color |
| **2. Simple** | Decrease confusion. |
| **3. Simple** | The product be easy to use on the first attempt by a member of the faculty without training |

**flexible and maintainable**

|  |  |
| --- | --- |
| **1.FM** | The product shall be used on both computer and cellphones |
| **2.FM** | The product be able to be modified to cope with new courses, subjects, staff, and departments. |

**safe and secure**

|  |  |
| --- | --- |
| **1.SS** | The product shall ensure that only authorized admins can manipulate information. |
| **2.SS** | The product shall be immune against threats and attacks |

**formal**

|  |  |
| --- | --- |
| **1.Formal** | The product shall comply with the college laws. |
| **2.Formal** | The product shall comply with the laws of the ministry of higher education in Egypt |

**Usability and Humanity**

|  |  |
| --- | --- |
| **1.UAH** | Centralized maintain all information. |
| **2.UAH** | Easy to search the information about Products . |
| **3.UAH** | The system will save significant amount of time and effort. |
| **4.UAH** | Easy to manage whole counseling procedure. |
| **5.UAH** | Facilitate communication between Admin and User. |
| **6.UAH** | Easy to use on the first attempt by someone without training. |

**Operational & Environmental REQS**

|  |  |
| --- | --- |
| **1.OAE** | website shall conserve data of .internet . |

**Look-and-Feel REQS**

|  |  |
| --- | --- |
| **1.Look and Feel** | website should be beautiful .. The colors are consistent.. |

**Reliability**

|  |  |
| --- | --- |
| **1. Reliability** | The system servers and services should be available when it is requested by the user. |
| **2. Reliability** | user can easily connect with admin or system is any problem happened. |
| **3. Reliability** | has a very high-quality maintaining system to avoid any crash. |

**OCL(object constrains language):**

context User

inv: self.age >= 18 -- Users must be adults

inv: self.address->notEmpty() -- Users must have a valid address

pre: self.password size >= 8 --Password must be strong

inv: self.email -> unique & not null

context Product

inv: self.price > 0 -- Product price must be positive

inv: self.quantityAvailable >= 0 -- Available quantity must be non-negative

inv: self.nameOfProduct -> not null

context ShoppingCart

inv: self.items->notEmpty() -- Shopping cart must not be empty

post: self.totalAmount >= 0 -- Total amount must be non-negative

context Order

inv: self.items->notEmpty() -- Order must contain at least one item

inv: self.status = 'Placed' or self.status = 'Shipped' or self.status = 'Delivered' -- Order status must be valid

context Payment

pre: self.amount >= 0 -- Payment amount must be non-negative

post: self.status = 'Pending' or self.status = 'Completed' or self.status = 'Failed' -- Payment status must be valid

context Shipping

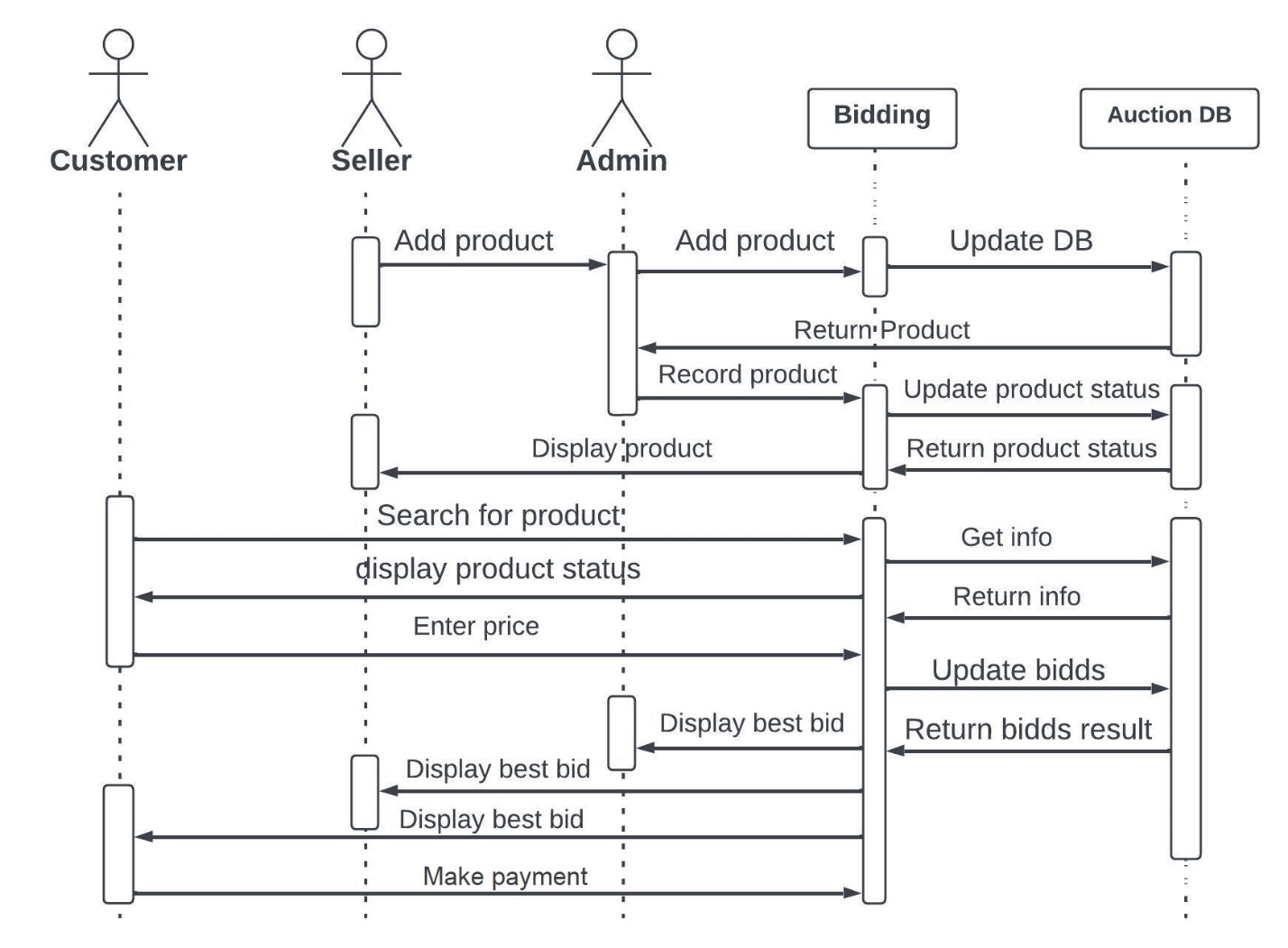
inv: self.address->notEmpty() -- Shipping address must be provided

post: self.status = 'Pending' or self.status = 'Shipped' or self.status = 'Delivered' -- Shipping status must be valid

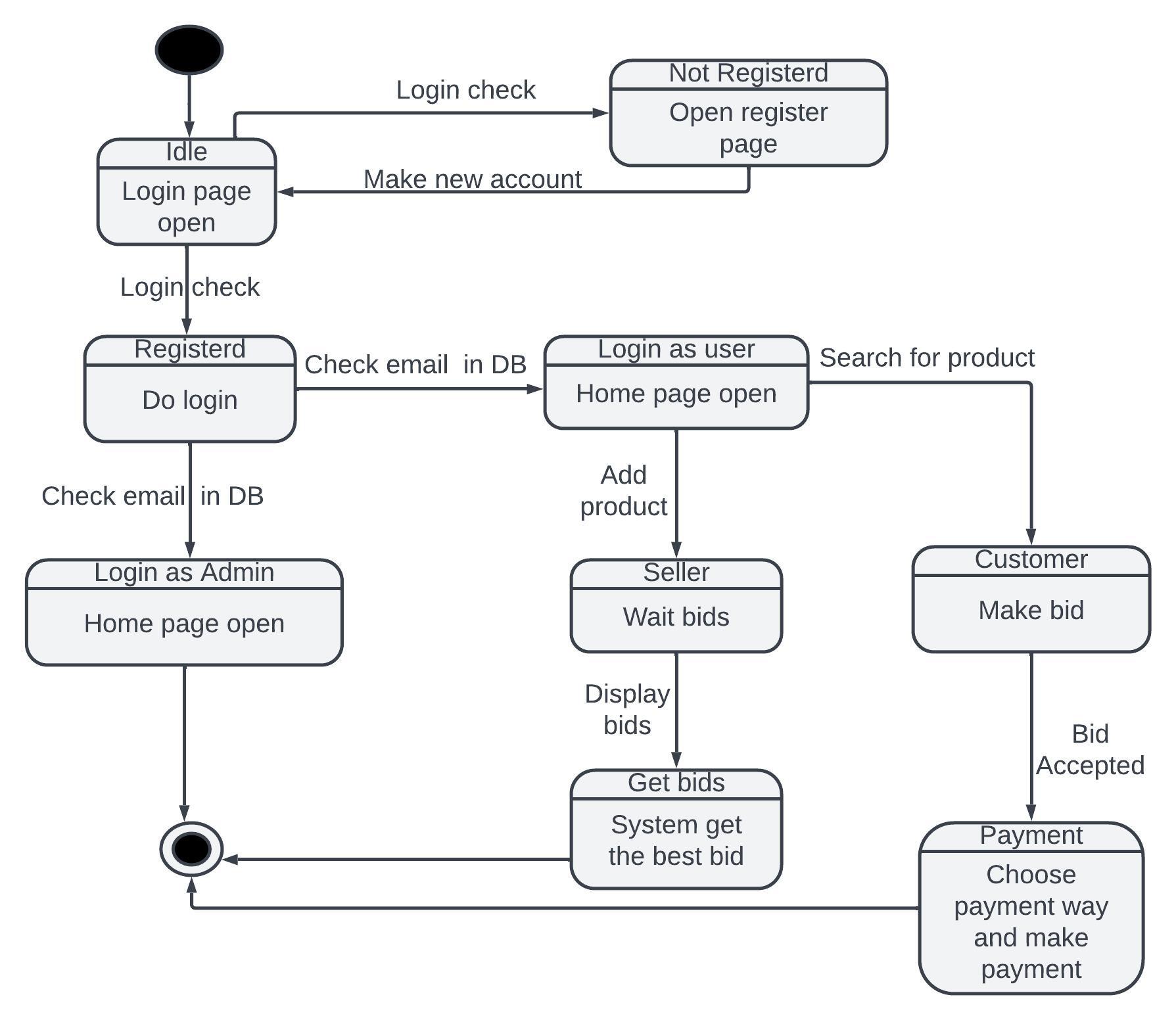
**Class diagram**



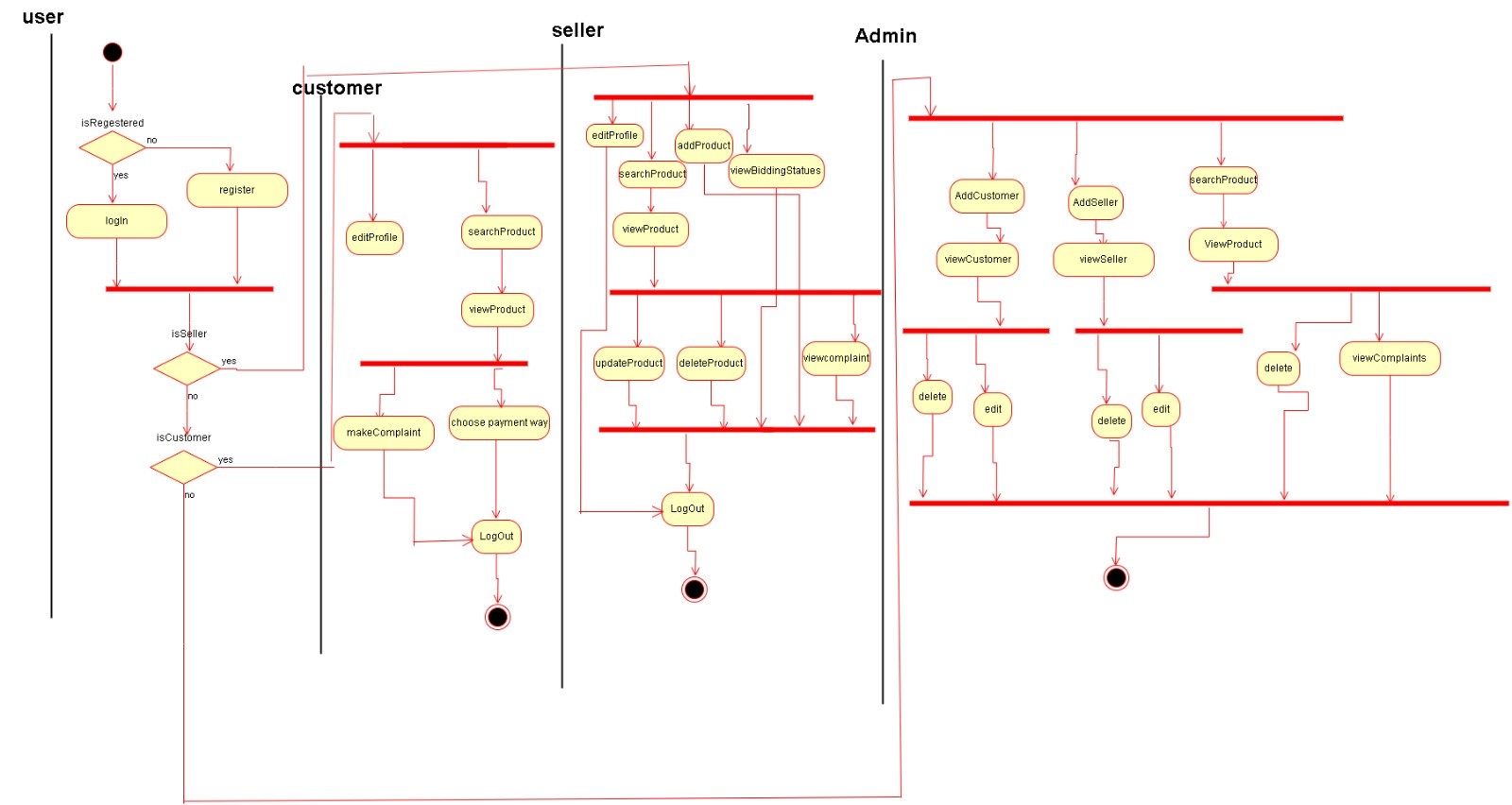
**Sequecnce diagram**



**State diagram**

****

**Activity diagram**

****