**Helwan University - Faculty of Computers & Artificial Intelligence** – **Computer Science Department Module: CS251 Software Engineering 1 – Spring “Semester 2” 2022-2023**

**THIS FORM IS FOR BOTH THE GENERAL & MEDICAL INFORMATICS PROGRAMMES**

# S E - I C O U R S E P R O J E C T ( P H A S E 1 C O V E R S H E E T )

**Discussions Scheduled for Week 8** *(more details will be announced later)***.**

* Print 1 copy of this cover sheet and attach it to a printed copy of the documentation *(SRS, … etc.)*. You must also submit softcopies of all your documents *(as PDFs)*; details will be announced later.
* Please write all your names in Arabic. Please make sure that your students’ IDs are correct.
* Handwritten Signatures for the attendance of all team members should be filled in before the discussion.
* Please attend the discussion on time *(announced separately)*, late teams will lose 5 grades.

# Project Name: \_ **An AI Multi-Agent Shopping Application** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Team Information *(typed not handwritten, except for the attendance signature)*:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **ID [Ordered by ID]** | **Full Name [In Arabic]** | **Attendance [Handwritten Signature]** | **Final Grade** |
| **1** | 20210033 | احمد باسم كمال عوده |  |  |
| **2** | 20210045 | احمد خالد جمعه عبدالسميع |  |  |
| **3** | 20210037 | احمد جمال عبدالحي |  |  |
| **4** | 20210025 | احمد اشرف احمد محمود |  |  |
| **5** | 20210049 | احمد خالد نصر عبدالعزيز |  |  |
| **6** | 20210047 | احمد خالد عبدالله عبدالقادر |  |  |
| **7** | 20210053 | احمد رمضان رجب منصور |  |  |

**Grading Criteria:**

## Page **1** of **2**

**Helwan University - Faculty of Computers & Artificial Intelligence** – **Computer Science Department Module: CS251 Software Engineering 1 – Spring “Semester 2” 2022-2023**

|  |  |  |  |
| --- | --- | --- | --- |
| **10 Items** |  | **Grade** | **Notes** |
| **1. Functional Requirements** | **1** |  |  |
| **2. Non-Functional Requirements** | **1** |  |  |
| **3. Use-Case Diagram(s)** *including general use-cases for the system, and the detailed use-cases description* | **2** |  |  |
| **4. System Architecture –** *including applied Architectural Pattern(s)* | **1** |  |  |
| **5. Activity Diagrams** | **2** |  |  |
| **6. Database Specification** *(ERD, Tables)* | **2** |  |  |
| **7. Class Diagram** *(Interfaces, Classes, Relations) –* An initial version based on the requirements and Use-Case/Activity diagrams. | **2** |  |  |
| **8. Object Diagrams** *(Including object diagrams that illustrate the preconditions and the post-conditions of selected functions)* | **1** |  |  |
| **9. Package Diagram(s)** | **1** |  |  |
| **10. Sequence Diagrams** *including* ***System Sequence Diagrams (SSDs)*** | **2** |  |  |

**15**

# Teaching-Assistant’s Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Page **2** of **2**

|  |  |
| --- | --- |
| **Software Requirements Specifications For An AI Multi-Agent Shopping Application** | Represented by  ***AHMAD BASSEM KAMAL AHMED KHALED GOMAA AHMED GAMAL ABDELHAI AHMED ASHRAF AHMED AHMED RAMADAN RAJAB AHMED KHALED NASR AHMED KHALED ABDALLAH*** |

1. **Introduction**

**1.1 Purpose**

The main purpose for this software is allowing users to shop online with the help of multiple agents. This application is made to provide a more efficient and personalized shopping experience for users.

The multi-agent aspect of the application allows for collaboration between different agents to provide a more extensive and accurate shopping experience.

Overall, the goal of a multi-agent shopping application is to provide a more efficient and enjoyable shopping experience for users, while also improving the overall efficiency and productivity of online businesses.

By using software agents to automate and optimize multiple aspects of the shopping process, this application can help businesses attract and retain customers, increase sales, and improve customer satisfaction.

1. **System Features**

**2.1 Non Functional Requirements**

1. **Security**: The ecommerce system must ensure the security of user data, including personal information and payment details. This includes measures such as encryption, firewalls, and secure
2. **Performance**: The ecommerce system must be able to handle a large number of users and transactions without slowing down or crashing. This includes measures such as load balancing, caching, and optimization of database queries.
3. **Reliability**: The ecommerce system must be reliable and available at all times, with minimal downtime or errors. This includes measures such as backup and recovery, fault tolerance, and disaster recovery planning.
4. **Scalability**: The ecommerce system must be able to scale up or down depending on the demand, without affecting the performance or reliability of the system. This includes measures such as horizontal scaling, vertical scaling, and cloud-based hosting.
5. **Usability**: The ecommerce system must be easy to use and navigate for all types of users, including those with disabilities. This includes measures such as clear and intuitive user interfaces, accessibility features, and user testing.
6. **Compliance**: The ecommerce system must comply with relevant laws and regulations, such as data protection laws, consumer protection laws, and payment card industry standards.
7. **Interoperability**: The ecommerce system must be able to integrate with other systems, such as payment gateways, shipping providers, and inventory management systems. This includes measures such as API documentation and support for industry standards.

**2.2 Functional Requirements**

|  |  |
| --- | --- |
| **Name, Description** | **Log in:** System enables users to use their accounts in order to purchase products. |
| **Actors** | Registered user |
| **Pre-Condition** | User is registered on website. |
| **Post-Condition** | User is logged in website. |
| **Main Success Path** | User successfully logs in and is able to perform any desired purchase from now. |
| **Alternate Path** | User is not registered in website so he can’t login. |

**Steps:**

1. User is registered on the website.
2. User adds his email and password.
3. System validates email and password.
4. If the email and password are correct, customer will log in.

|  |  |
| --- | --- |
| **Name, Description** | **Sign Up:** System adds new user to the system. |
| **Actors** | User |
| **Pre-Condition** | Email is not registered. |
| **Post-Condition** | Email is registered. |
| **Main Success Path** | User can register on the website with different roles (customer, seller, agent), user inserts data: as a customer (name, email, password, phone number), as a seller (contact information, business information, business address), as an agent (name, email, password, phone number, national id). |
| **Alternate Path** | User can’t complete his/her registration because the email is used before, or missing information |

**Steps:**

1. User clicks on sign up to the website.
2. User writes his/her information.
3. Information is added successfully.
4. A new account is added to the website database and the user can log in with it.

|  |  |
| --- | --- |
| **Name, Description** | **Add to Cart:** Customer can add products to his/her cart, where products are collected to be purchased later. |
| **Actors** | Customer |
| **Pre-Condition** | A product must be available and not out of stock, so that the customer can add it to the cart. |
| **Post-Condition** | The wanted product is successfully added to the customer’s cart. |
| **Main Success Path** | The wanted product is available then the customer adds it to his/her cart. |
| **Alternate Path** | The wanted product is out of stock then the customer can’t add it to the cart. |

**Steps:**

1. The customer selects a specific product (from Home Page or after searching).
2. The customer adds the wanted product to his/her cart.

|  |  |
| --- | --- |
| **Name, Description** | **Modify Cart:** Customer can modify products in his/her cart by either adding or reducing a product quantity or removing a whole product from the cart. |
| **Actors** | Customer |
| **Pre-Condition** | The customer’s cart shouldn’t be empty, contains at least one product. |
| **Post-Condition** | The modification is done either by changing a product quantity or removing it. |
| **Main Success Path** | The cart is not empty, and the customer does the wanted modifications. |
| **Alternate Path** | The cart is already empty so the customer can’t do any modifications. |

**Steps:**

1. The customer opens his/her cart.
2. Select the wanted product to be modified.
3. Modification is done if it meets the requirements.

|  |  |
| --- | --- |
| **Name, Description** | **Add Review:** Customer can add review and rate products. |
| **Actors** | Customer |
| **Pre-Condition** | Customer is logged in and the wanted product is shown. |
| **Post-Condition** | The review and the rate are added. |
| **Main Success Path** | Review is added successfully |
| **Alternate Path** | Can’t add a review if you don’t login. |

**Steps:**

1. Customer is logged in.
2. Customer selects a product.
3. Customer adds a review and a rate to the product.

|  |  |
| --- | --- |
| **Name, Description** | **Make order**: Customer orders his products that are in cart. |
| **Actors** | Customer |
| **Pre-Condition** | Customer is logged in and cart is not empty. |
| **Post-Condition** | Customer is ready for payment. |
| **Main Success Path** | Order is made. |
| **Alternate Path** | If the credit card expiry date passed, not enough credit, or user entered wrong card information the order can’t be done. Also, if some product gets out of stock the purchase is blocked. |

**Steps:**

1. Customer is logged in.
2. Customer selects one or more products.
3. Customer adds product to cart.
4. Customer proceeds to checkout and payment confirmation.
5. Order made successfully.

|  |  |
| --- | --- |
| **Name, Description** | **Pay Online:** Customer pays for his/her orders. |
| **Actors** | Customer |
| **Pre-Condition** | Valid card information and enough credit. |
| **Post-Condition** | Payment confirmed and order is made. |
| **Main Success Path** | Products are purchased successfully. |
| **Alternate Path** | Invalid card or not enough credit. |

**Steps:**

1. System ask user about the payment method he/she prefers.
2. User write his/her information.
3. The transaction is successfully done.

|  |  |
| --- | --- |
| **Name, Description** | **Add Product:** Seller adds a new product. |
| **Actors** | Seller |
| **Pre-Condition** | Product is not added before and seller must be logged in. |
| **Post-Condition** | Product is added and ready to be purchased. |
| **Main Success Path** | New product's data (name, store name, price, brand, category, release year, description, keywords) is added. |
| **Alternate Path** | Redundancy is checked for the product, and it already exists, so it will not be added. |

**Steps:**

1. Seller clicks on add a new product.
2. System asks the seller to enter the product details.
3. Seller writes the information of the new product.
4. System adds the product successfully.

|  |  |
| --- | --- |
| **Name, Description** | **Modify Products:** Seller modifies products’ data. |
| **Actors** | Seller |
| **Pre-Condition** | Wanted product exists. |
| **Post-Condition** | Seller updates product information. |
| **Main Success Path** | Product is modified. |
| **Alternate Path** | Seller can’t update product information |

**Steps:**

1- Seller clicks on the products he intends to modify.

2- Seller updates one or more fields for the product.

3- Seller saves modification and it’s now applied visible to all users.

|  |  |
| --- | --- |
| **Name, Description** | **Delete Product:** Seller deletes one or more of his shown products. |
| **Actors** | Seller. |
| **Pre-Condition** | Seller must have one or more shown products. |
| **Post-Condition** | Product deleted successfully. |
| **Main Success Path** | All product’s data is deleted from the system. |
| **Alternate Path** | The product which is wanted to be deleted has been requested from one or more customer so it can’t be deleted. |

**Steps:**

1. Seller opens products page.
2. Seller selects the product he wants do delete.
3. Seller clicks on delete product.
4. System checks for requests on this product.
5. Product deleted successfully.

|  |  |
| --- | --- |
| **Name, Description** | **View Requests Information:** Seller needs to track where he sells most of his/her products, so he/she can access some customers’ data (Country, City) to track orders’ traffic. |
| **Actors** | Seller |
| **Pre-Condition** | Order is requested. |
| **Post-Condition** | Seller can access some customers’ data. |
| **Main Success Path** | Seller can view customer’s address (country, city). |
| **Alternate Path** | There is an error, and the seller can’t see the customer data |

**Steps:**

1. Seller clicks view requests Information button.
2. System displays some information about the users which buy his/her products (country, city).

|  |  |
| --- | --- |
| **Name , Description** | **Agent View Orders:** Agent can view all submitted orders in order to choose any of the available orders in reach to deliver them to the customer. |
| **Actors** | Agent |
| **Pre-Condition** | Agent is logged in. |
| **Post-Condition** | Agent will be able to accept orders |
| **Main Success Path** | All sent orders can be viewed by all agents. |
| **Alternate Path** | No request available. |

**Steps:**

1. Notification or order will be sent to the agent.
2. The Agent can see the orders on the orders page.

|  |  |
| --- | --- |
| **Name , Description** | **Agent accepts orders:** Agent accepts one or more from the sent orders and delivers the requested products to their customers. |
| **Actors** | Agent |
| **Pre-Condition** | Agent is logged in and at least one order is requested that is not accepted by any other agent. |
| **Post-Condition** | Order is removed from the agent page; order acceptance notification is sent to the user and the seller. |
| **Main Success Path** | Agent can accept orders. |
| **Alternate Path** | The order is accepted by another agent and can’t be accepted by this agent. |

**Steps:**

1. The agent will receive notification about the order.
2. Orders page will display available orders and their locations for the agent.
3. The agent can accept any order.

|  |  |
| --- | --- |
| **Name , Description** | **Agent modifies order state:** Agent can update accepted order’s status. |
| **Actors** | Agent |
| **Pre-Condition** | Agent has accepted at least one order, on which the modification can be done. |
| **Post-Condition** | The modification is done, and the order’s status is updated. |
| **Main Success Path** | The modification is done successfully. |
| **Alternate Path** | The agent has no accepted orders on which the modification can be done. |

**Steps:**

1. The agent can choose any order that he has already accepted.
2. Agent can change the order’s status to “charged”, “on delivery” or “delivered” according to the progress of the delivery process.

|  |  |
| --- | --- |
| **Name, Description** | **View Request Page:** Customer can view order state and track it until delivery. |
| **Actors** | Customer |
| **Pre-Condition** | Customer has requested at least one order and that order has been accepted. |
| **Post-Condition** | Customer will be able to see order's details and status. |
| **Main Success Path** | Customer can view updates done by agents on his orders. |
| **Alternate Path** | There is an error in the order state, and it stuck in the same state. |

**Steps:**

1. Customer opens the order page and checks the order state to know when it will arrive.
2. System will display the order state.

|  |  |
| --- | --- |
| **Name , Description** | **Search Products:** Customer can search on a specific product |
| **Actors** | Customer |
| **Pre-Condition** | Website is opened. |
| **Post-Condition** | Filtered results are shown depends on user’s search words. |
| **Main Success Path** | Customer can search on specific products. |
| **Alternate Path** | No products shown. |

**Steps**:

1. Customer opens the website.
2. Customer can search on any product, brand, or keyword.

|  |  |
| --- | --- |
| **Name, Description** | **Filter Products:** User can filter the search results |
| **Actors** | Customer |
| **Pre-Condition** | Search is done. |
| **Post-Condition** | Search results are filtered. |
| **Main Success Path** | Customer can filter the search results by (price, rating, brand, category, release date). |
| **Alternate Path** | There is an error, and the products aren’t filtered as costumer’s hope. |

**Steps:**

1. User applies filters that depend on his/her needs.
2. System filters the products and displays.

|  |  |
| --- | --- |
| **Name, Description** | **Add Coupon:** Seller can add a discount coupon that applies for all his products. |
| **Actors** | Seller |
| **Pre-Condition** | Seller is logged in and same coupon text doesn’t exist. |
| **Post-Condition** | Coupon is added successfully. |
| **Main Success Path** | Coupon is added to the system and the customer is able to use it to get a discount. |
| **Alternate Path** | Coupon is not added. |

**Steps:**

1. Seller opens coupon page.
2. Seller adds coupon information (coupon text, expiry date and discount).
3. Seller submits information and system checks for validity.
4. If coupon information is valid, then the coupon is added successfully.

|  |  |
| --- | --- |
| **Name , Description** | **Use coupon:** Customer can use a coupon to get a discount for some products. |
| **Actors** | Customer |
| **Pre-Condition** | Coupon exists and order is made. |
| **Post-Condition** | A discount is applied for certain products. |
| **Main Success Path** | Product’s price is decreased. |
| **Alternate Path** | Coupon is not valid, and no discount applies. |

**Steps:**

1. Customer selects some products.
2. Customer adds product to cart.
3. Customer makes an order.
4. Customer uses coupon by entered coupon text.
5. A discount is applied for certain products.

|  |  |
| --- | --- |
| **Name, Description** | **Log Out:** Registered user can log out of the system. |
| **Actors** | Registered user |
| **Pre-Condition** | User must be logged in. |
| **Post-Condition** | User logged out successfully. |
| **Main Success Path** | System is shown for the user as guest mode. |
| **Alternate Path** |  |

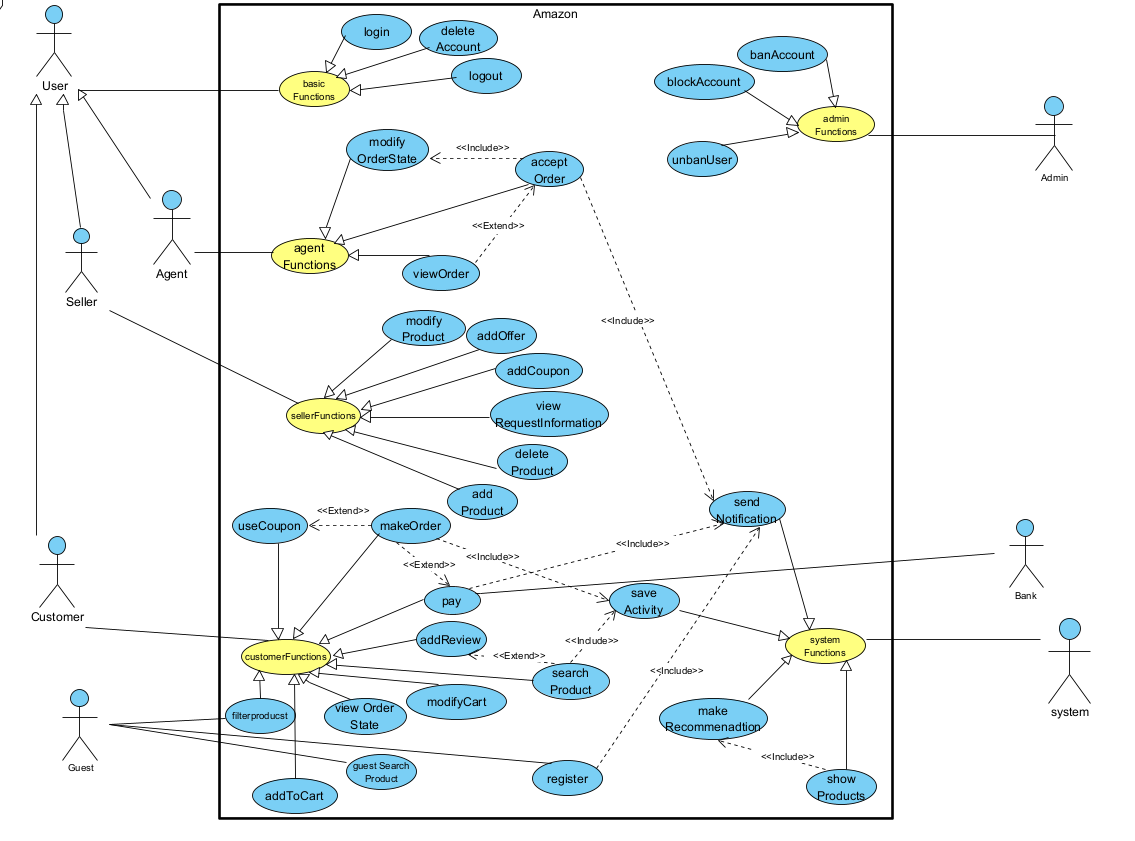
**Steps:**

1. User clicks on log out.
2. User logs out successfully.

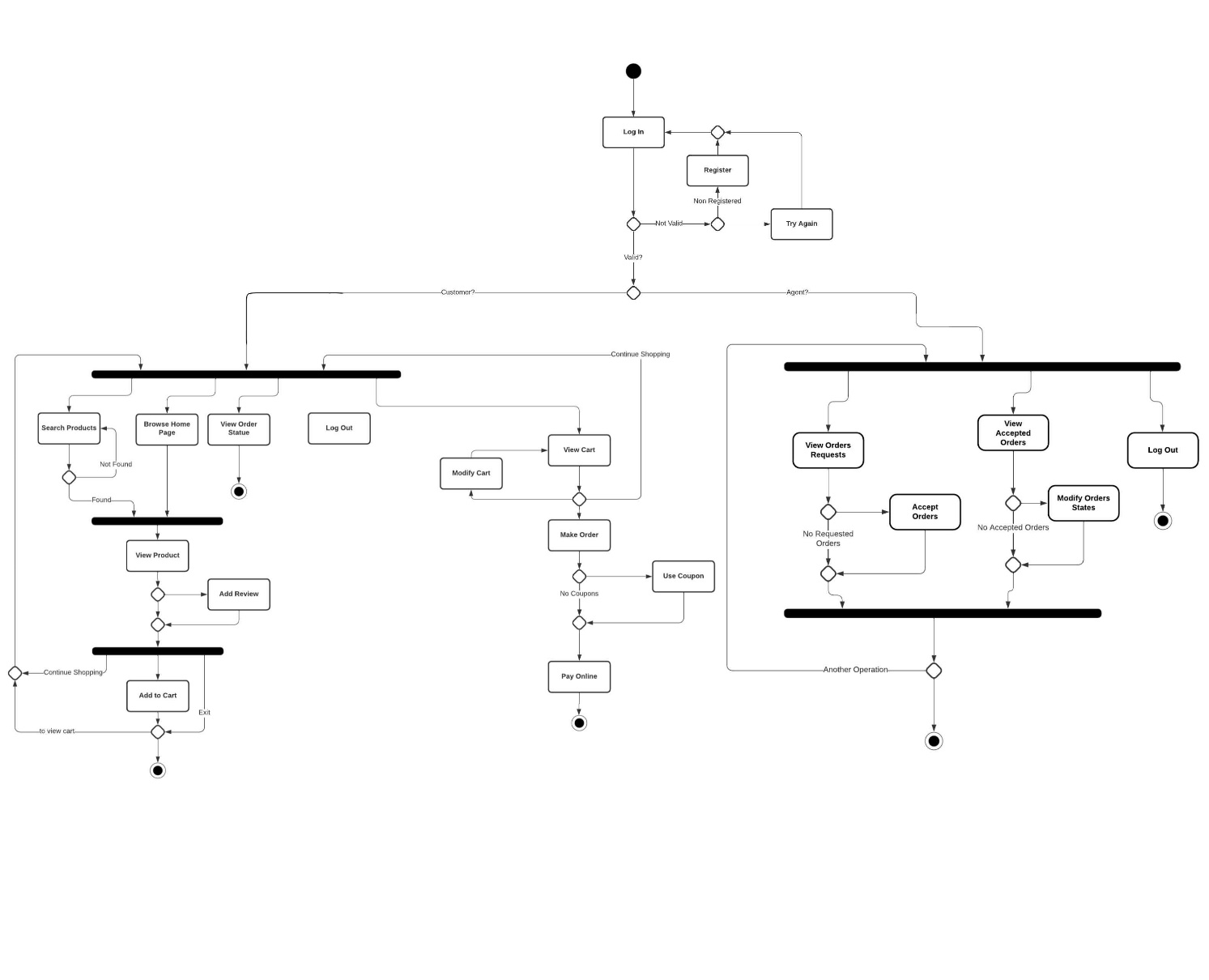
|  |  |
| --- | --- |
| **Name, Description** | **Delete Account:** User can delete his Account from the system. |
| **Actors** | Registered User |
| **Pre-Condition** | User must have an account. |
| **Post-Condition** | Account deleted successfully. |
| **Main Success Path** | User has no account on the system. |
| **Alternate Path** | User entered wrong on checking before deletion confirmation. |

**Steps:**

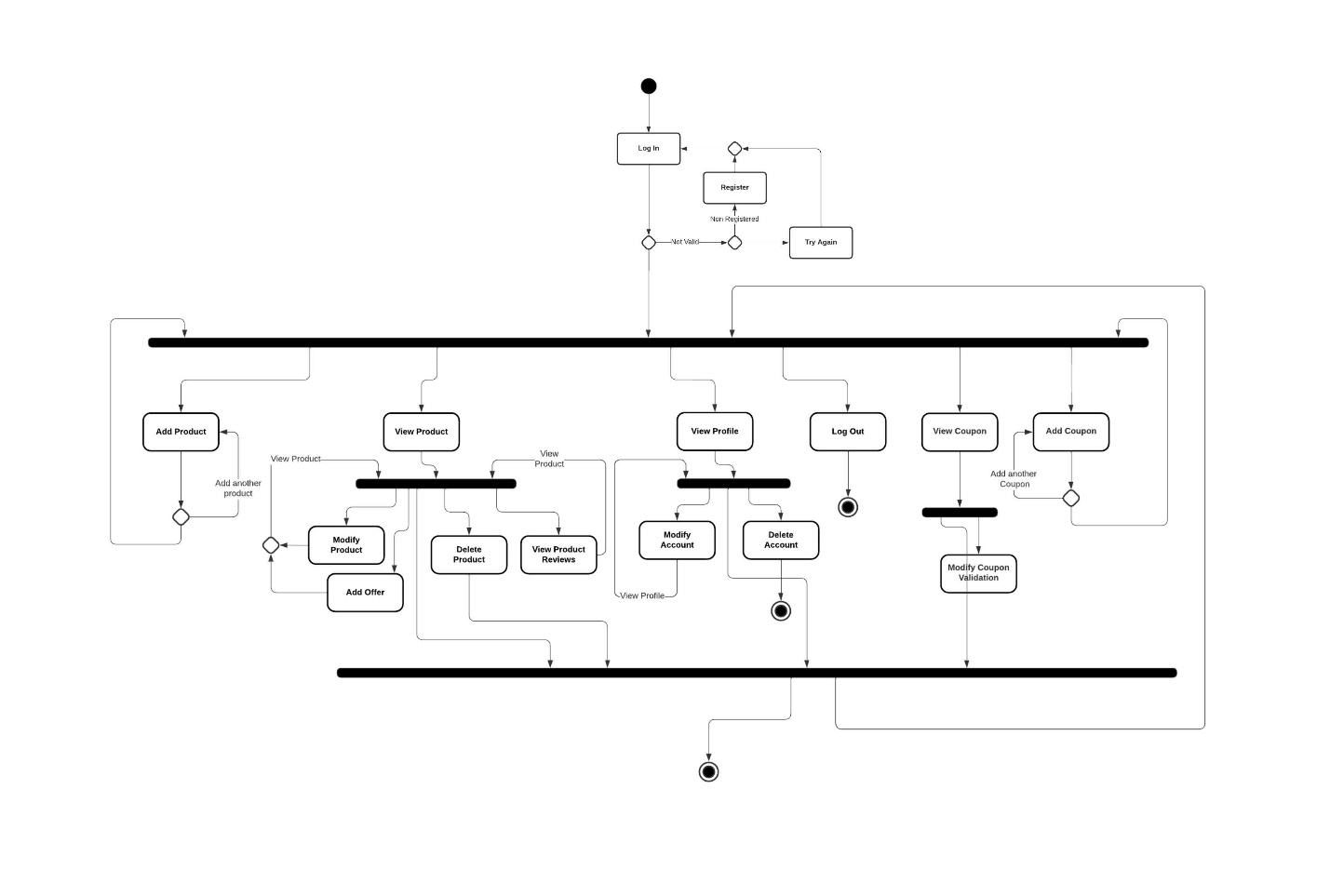
1. User opens account page.
2. User click on Delete account.
3. User enters his password.
4. System verifies user’s password.
5. Account is deleted.
6. **Analysis Models**
   1. **Use Case Diagram**



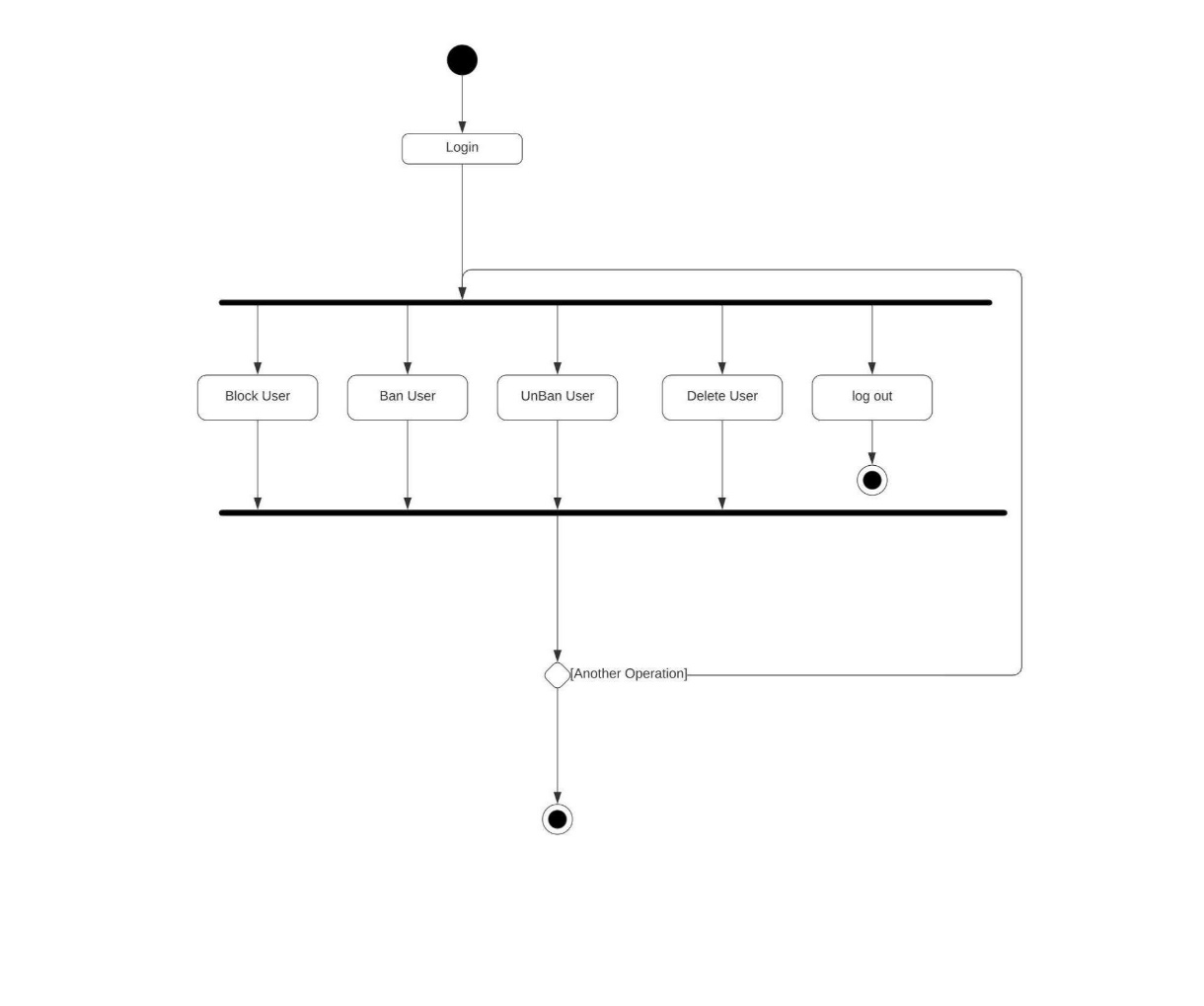
* 1. **Activity Diagram**
* **Customer and Agent activity diagram**



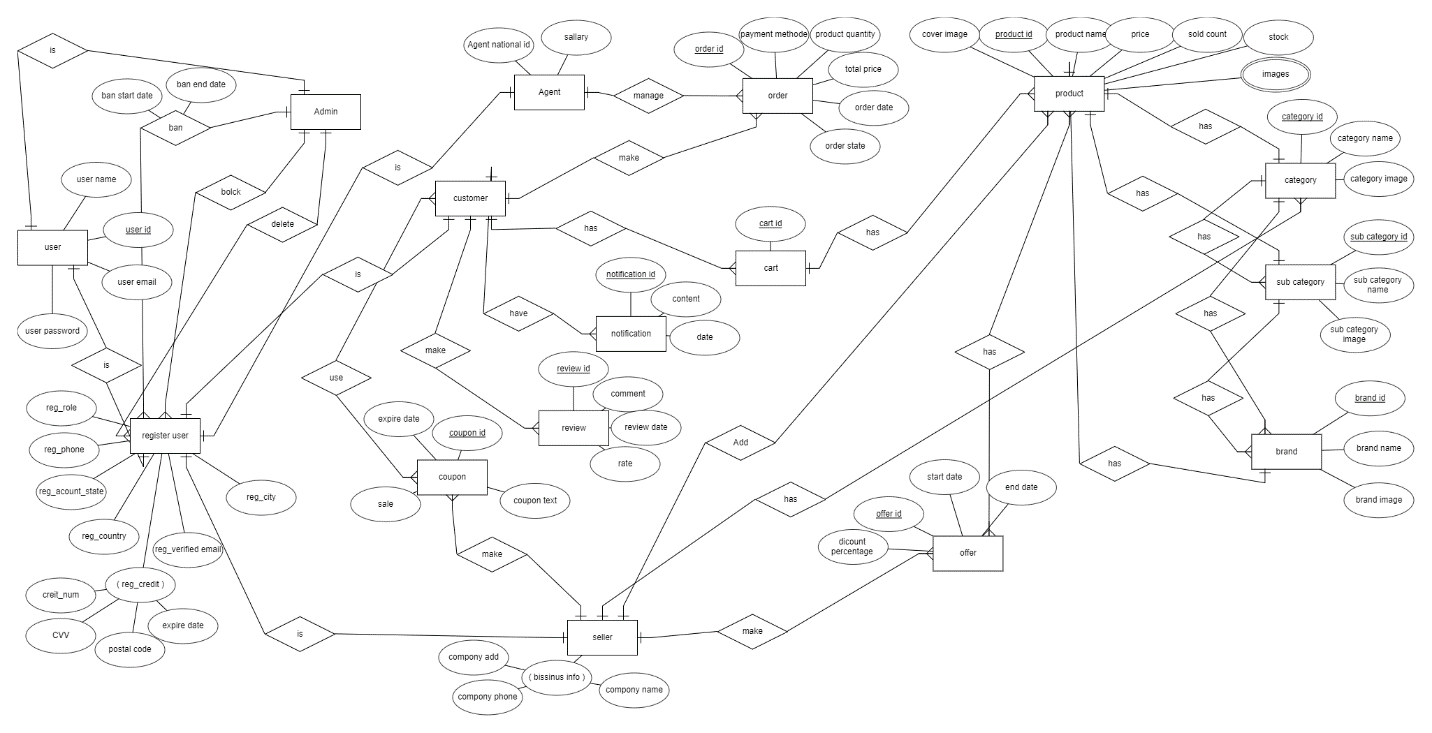
* **Seller activity diagram**

****

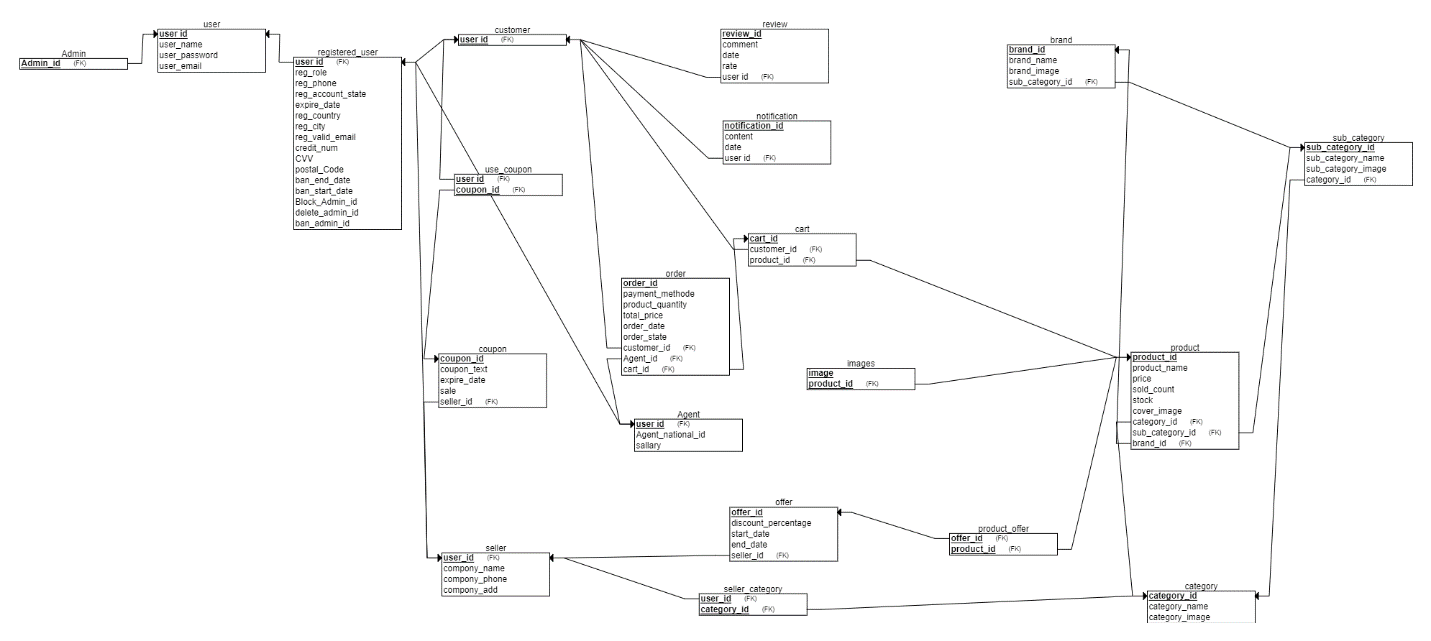
* **Admin activity diagram**

****

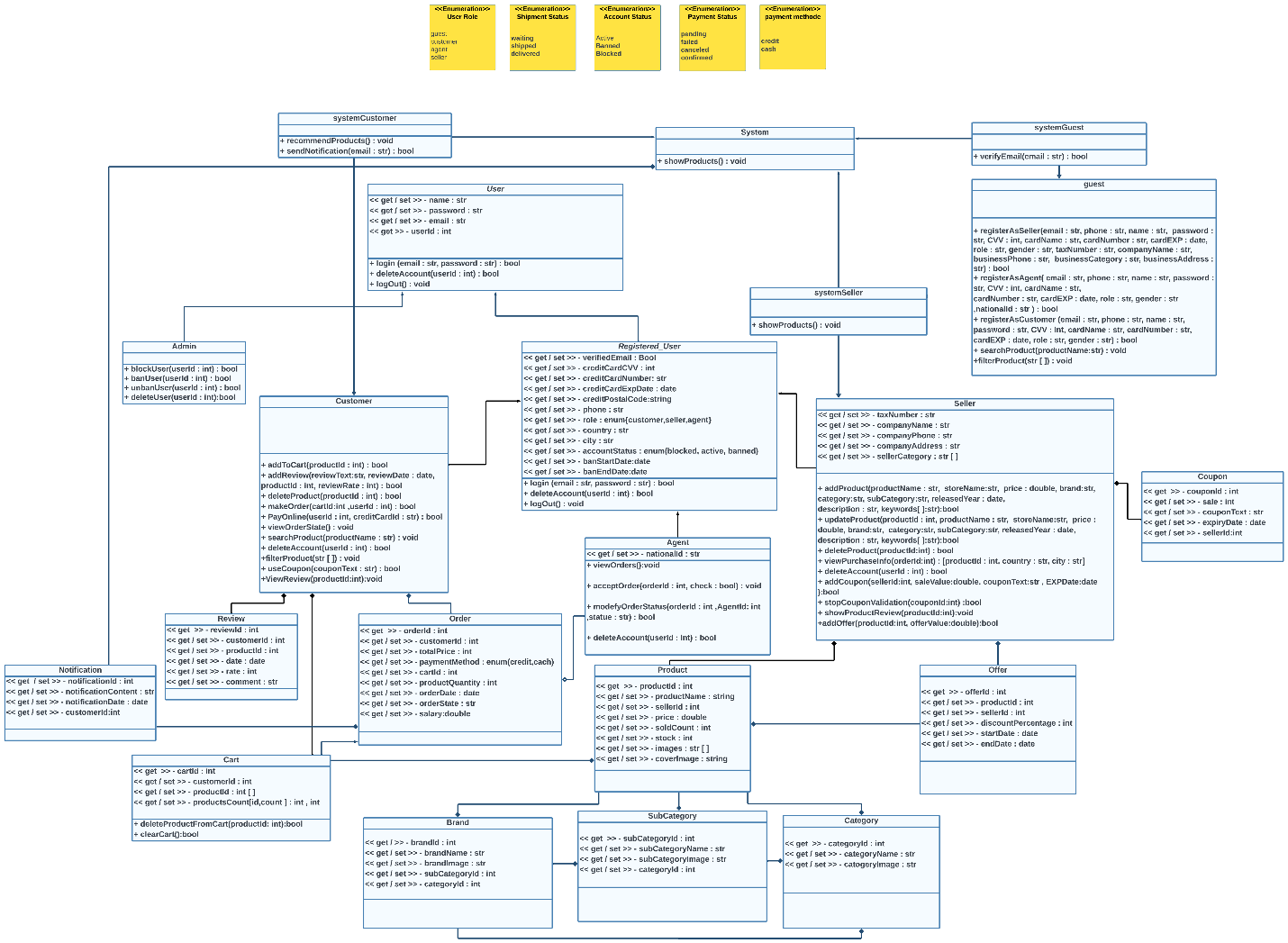
* 1. **Database Specifications**
* **Database ERD**

****

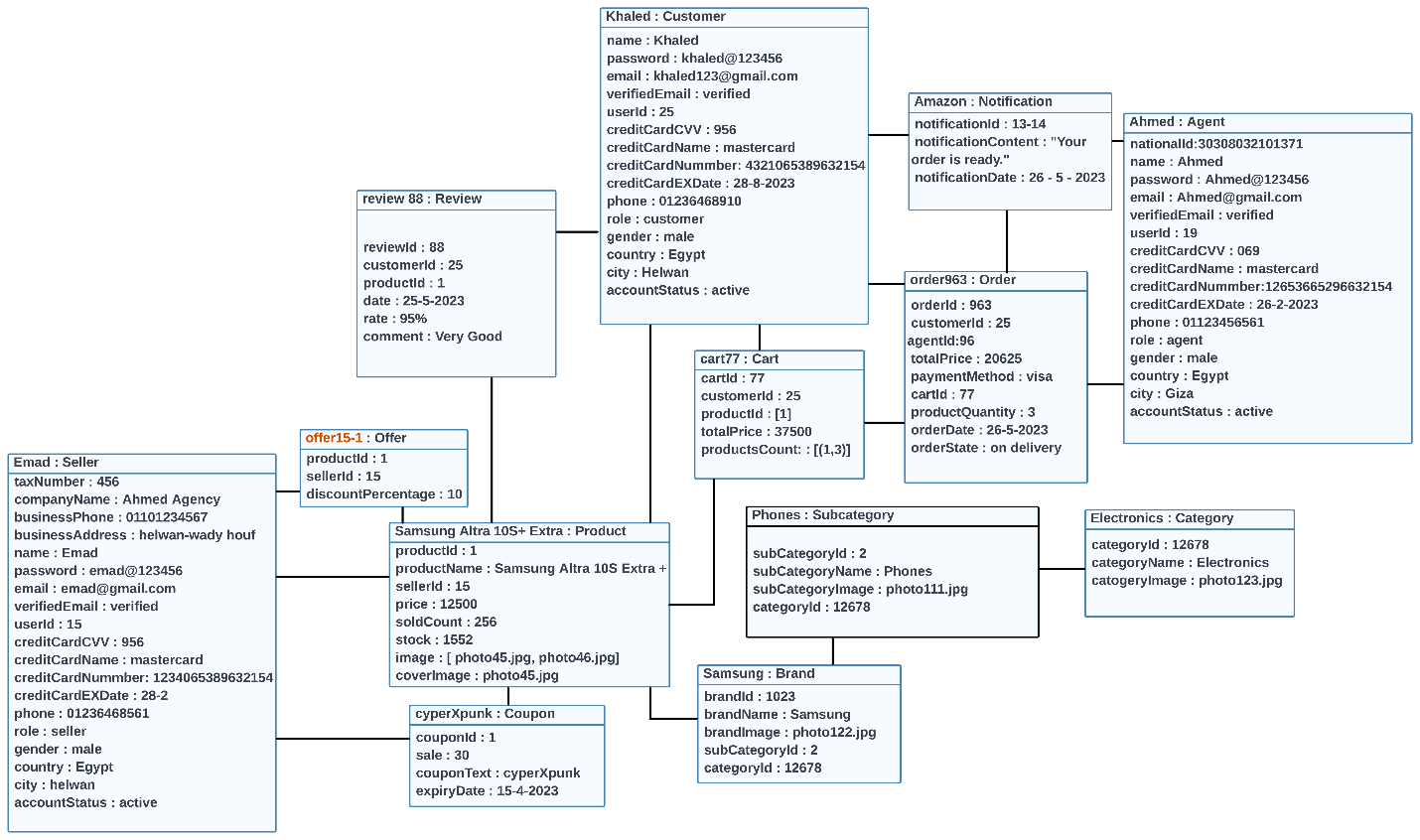
* **Database Tables**

****

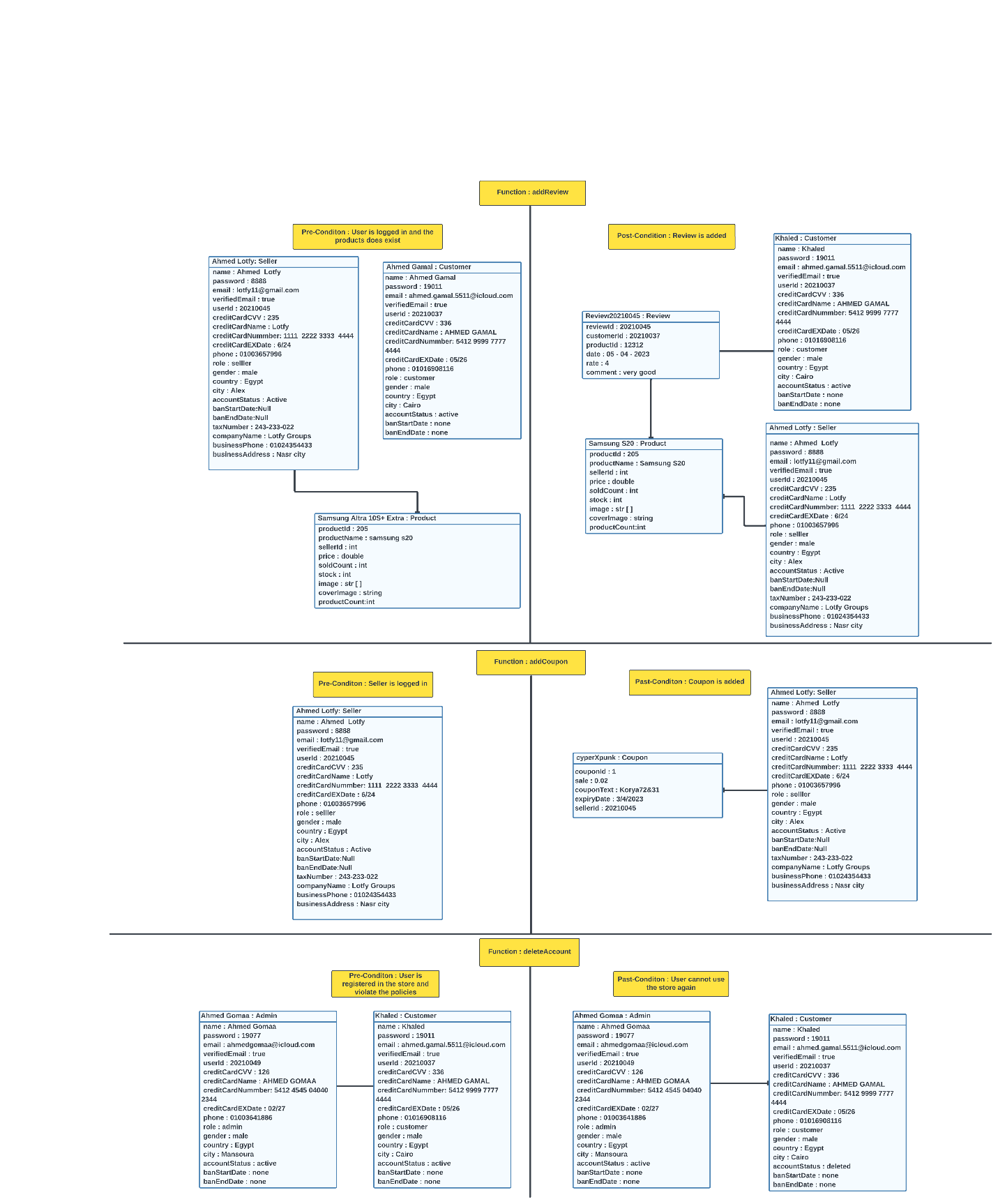
* 1. **Class Diagram**

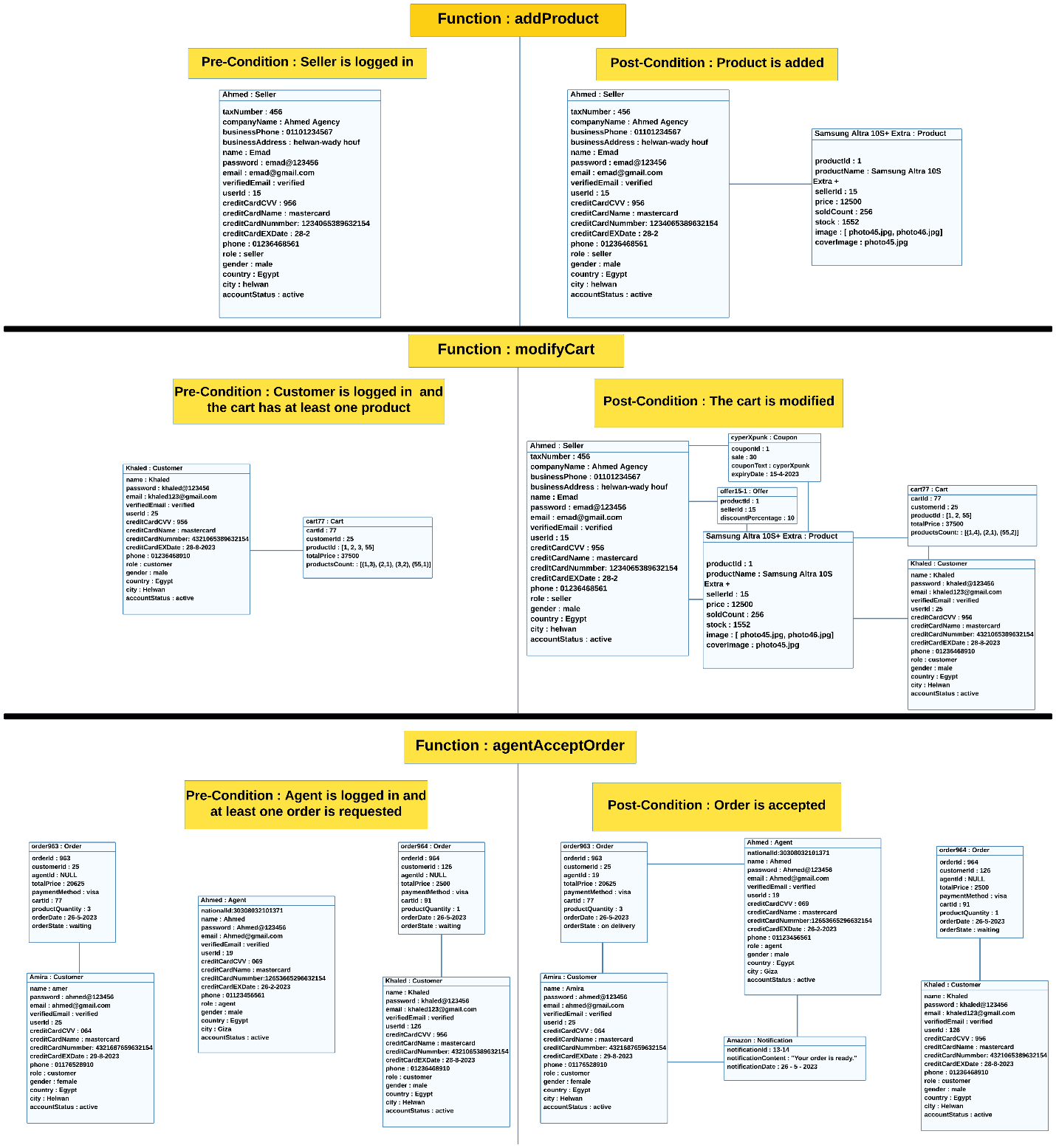
****

* 1. **Object Diagram**

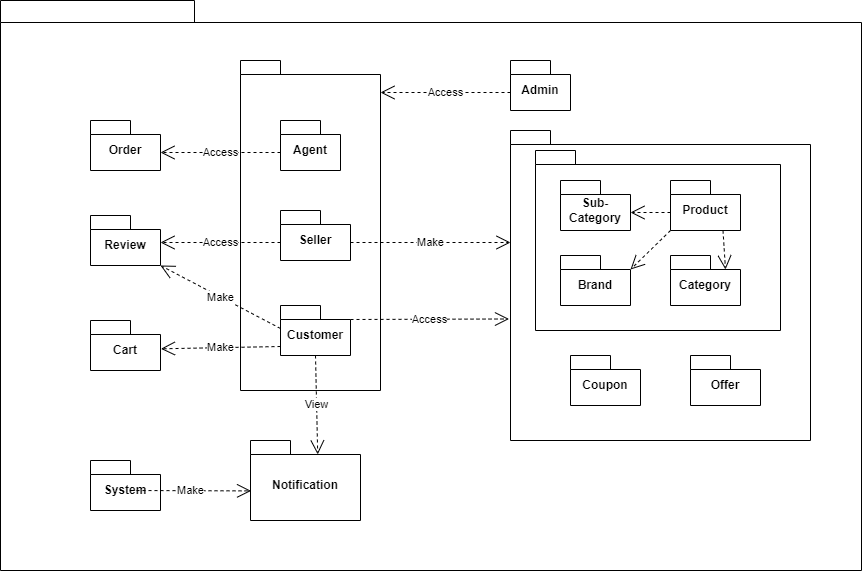
****

* **Some selected functions illustration by Object Diagram**

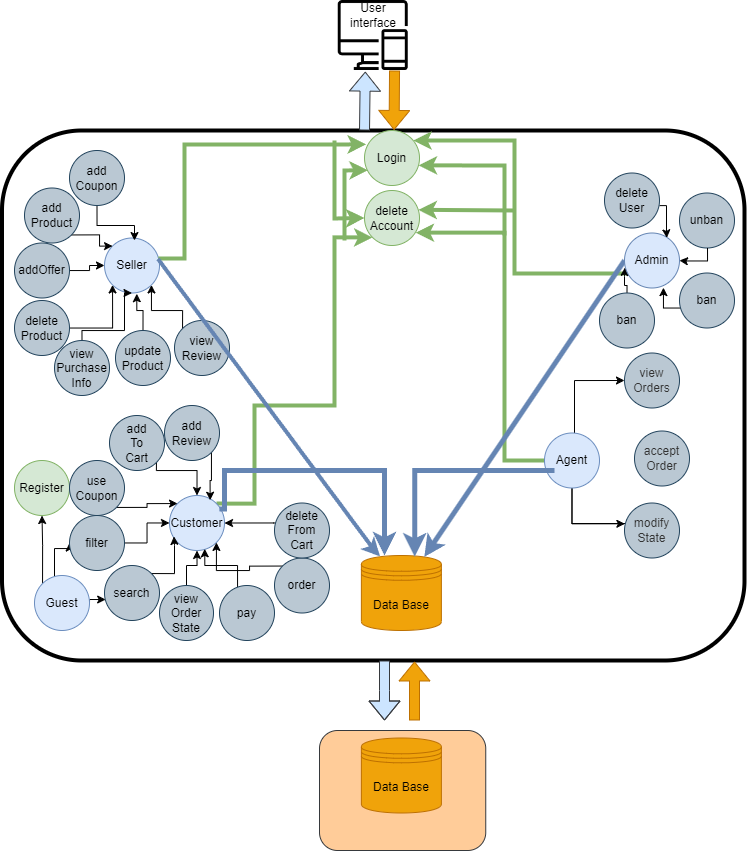


****

* 1. **Package Diagram**

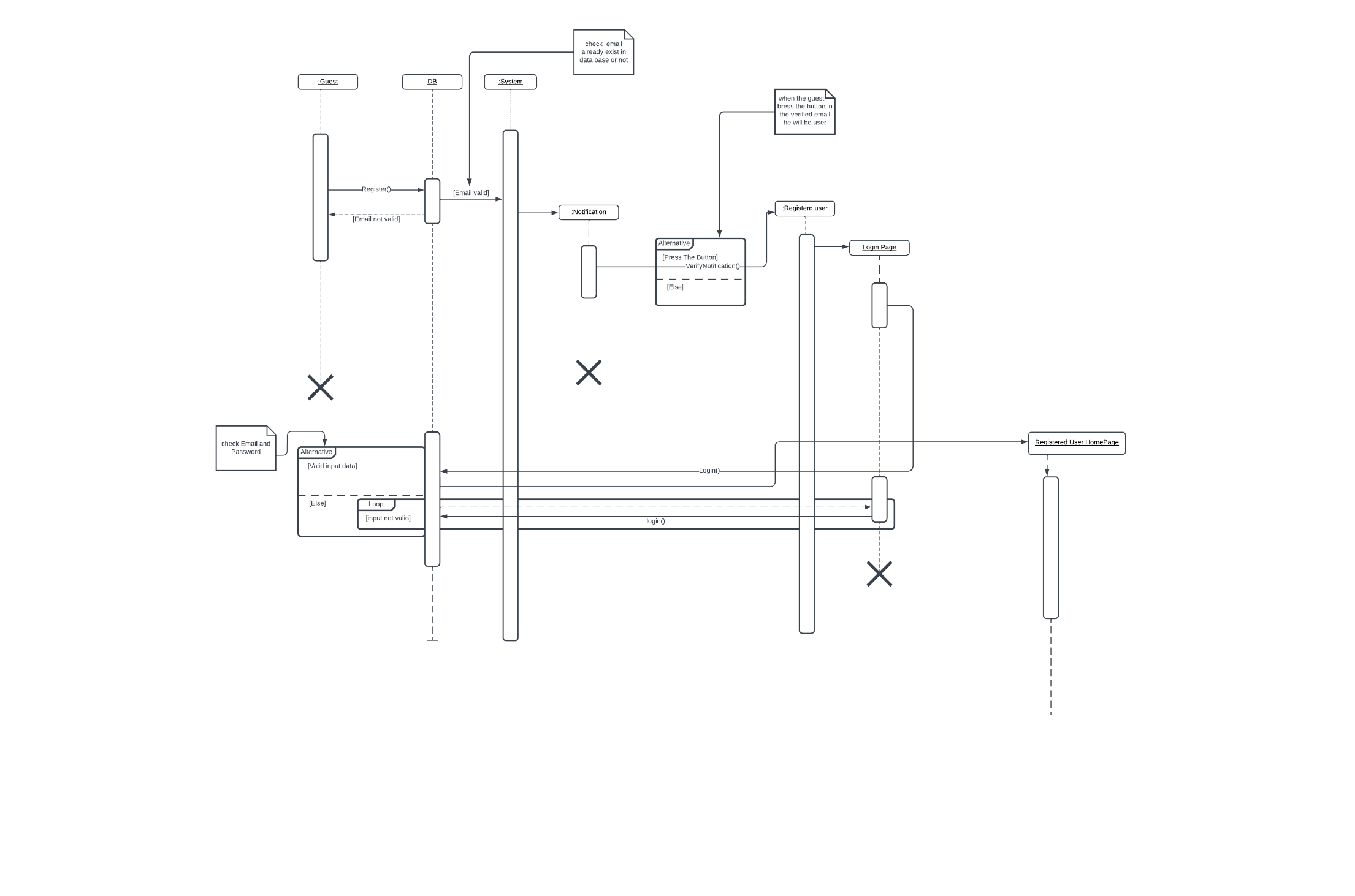
****

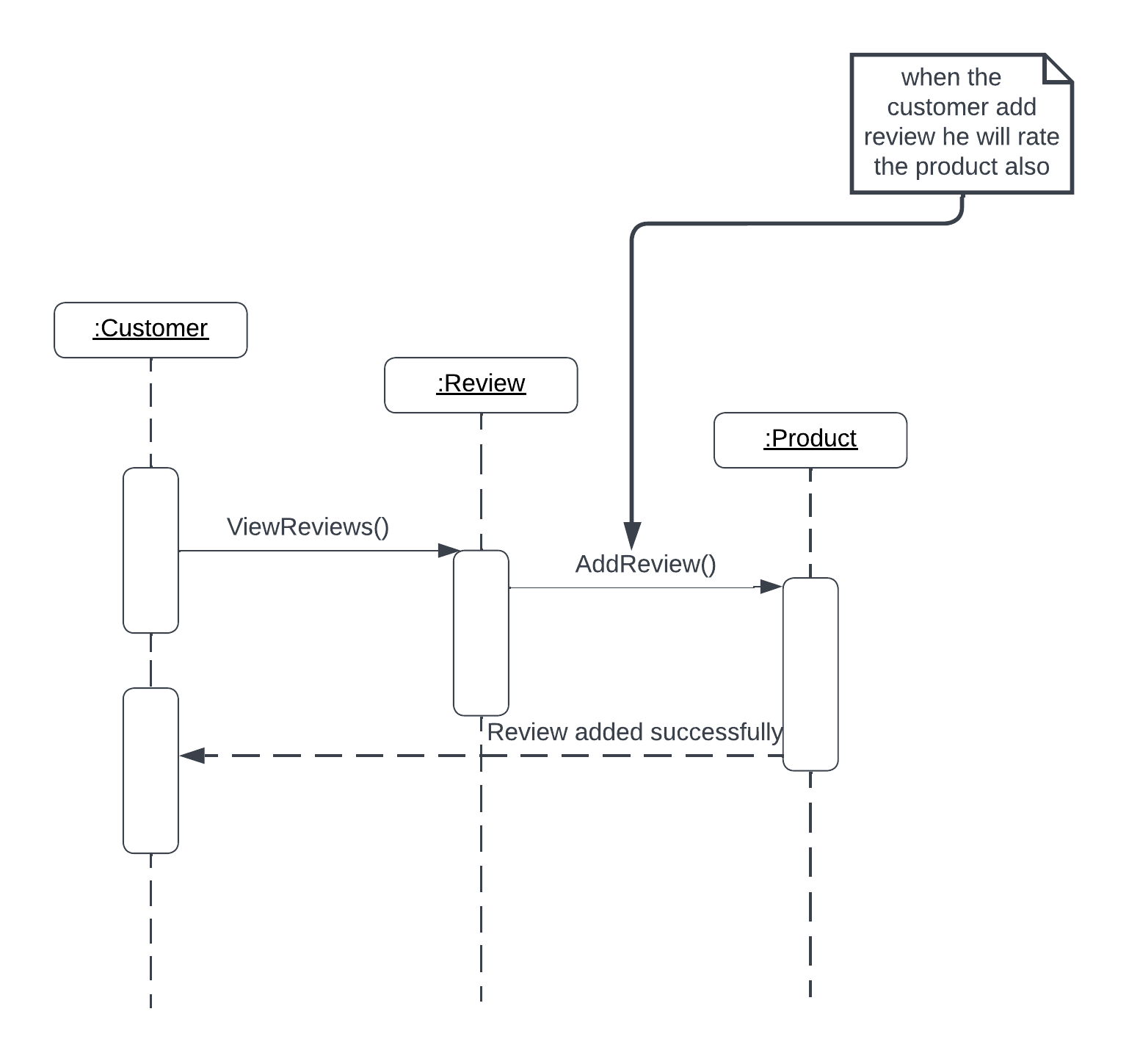
* 1. **System Architecture**
* **Model View Controller (MVC)**

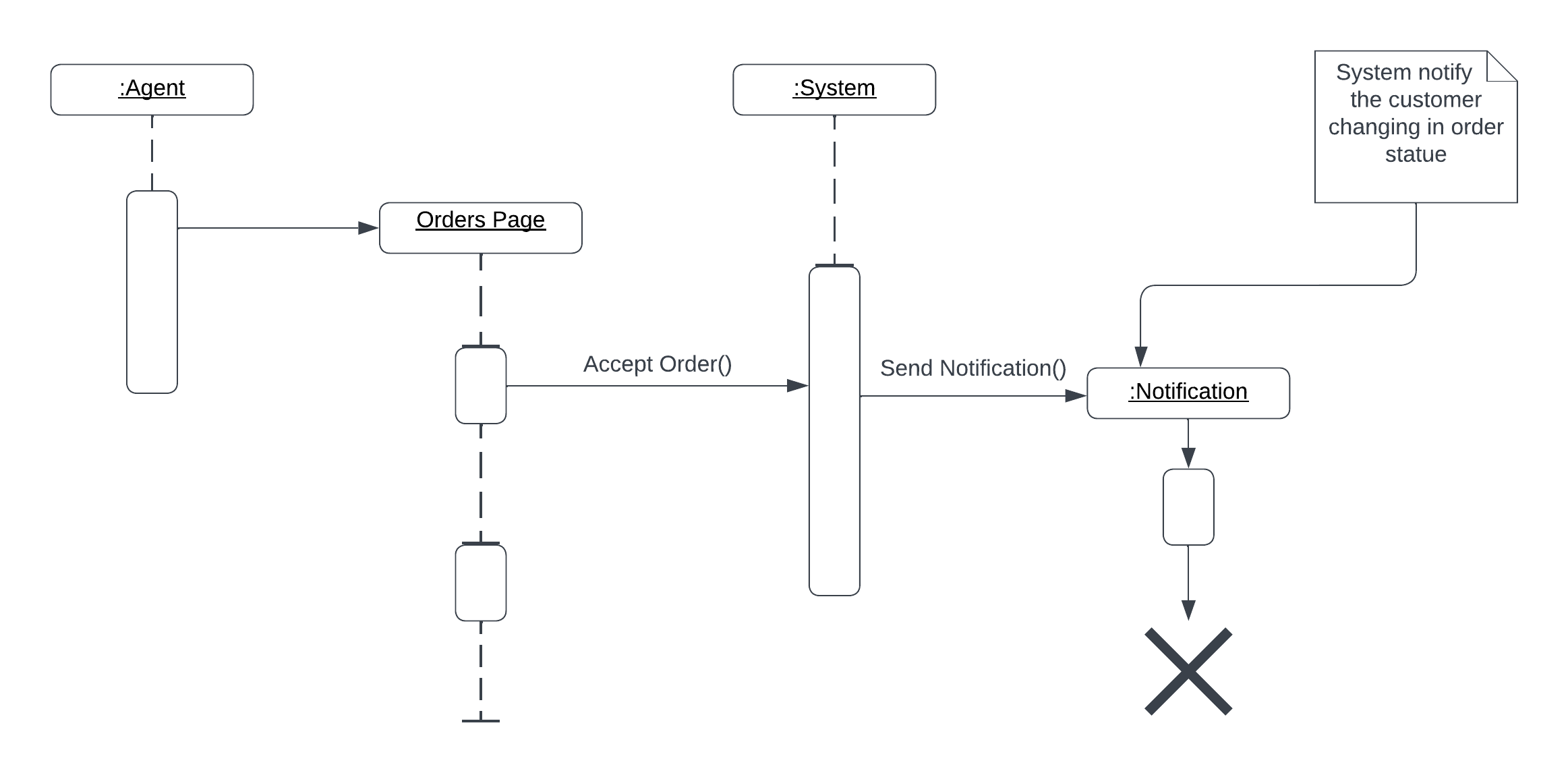


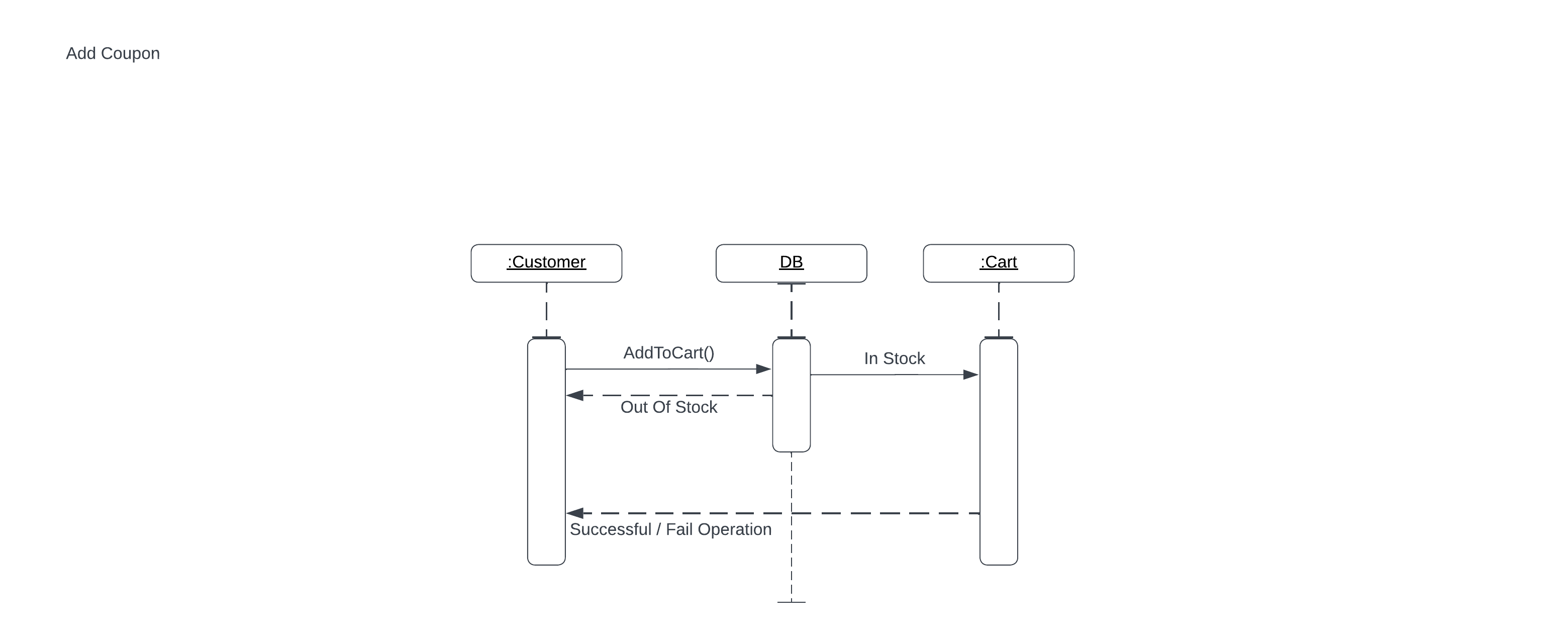
**3.9 Sequence Diagram**

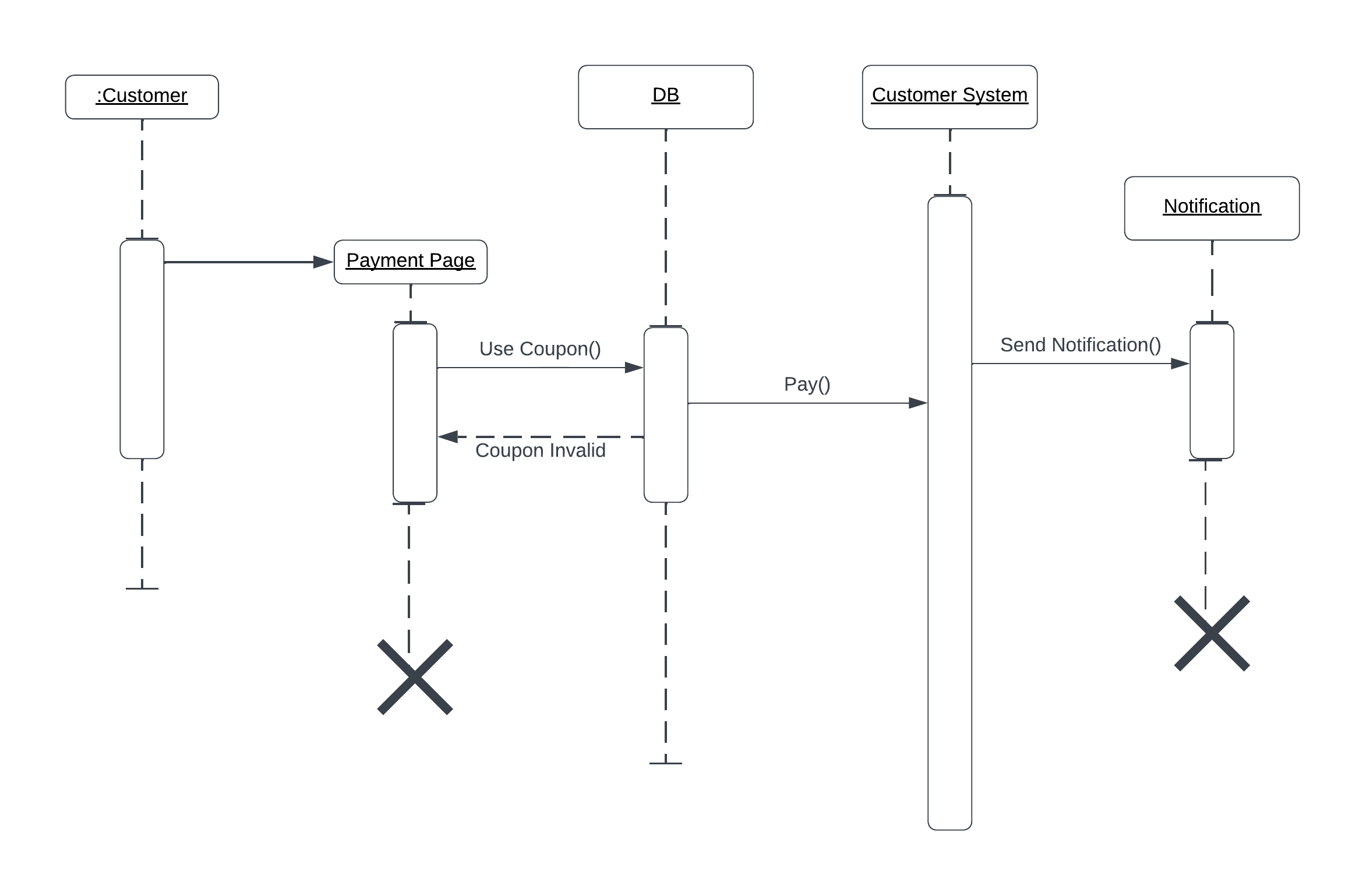
* **Some selected functions illustration by Sequence Diagram**

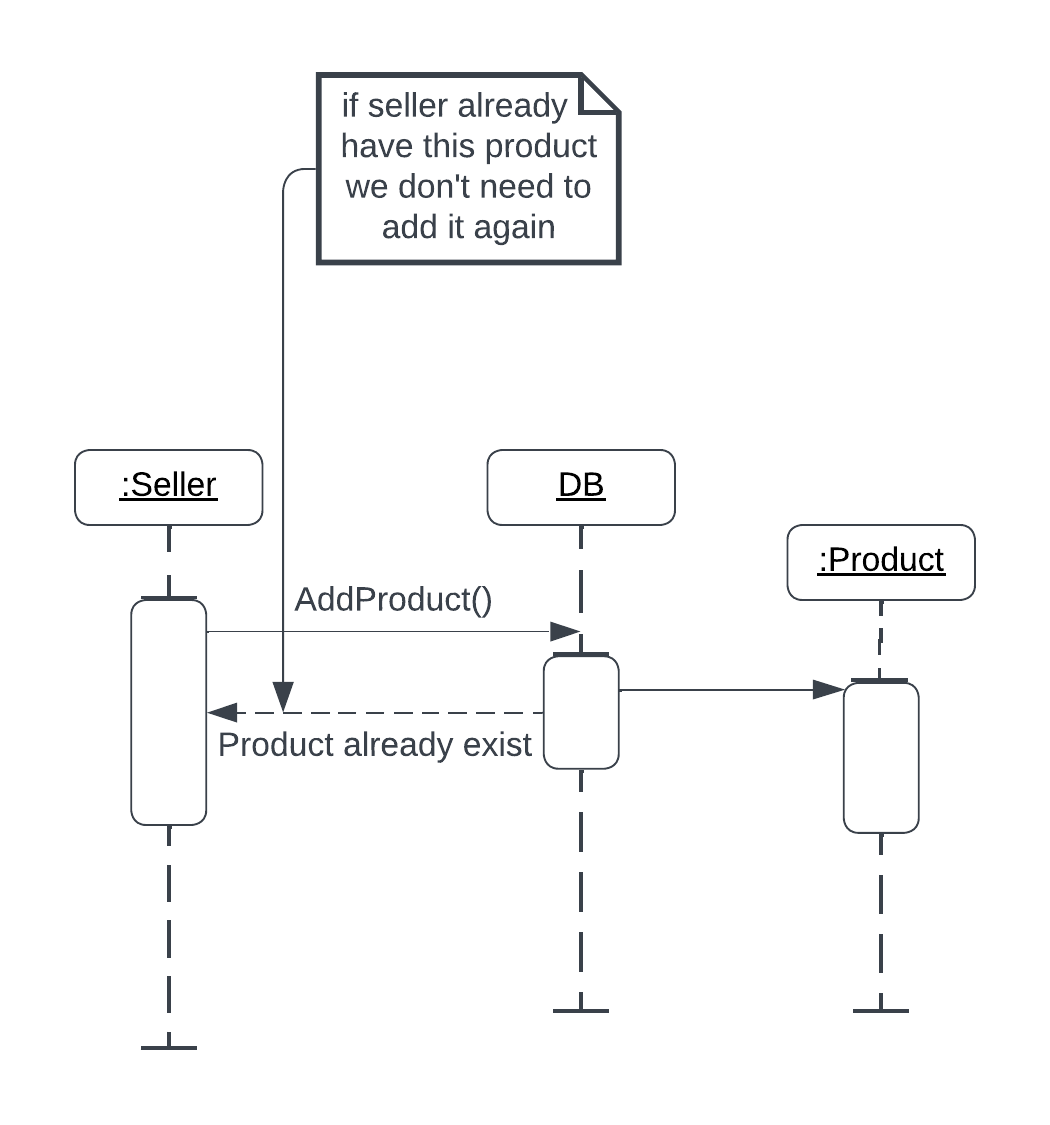
****

****

****

****

****

****