

1	هاجر سيد راشد	20211023
2	ثناء خيرى سيد محمد	20210247
3	جواهر ابراهيم على حسن	20210258
4	رودينا يحيى محمد	20210344
5	منى محمد عبد العزيز عامر	20210967
6	هدى ماهر محمد	20211031

DearCustomer Platform

Content:-

- 1)Description of the platform.....
- 2)Functional Requirements.....
- 3)Non-Functional Requirements.....
- 4)Use-Case diagram.....
- 5)Description of Use-Case diagram.....
- 6)Activity diagrams.....
- 7)Sequence diagrams.....
- 8)Class diagram without OCL.....
- 9)Class diagram with OCL.....
- 10).....ERD diagram
- 11).....SYS architecture
- 12).....Scope architecture

Description of the platform:-

"DearCustomer" is an e-commerce platform designed to provide users with a seamless shopping experience. Users have the ability to create accounts, log in, and access a variety of features tailored to enhance their shopping journey.

Upon accessing the platform, allowing them to browse through products specific to their preferences. To facilitate easier product discovery, users can utilize search bars and category filters.

Each product on the platform comes with detailed information including images, descriptions, prices, and available sizes/colors, ensuring users have all the necessary information before making a purchase decision. Users can add products to their shopping cart for later purchase and remove items if needed.

Furthermore, "DearCustomer" offers discounts for purchases exceeding a certain amount, incentivizing users to explore more products and enjoy savings on their orders. Users can conveniently choose from various payment methods including credit/debit cards, PayPal, and more.

Administrators have access to additional functionalities such as creating and managing categories for new products, adding, editing, or deleting products, and handling category deletions. Users also have the flexibility to manage their accounts by logging out or deleting their accounts from the platform.

"DearCustomer" aims to provide a user-friendly and efficient e-commerce experience, catering to both users and administrators with its comprehensive set of features.

Functional Requirements:-

1) Sign-up :

Users should be able to create accounts to access the platform.

2) Log-in :

Allow users to access their accounts on the platform.

3) Choose Category :

help customers find products easily through search bars, filters by category.

4) View details :

Customers should be able to view detailed information about each product, including images, descriptions, prices, and available sizes/colors.

5) Add to cart :

Customers should be able to add products to their shopping cart for later purchase.

6) Remove from cart :

Customer should be able to remove products from their shopping cart.

7) Get discount :

Customers can take a discount if they buy products worth more than 100\$.

8) Choose way to payment :

Customers should be able to choose from various payment methods such as credit/debit cards, PayPal, etc.

9) Create category :

Administrators should be able to create a category of the new products.

10) Delete category :

Administrators should be able to delete a category if not exist in it products.

11) Add product :

Administrators should be able to add a new product .

12) Edit product :

Administrators should be able to edit the information of a product.

13) Delete product :

Administrators should be able to delete a product .

14) Log-out :

Users should be able to log out from the platform.

Non-Functional Requirements:-

1) Performance :

The application should load quickly and respond promptly to user interactions to provide a seamless shopping experience.

2) Scalability :

The platform should be able to handle increasing numbers of users and products without a significant decrease in performance.

3) Reliability :

The platform should be reliable and available 24/7 with minimal downtime for maintenance or updates.

4) Security :

The platform should adhere to industry-standard security practices to protect user data, including encryption of sensitive information and secure authentication mechanisms.

5) Usability :

The platform should have an intuitive user interface and navigation structure to make it easy for users to find and purchase products.

6) Availability :

The platform should have high availability, with redundant systems and failover mechanisms to minimize downtime.

7) Error Handling :

The platform should handle errors gracefully and provide informative error messages to users when issues occur.

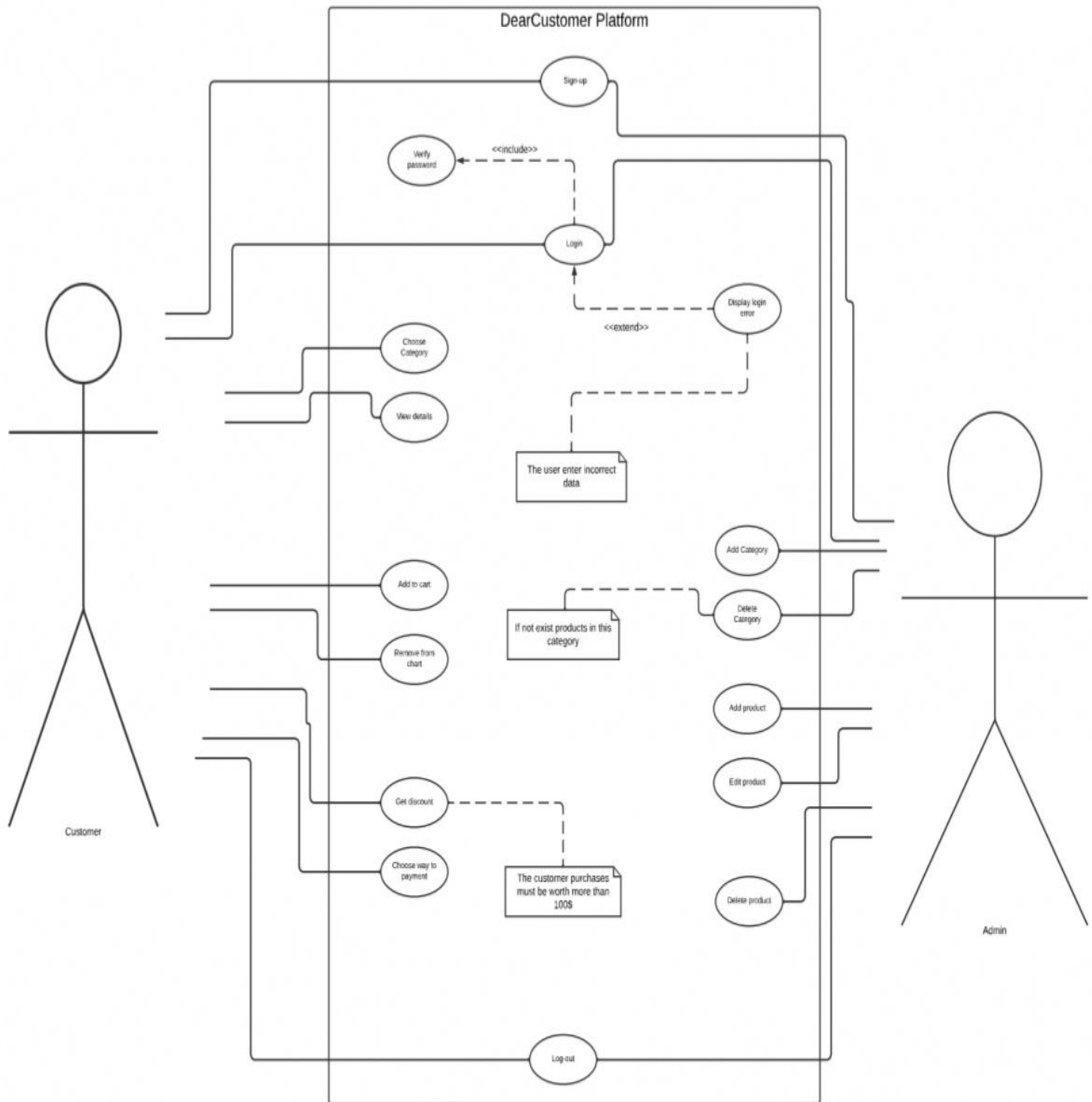
8) Mobile Responsiveness :

The platform should be optimized for mobile devices, with a responsive design that adjusts layout and functionality based on screen size.

9) Backup and Recovery :

The platform should have regular backups and a disaster recovery plan in place to minimize data loss in case of system failures.

Use-Case diagram:-



Description of Use-Case diagram:-

Use-case	Sign -in
Initiator	Customer & Admin
Pre-condition	The user must have access internet and a valid email address
Post-condition	The user will have a new account, with all features available to registered user
Main success scenario	The user enter registering data then will create the account
Goal	Allo user to create new account

Use-case	Log -in
Initiator	Customer & Admin
Pre-condition	The user must have an account on the website and enter correct information
Post-condition	The user can use the website, with all features available to the user
Main success scenario	The user enter correct information of his account then can use the platform.
Goal	Allo users access their accounts.

Use-case	Choose category
Initiator	Customer

Pre-condition	The customer must be logged -in to his account .
Post-condition	The system must display products of the category that customer choosed .
Main success scenario	The customer log in the website then choose the category, then the system display the products .
Goal	help customer find products easily through search bars, filters by category.

Use-case	View details
Initiator	Customer
Pre-condition	The customer must be logged -in to his account and choosed a store.
Post-condition	The system must display the details of the product that customer choosed .
Main success scenario	The customer log in the website then choose the store,then customer click on view details , then the system display the details of product .
Goal	Customers should be able to view detailed information about each product, including images, descriptions, prices, and available sizes/colors.

Use-case	Add to cart
Initiator	Customer
Pre-condition	The customer must be logged -in to his account and choosed a store and click on view details.
Post-condition	The system create a cart and add the product that customer choosed .
Main success scenario	The customer click on add to cart , then the system create a cart if this the product is the first and add the product to this cart.
Goal	Customers should be able to add products to their shopping cart for later purchase.

Use-case	Remove from cart
Initiator	Customer
Pre-condition	The customer must be logged -in to his account and have at least one product in his cart.
Post-condition	The system delete the product that customer choosed .
Main success scenario	The customer click on the cart , then remove the product from cart.

Goal	Customers should be able to remove products from their shopping cart.
------	---

Use-case	Get discount
Initiator	Customer
Pre-condition	The customer must be logged -in to his account and choosed a store and he/she buy products woth more than 100\$.
Post-condition	The customer will take a discount on his purchases .
Main success scenario	The customer add products to the cart and their worth more than 100\$.
Goal	Customers can take a discount if they buy products woth more than 100\$.

Use-case	Add category
Initiator	Admin
Pre-condition	The admin must be logged -in to his account .
Post-condition	The system will add this category to the customer.
Main success scenario	The admin enter the name of the new category then the system display this category to the customer.
Goal	Administrators should be able to create a category of the new products.

Use-case	Delete category
Initiator	Admin
Pre-condition	The admin must be logged -in to his account and this category mustn't have products in itself.
Post-condition	The system will delete this category and disappear from the customer.
Main success scenario	The admin determine the category that went to delete, then system disappear this category from the customer.
Goal	Administrators should be able to delete a category if not exist in it products.

Use-case	Add product
----------	-------------

Initiator	Admin
Pre-condition	The admin must be logged -in to his account .
Post-condition	The system will add this product in it's category and display it to the customer.
Main success scenario	The admin add a product and determine the category that this product belongsto it , then system display this product to the customer.
Goal	Administrators should be able to add a new product .

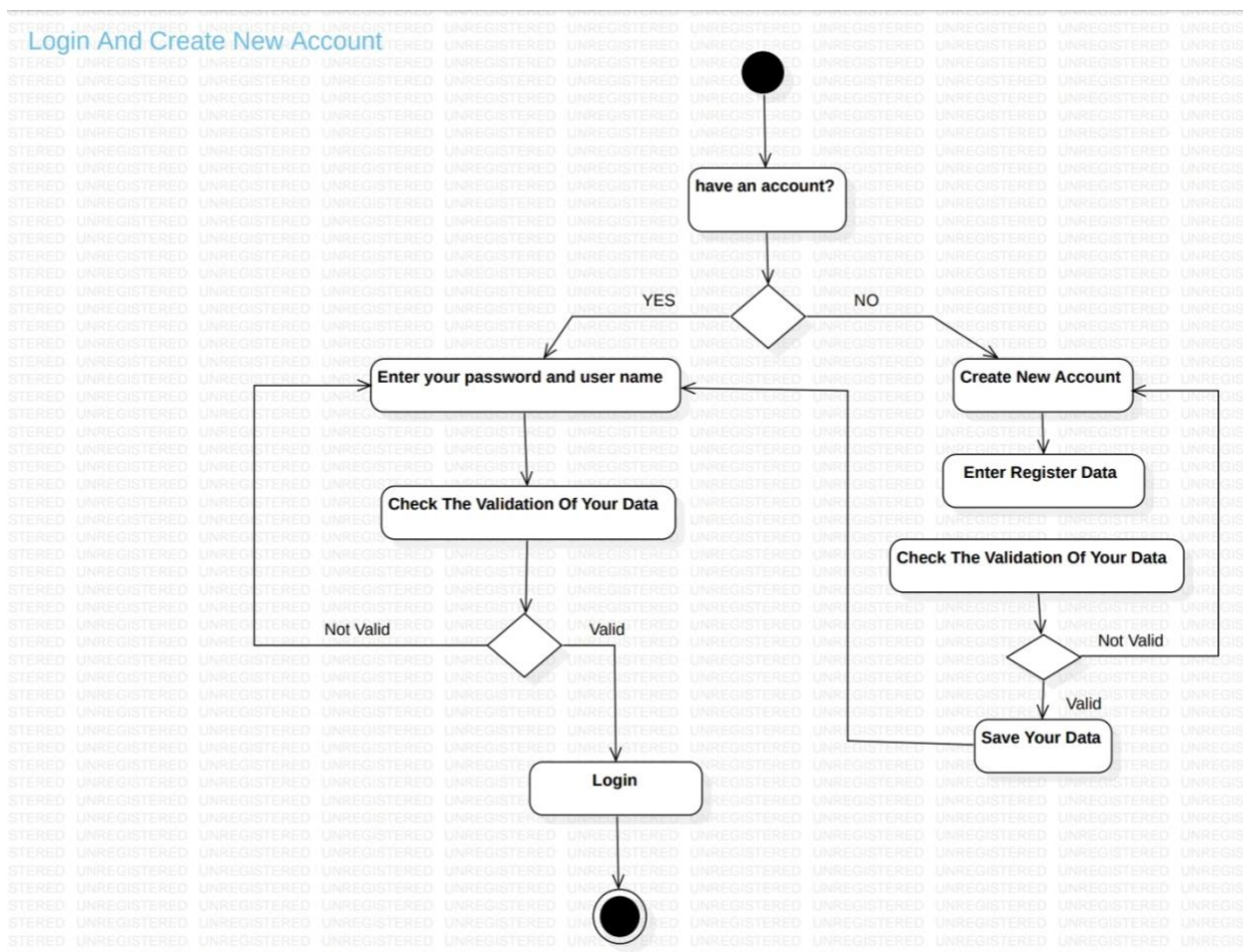
Use-case	Edit product
Initiator	Admin
Pre-condition	The admin must be logged -in to his account .
Post-condition	The system will display the new information about the product that admin edit it ,and display it to the customer.
Main success scenario	The admin determine the product that went to edit, then system display the new information to the customer and delete the old information of this product.
Goal	Administrators should be able to edit the information of a product.

Use-case	Delete product
Initiator	Admin
Pre-condition	The admin must be logged -in to his account .
Post-condition	The system will disappear the new product that admin choosed,and disappear it from the customer.
Main success scenario	The admin determine the product that went to delete, then system disappear the product from the customer and delete this product.
Goal	Administrators should be able to delete a product .

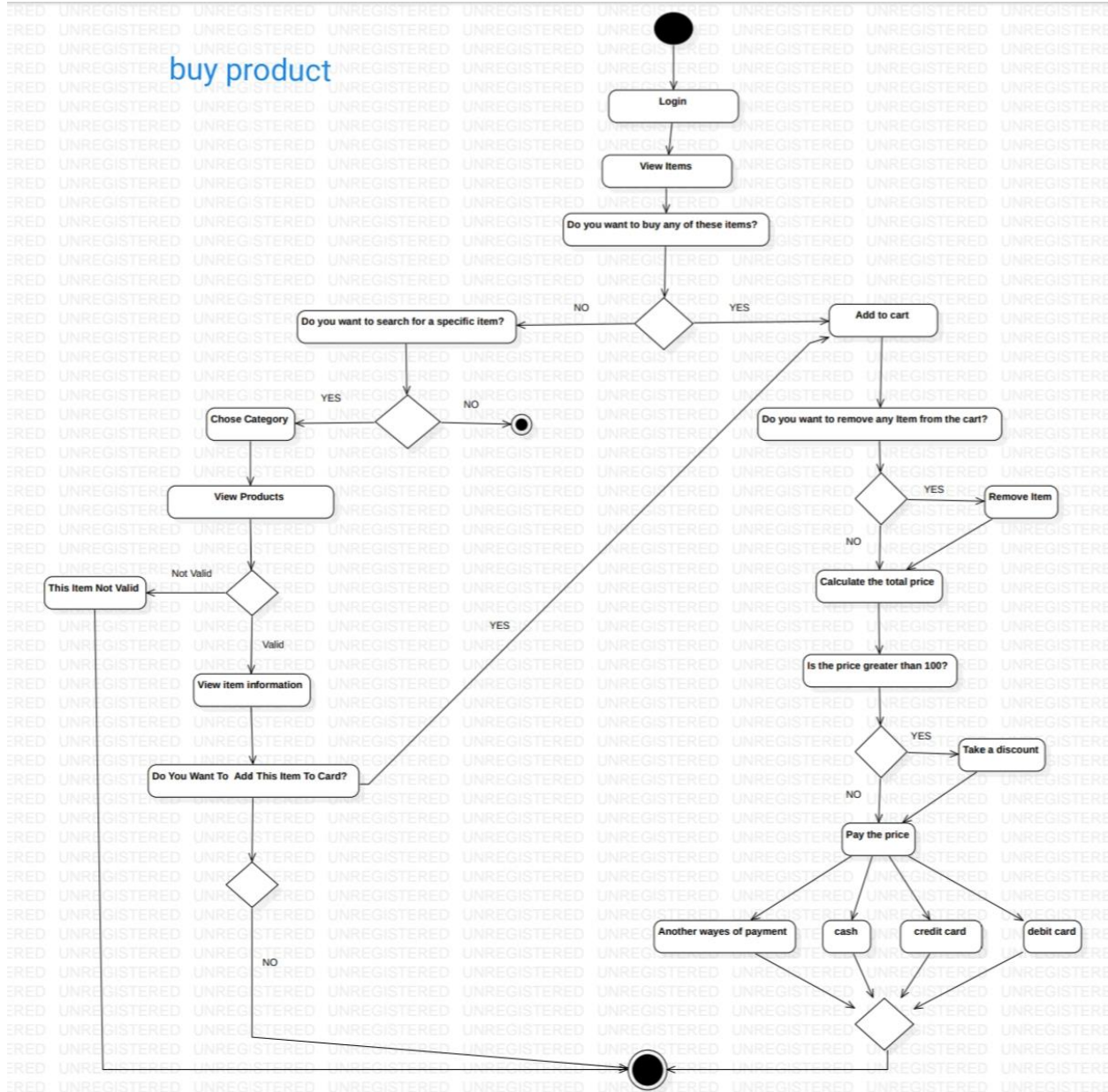
Use-case	Log -out
Initiator	Customer & Admin
Pre-condition	The user must be logged -in to his account .

Post-condition	The user cannot use the website and cannot access his/her features.
Main success scenario	The user logged-in to the website and want to log-out from it.
Goal	Users should be able to log out from the website.

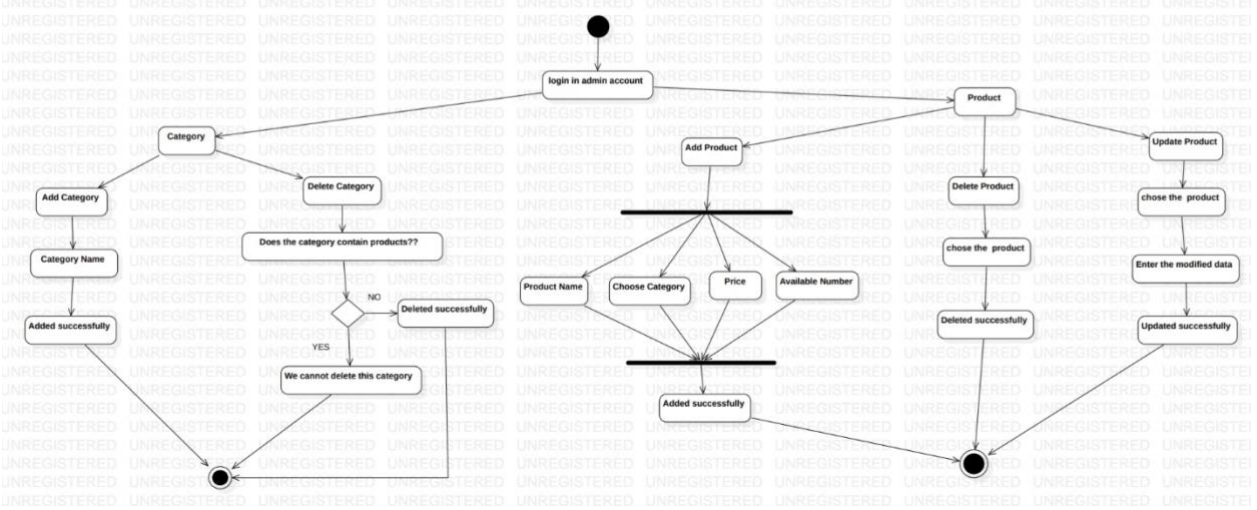
Activity diagrams:-



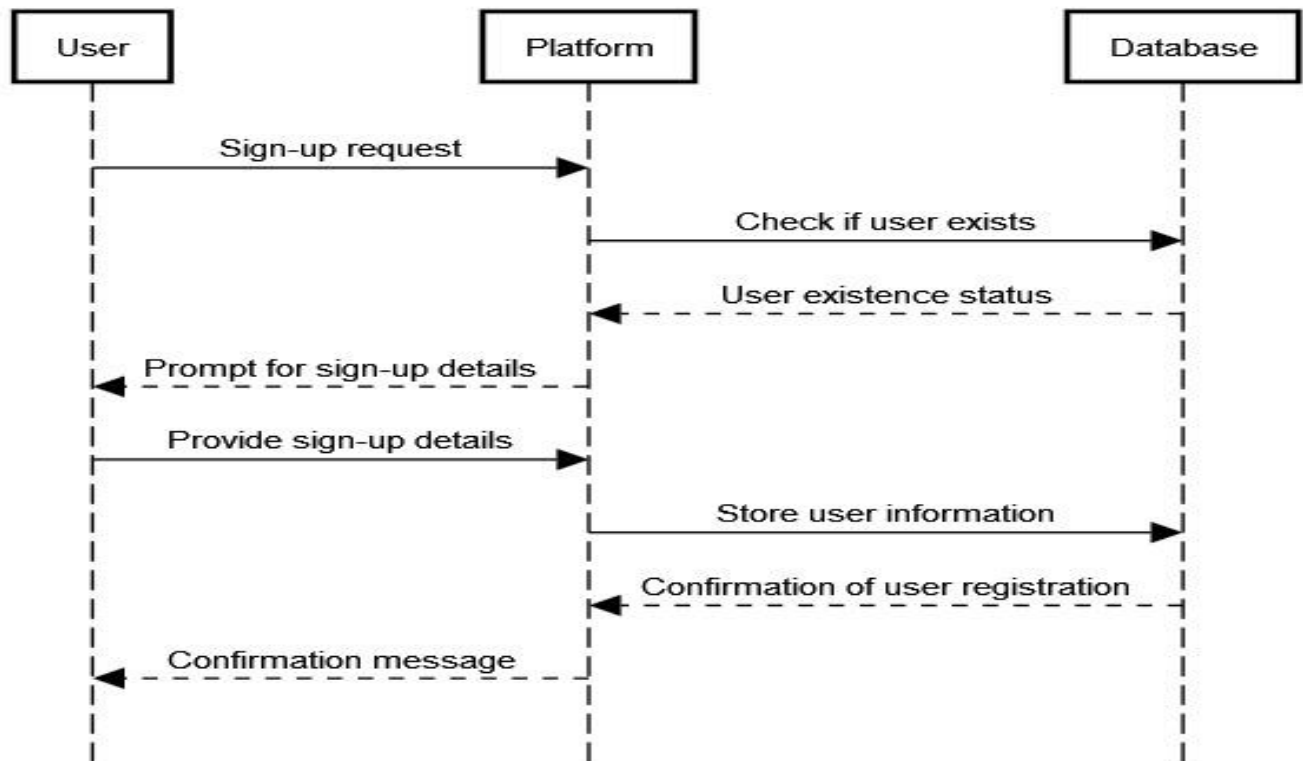
buy product

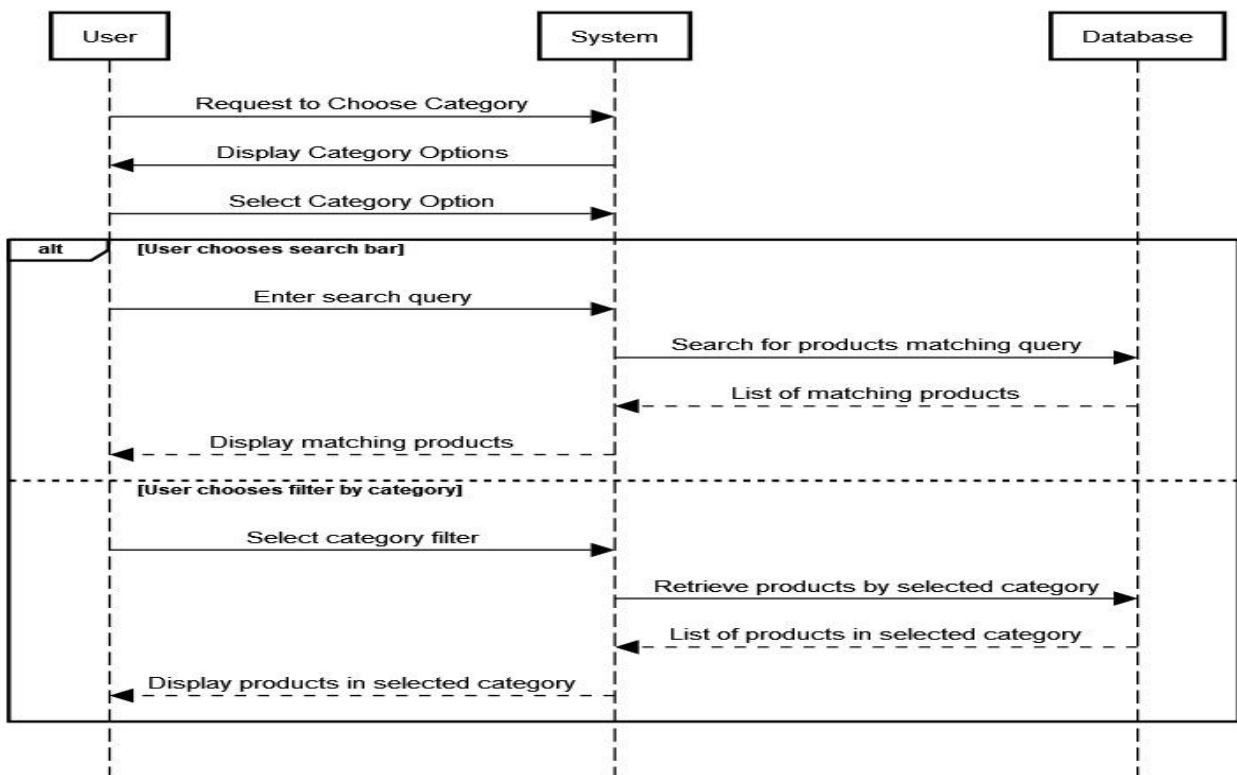
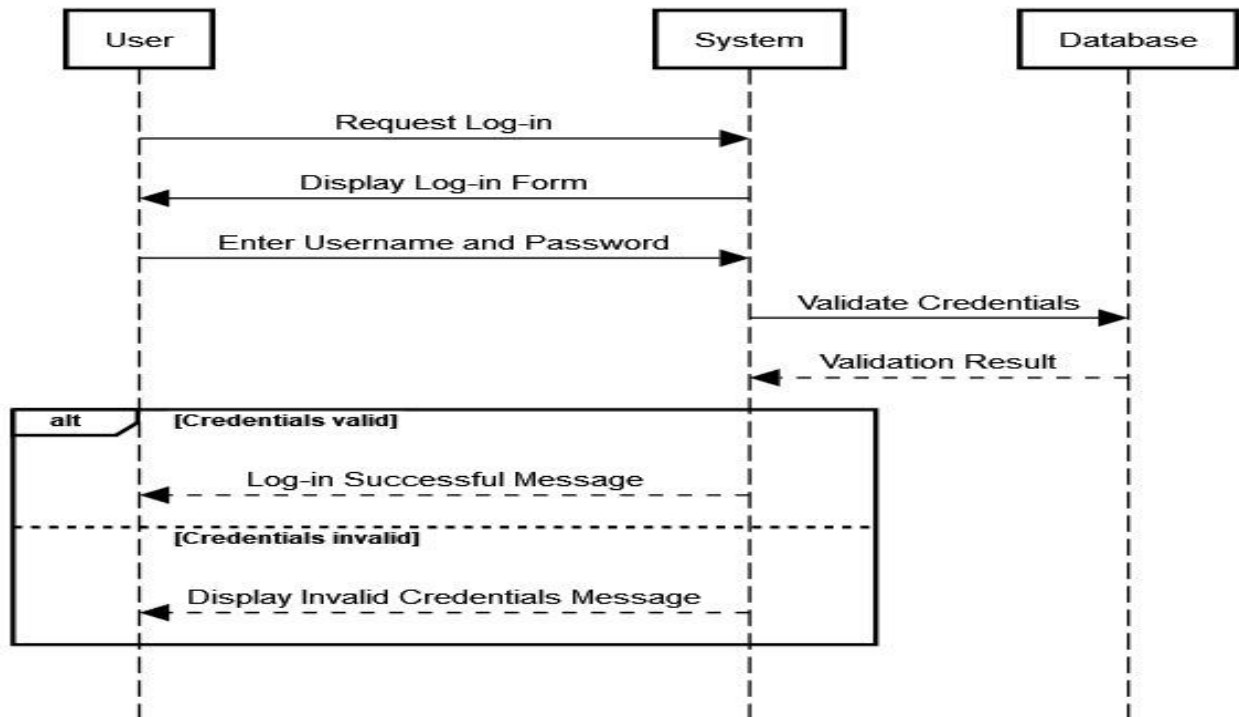


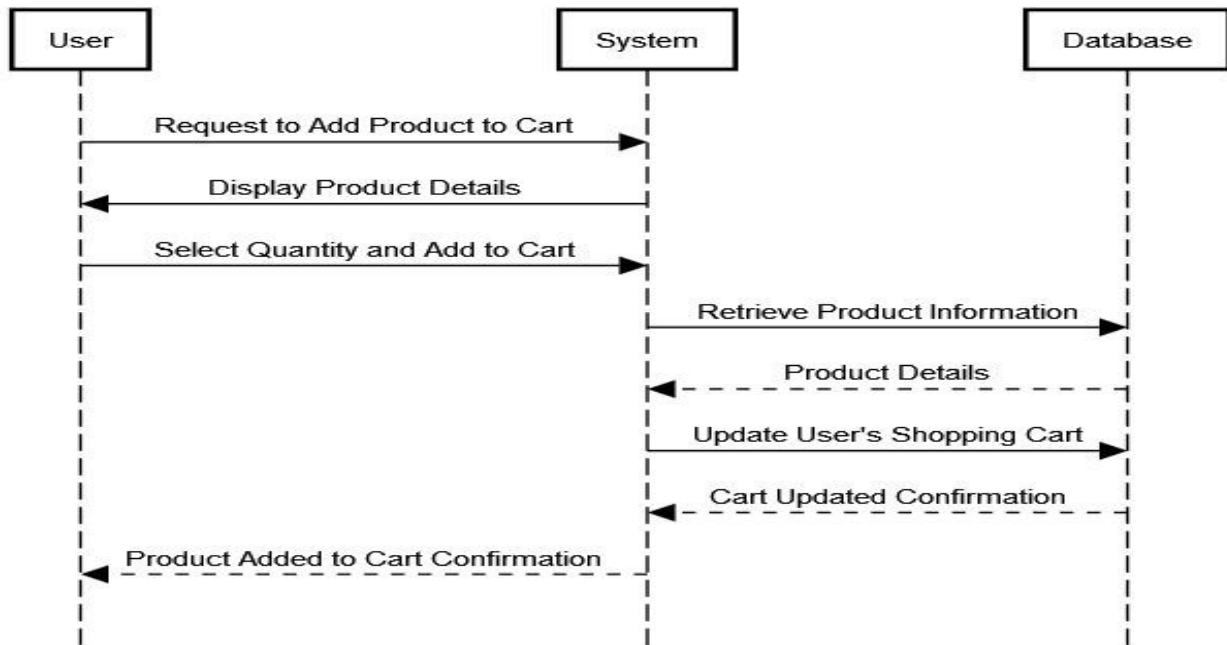
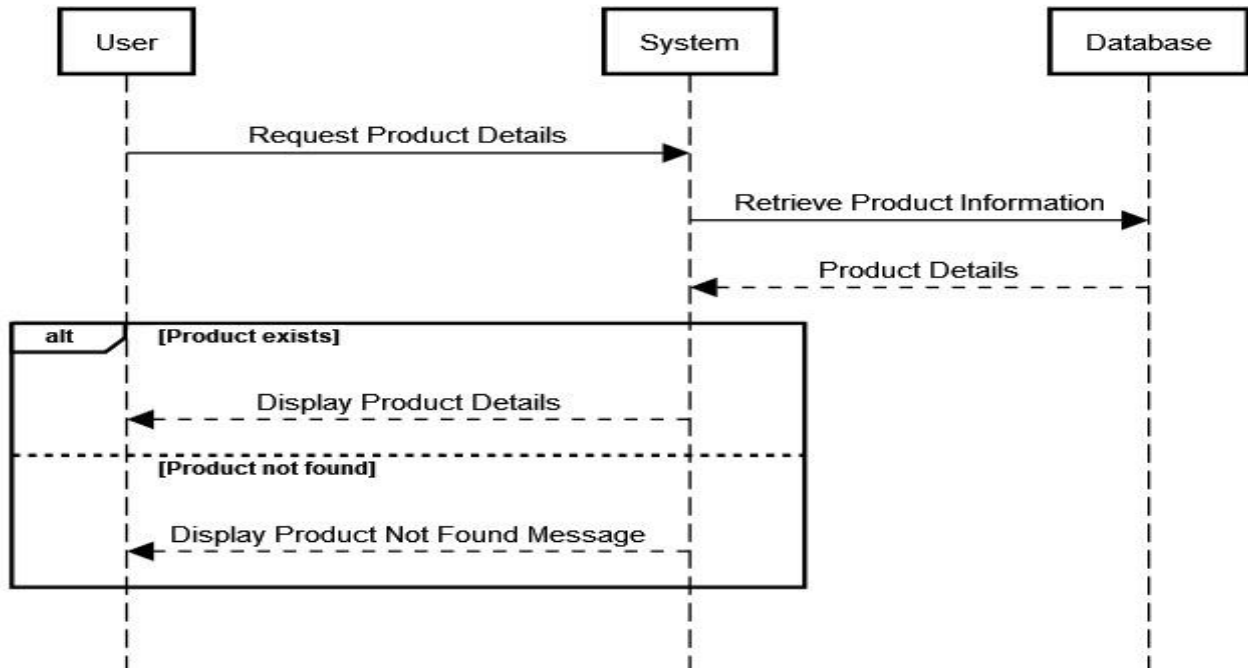
Category and Product

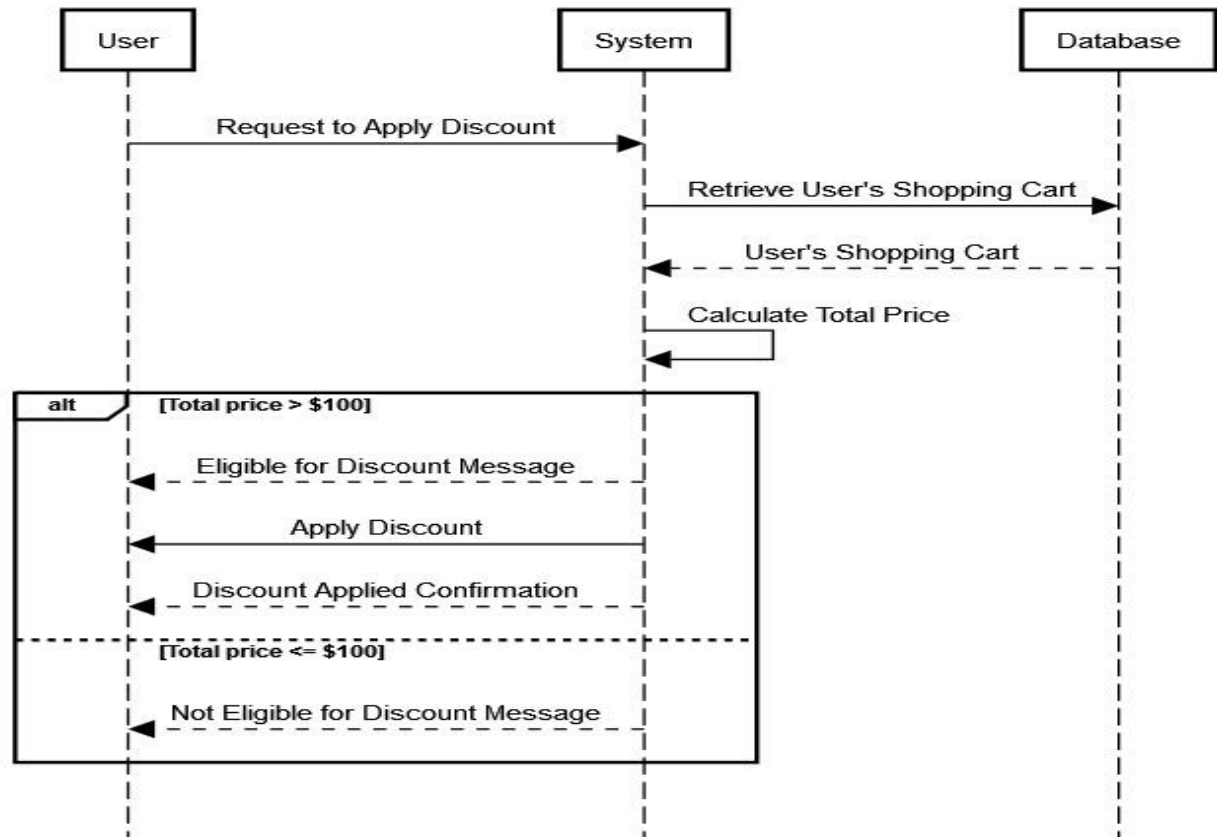
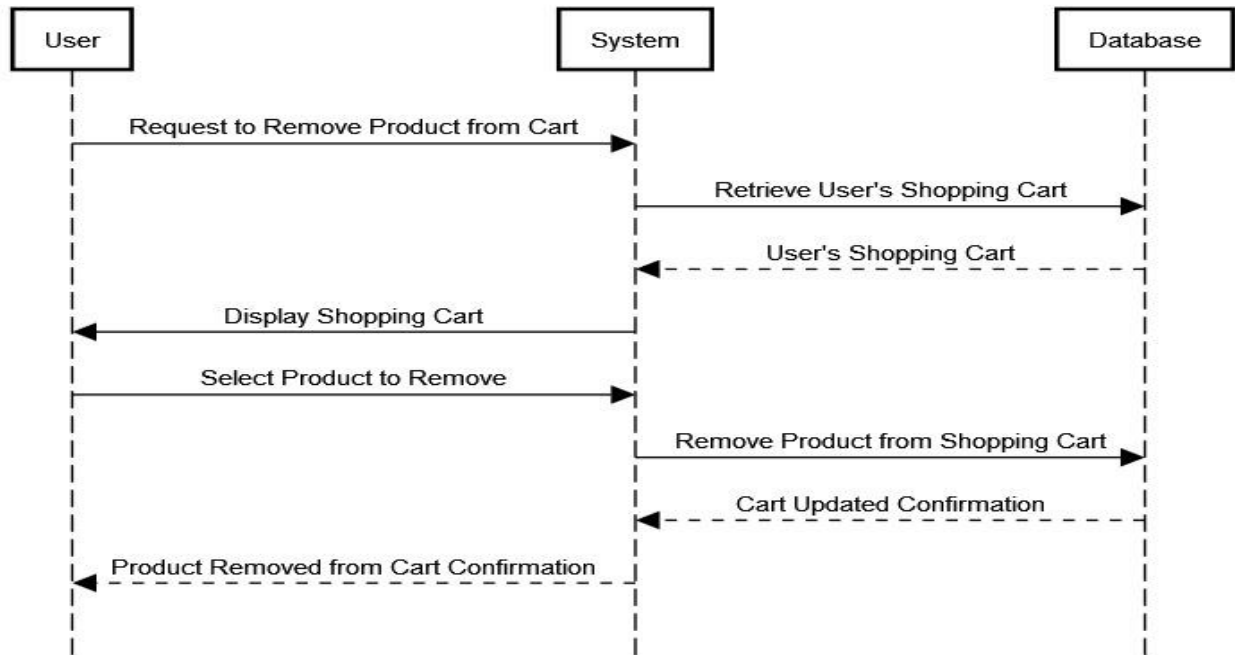


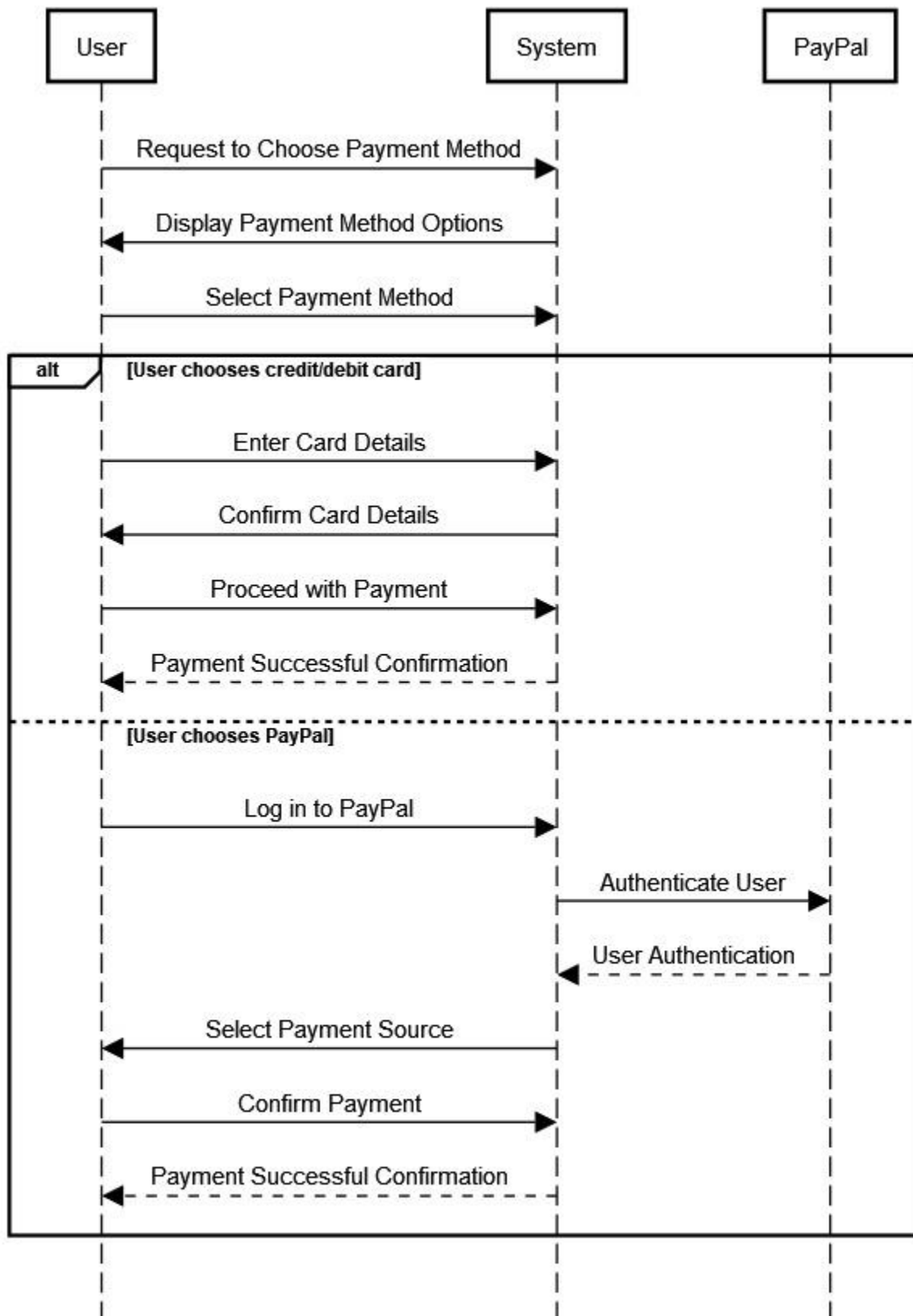
Sequence diagrams:-

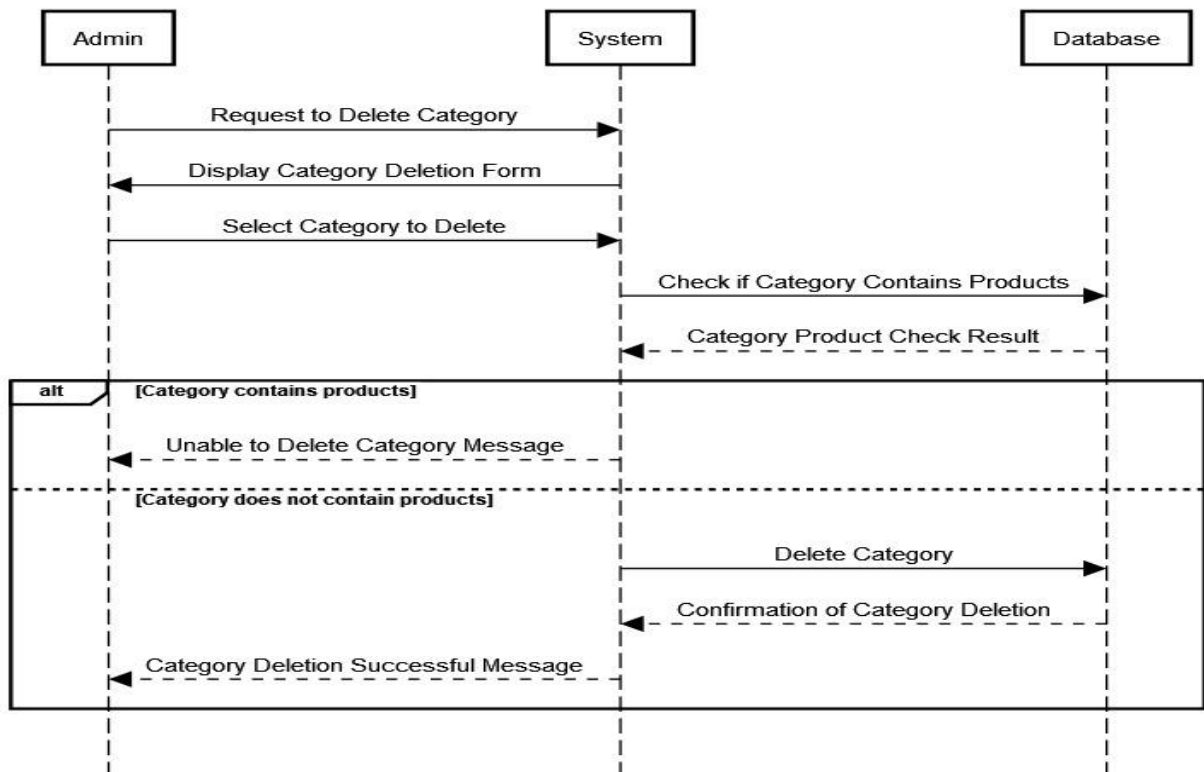
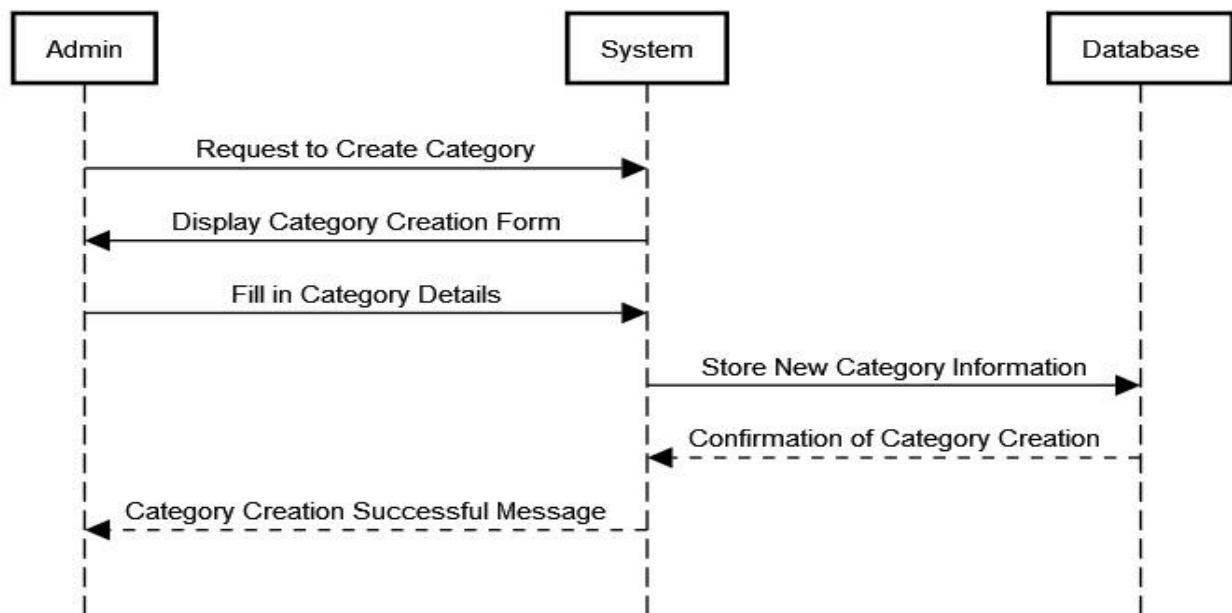


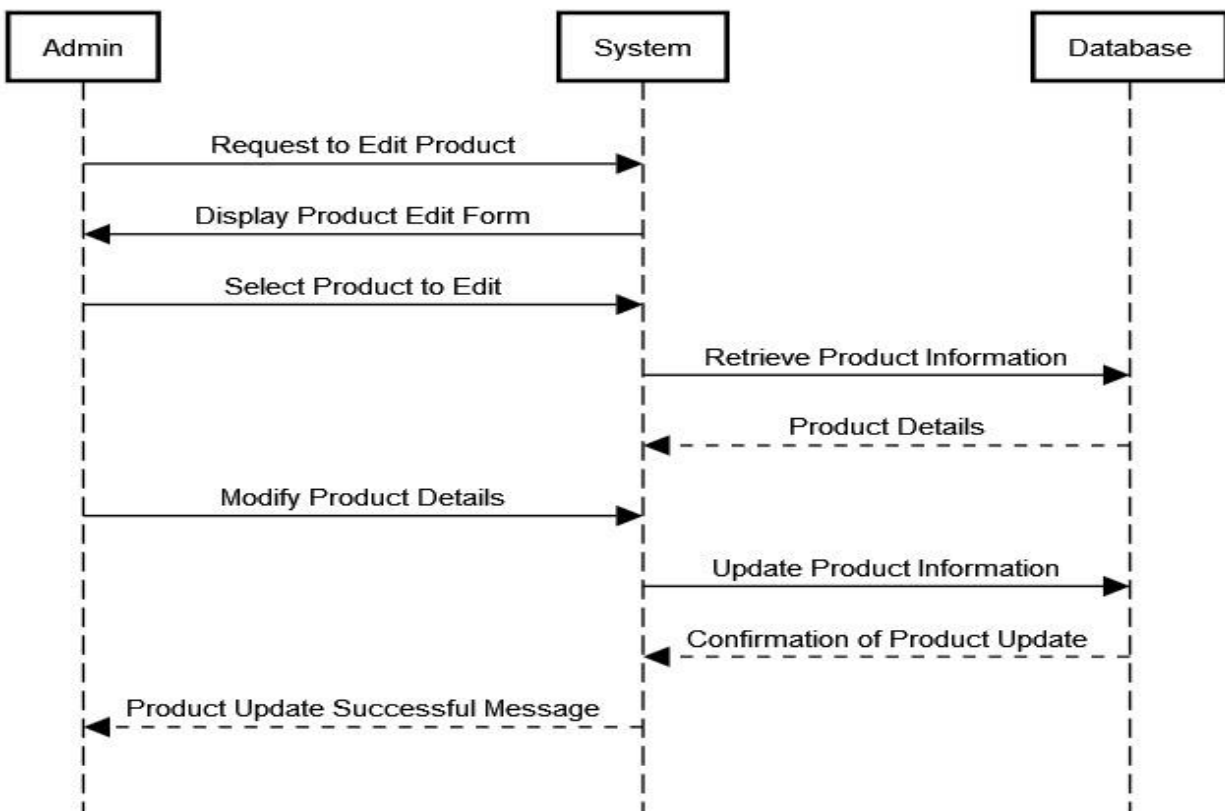
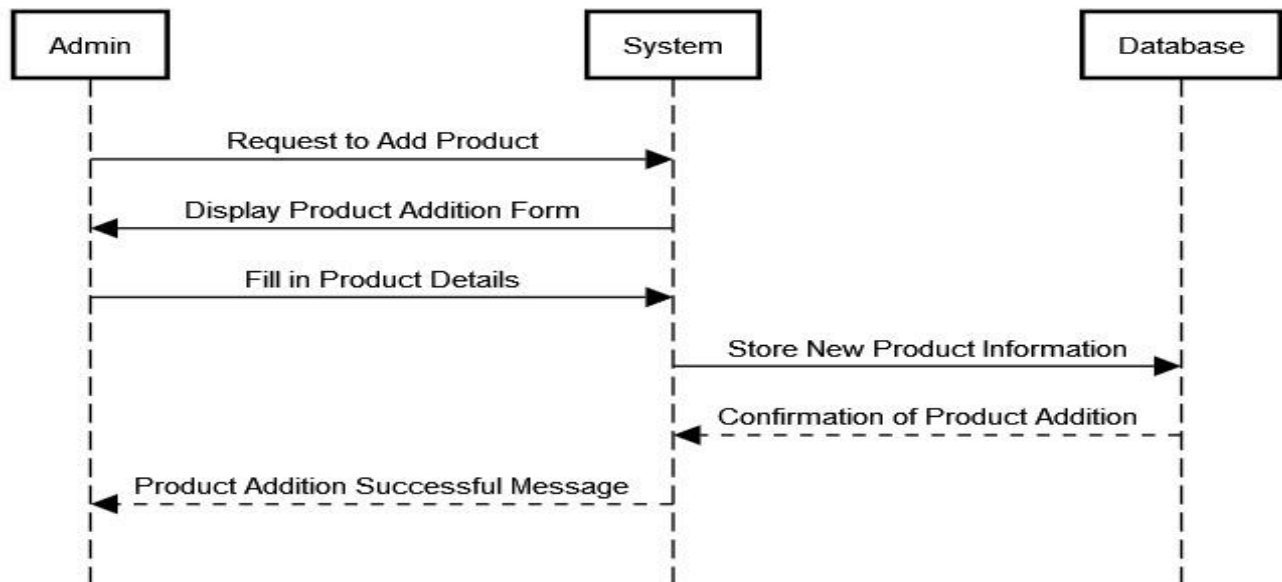


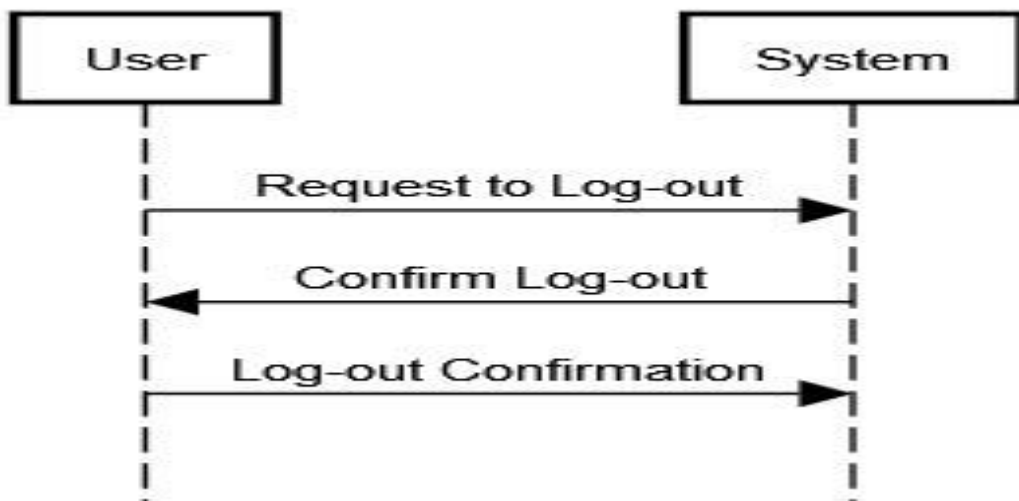
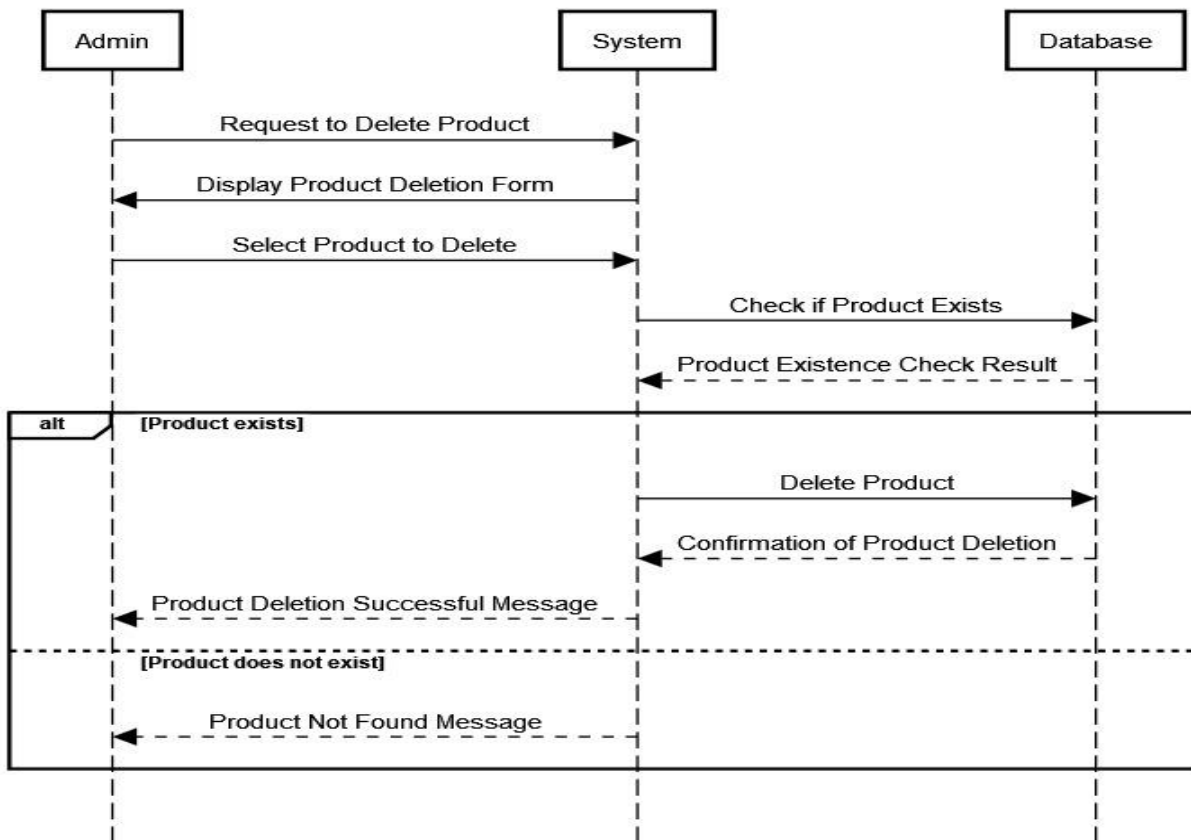




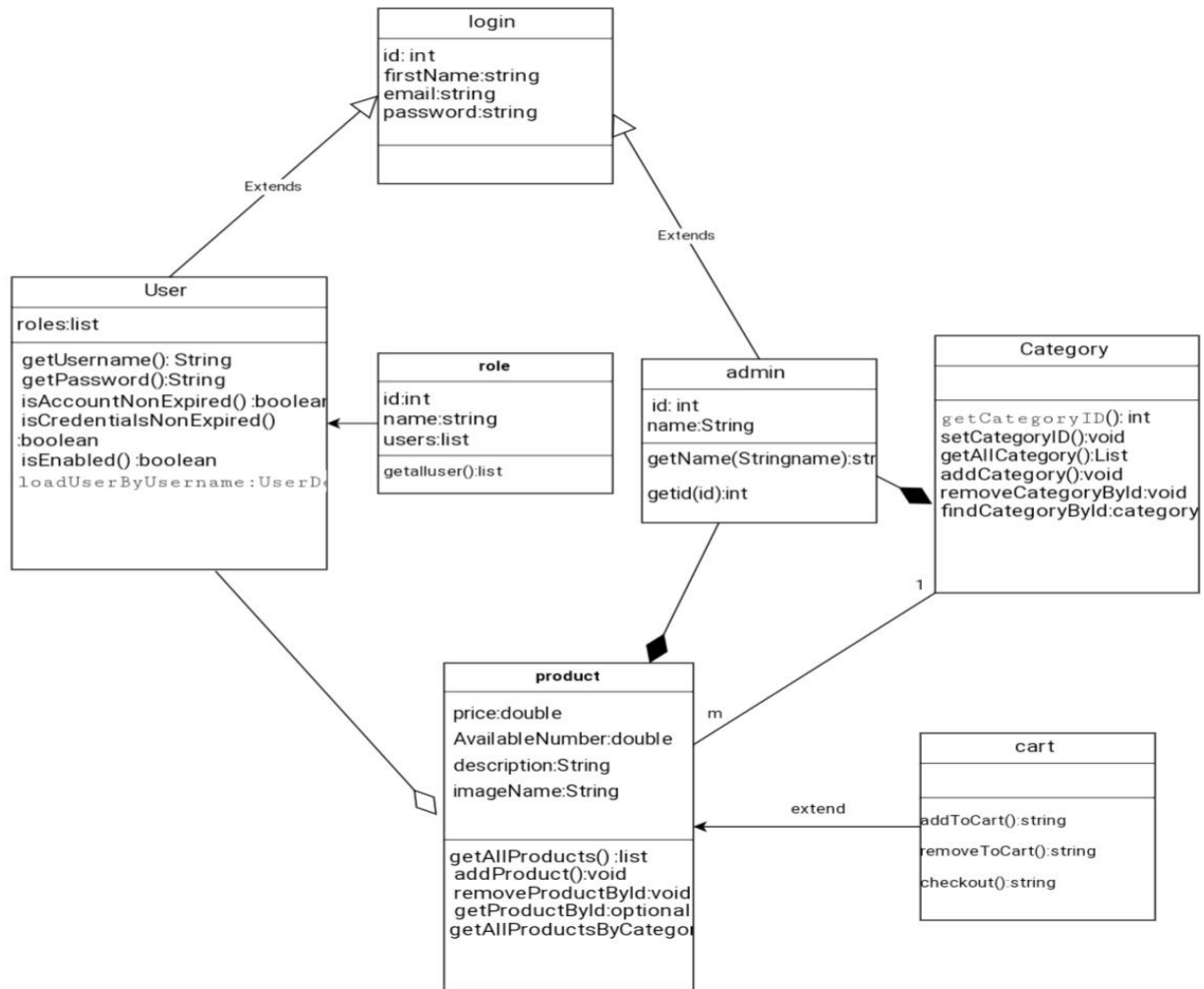




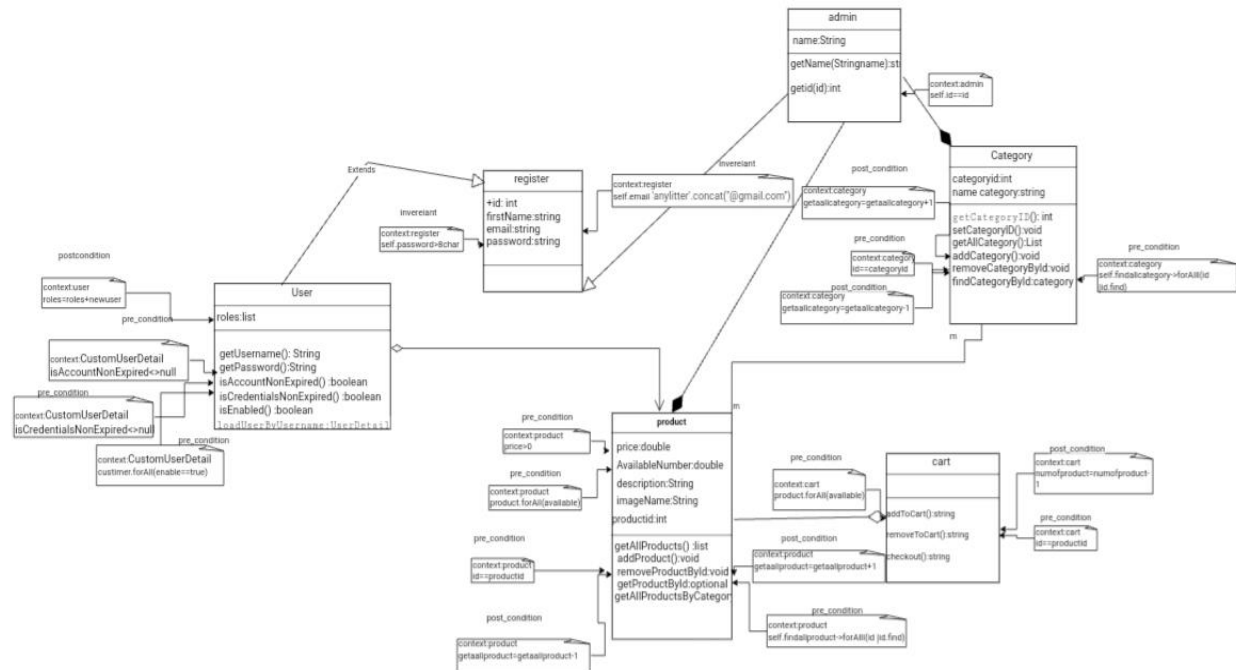




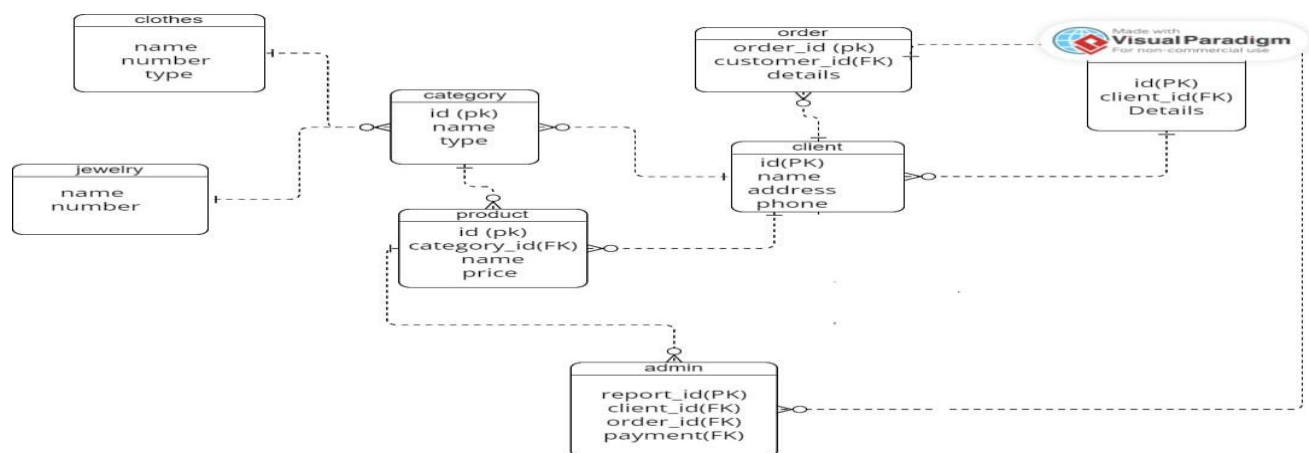
Class diagram without OCL:-



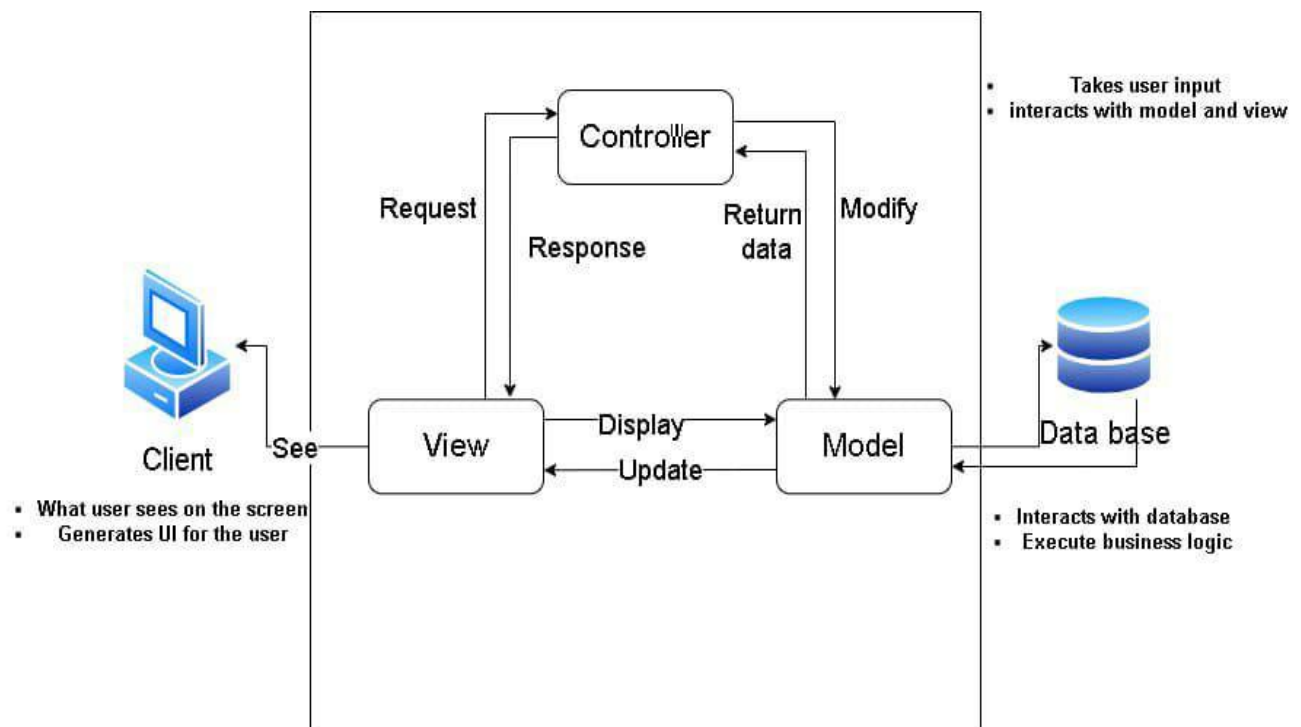
Class diagram with OCL:-



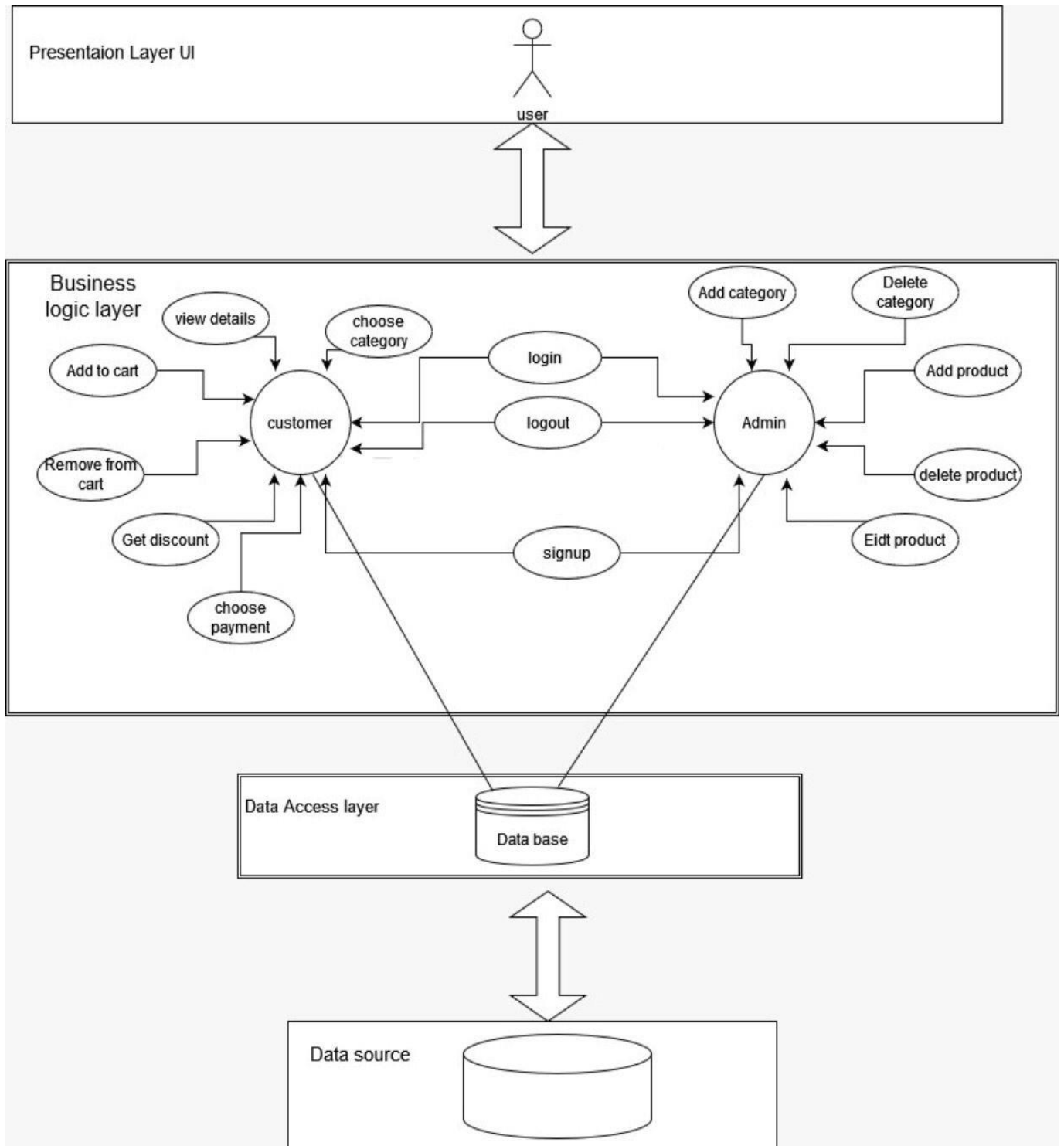
ERD diagram:-



SYS architecture:-



Scope architecture:-



The scope architecture of the "DearCustomer" e-commerce platform can be outlined as follows:

1. User Interface (UI):

- The UI is the front-facing component of the platform, providing users with an intuitive and visually appealing interface to interact with.
- It includes features such as user registration, login, product browsing, cart management, checkout, and account settings.
- The UI should be responsive and accessible across different devices and screen sizes.

2. Backend Services:

- Backend services handle the core functionality of the platform, including user authentication, product management, cart management, payment processing, and account management.
- These services are responsible for processing user requests, interacting with the database, and executing business logic.

- They should be scalable, robust, and secure to handle concurrent user interactions and ensure data integrity.

3. Database Management:

- The database stores all essential data for the platform, including user accounts, product information, shopping cart details, and transaction records.
- It should be designed to efficiently store and retrieve data, ensuring fast performance and scalability.
- Proper data modeling and indexing are essential to support complex queries and maintain data consistency.

4. Integration with Third-Party Services:

- Integration with third-party services is required for functionalities such as payment processing (e.g., credit/debit card processing, PayPal integration), email notifications, and analytics.
- APIs provided by external services should be utilized to seamlessly integrate their features into the platform.

5. Administrative Dashboard:

- An administrative dashboard allows administrators to manage the platform efficiently.

- It provides functionalities for managing products (add, edit, delete), categories, user accounts (suspend, delete), viewing analytics, and generating reports.

- Access to the administrative dashboard is restricted to authorized administrators through secure authentication mechanisms.

6. Security and Compliance:

- Security measures should be implemented throughout the platform to protect user data, prevent unauthorized access, and mitigate security threats.

- Compliance with data protection regulations (e.g., GDPR, CCPA) and industry standards (e.g., PCI DSS for payment processing) is essential to maintain user trust and avoid legal issues.

7. Scalability and Performance:

- The architecture should be designed for scalability to accommodate growing user traffic and data volume.

- Techniques such as load balancing, caching, and horizontal scaling should be employed to ensure optimal performance and responsiveness.

8. Monitoring and Maintenance:

- Monitoring tools and logging mechanisms should be integrated to track system performance, identify issues, and troubleshoot errors.

- Regular maintenance activities such as database backups, software updates, and security patches should be performed to ensure the stability and security of the platform.