Diagon Alley

Overview

Project Name	Diagon Alley.
Project Members	1. Ahmed Tarek Elabd 2. Menna Salem Mahmoud 3. Rwan Alaa Matter 4. Shahd Mohamed Samir 5. Afaf Ibrahim Elnagar 6. Yomna Mohamed Emara
Background	Diagon Alley is an Art Gallery Store that will offer users an easy, interactive way to browse, purchase various artworks and learn about famous artists. The website will include e-commerce functionalities, art information, and personalized recommendations.
Goals	Provide a seamless shopping experience, secure transactions, and efficient inventory and order management.

Table Content :

REQUIREMENTS

Functional Requirements

- ## User Functional Requirements :

User Registration and Authentication	• **Login/Sign-up**: Users can create an account or log in with an email and password.
Product Browsing and Search	• **Product Categories**: Users can browse Art Product based on categories (Paints, Statues, Photography). • **Search and Filter**: Users can search for plants and apply filters like price, Type (Classic, Modern) • **Product Details**: Each Product has a detailed page with Images, Description, Price, and Size.
Shopping Cart and Checkout	• **Add to Cart**: Users can add art products to their cart. • **View Cart**: Users can view and modify the cart before checkout. • **Checkout Process**: Users can complete a checkout with options for delivery or pickup, and view shipping charges.
Payment and Order Management	• **Payment Methods**: Integrate secure payment gateways (credit card, PayPal, Strip...), or can pay cash on deliver. • **Order Tracking**: Users can track the status of their orders (processing, shipped, delivered) • **Order History**: User can see art products of all other orders he make before .

- **Admin Functional Requirements :**

User management	• **View Users**: Admin can view a list of registered users with details like username, email, phone number and number of orders. • **Deactivate/Activate Users**: Admin can deactivate user accounts as needed
Product Management	• **Add New Product**: Admin can add new product with details like name, price, description, size as width and height, category, sub category and image. • **View Products**: Admin can view products details like image, name, category, sub category and price • **Delete Product**: Admin can remove a product from the store.
Order Management	• **View Orders**: Admin can view a list of all orders with details like username of the person who purchase, Purchased Artworks, Date of the order and the status as pending or done • **Cancellations**: Admin can Cancel orders as needed.

Non-Functional Requirements

Security	• **Data Protection:** Secure user information and payment data with HTTPS and encryption . • **Regular Backups:** Perform regular backups to prevent data loss. • **Authentication**: Implement secure user authentication mechanisms to ensure only verified users have access. • **Authorization**: Role-based access control should be used to limit access based on user roles (admin, registered user). Specific actions and data should only be accessible to authorized users.

Usability	- **Responsive Design:** The website should be accessible on various devices (desktop, tablet, mobile). - **Intuitive Navigation:** Simple, user-friendly navigation to ensure easy access to products and information.
Availability	- **Reliability:** The website should have 99.9% uptime. - **Error Handling:** Clear error messages if any issues arise during browsing or checkout.
Performance	- **Loading Speed:** Pages should load within few seconds to ensure a smooth user experience. - **Scalability:** The website should handle increased traffic during high-demand periods.
Maintainability	- **Documentation:** Provide documentation for future updates and bug fixes. - **Modular Code Structure:** Use a modular code structure to simplify maintenance and feature updates.
SEO	- **SEO-Friendly:** Optimize for search engines to increase visibility.

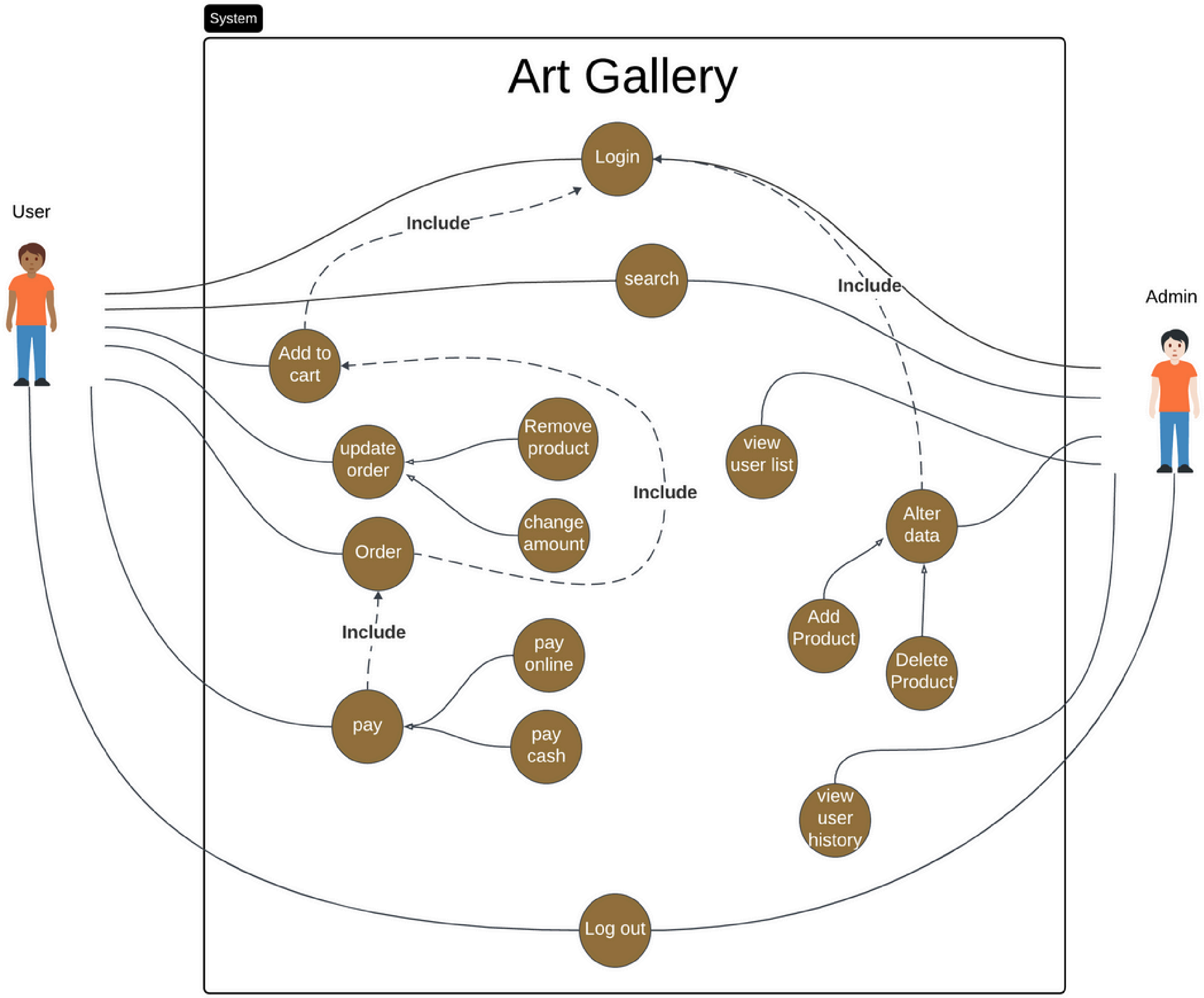
System Requirements

Frontend	- React.js, JavaScript , Bootstrap & Tailwind.css for Responsive Design.
Backend	- ASP.NET Core (Web API),Entity Framework
Database	- SQL Server.

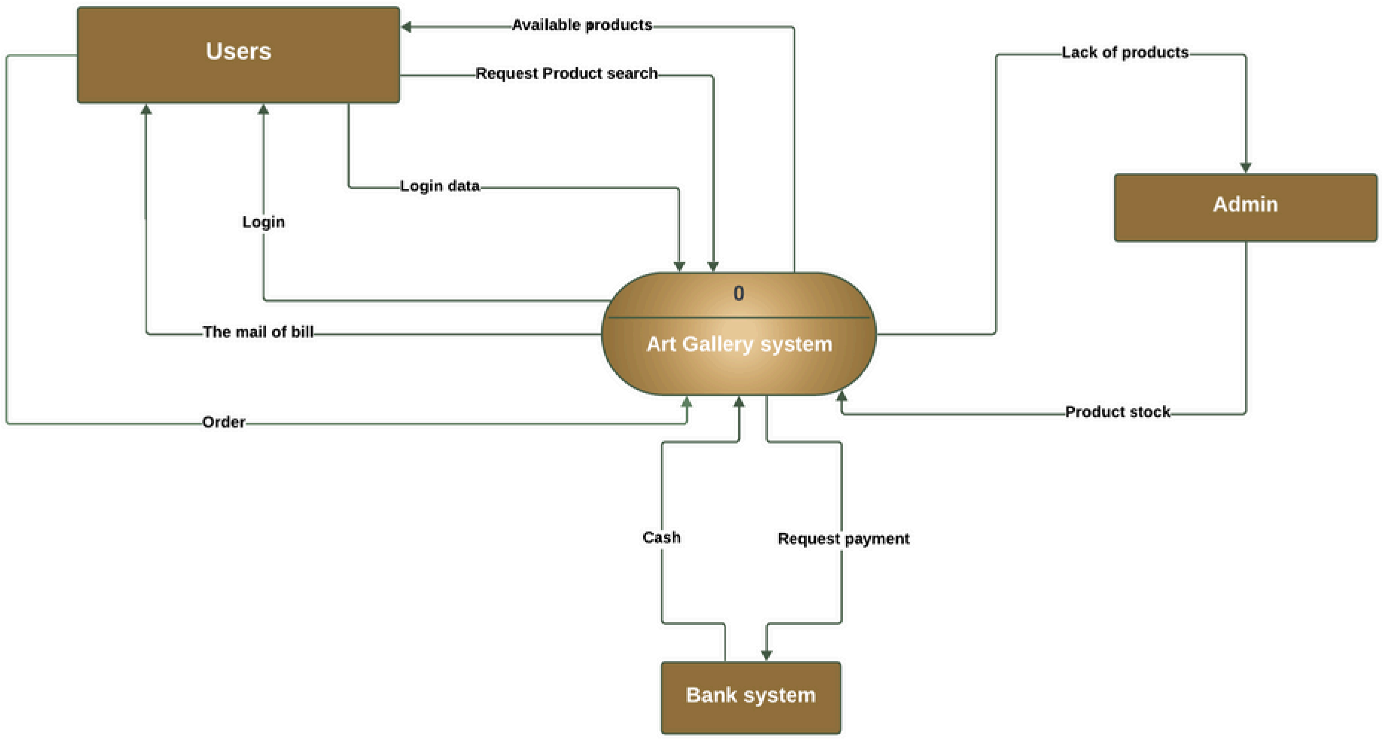
Authentication	• JWT for session tokens, or OAuth for third-party logins.
Payment Integration	• PayPal API.
Version Control	• GitHub/GitLab for source code management.

Design Phase

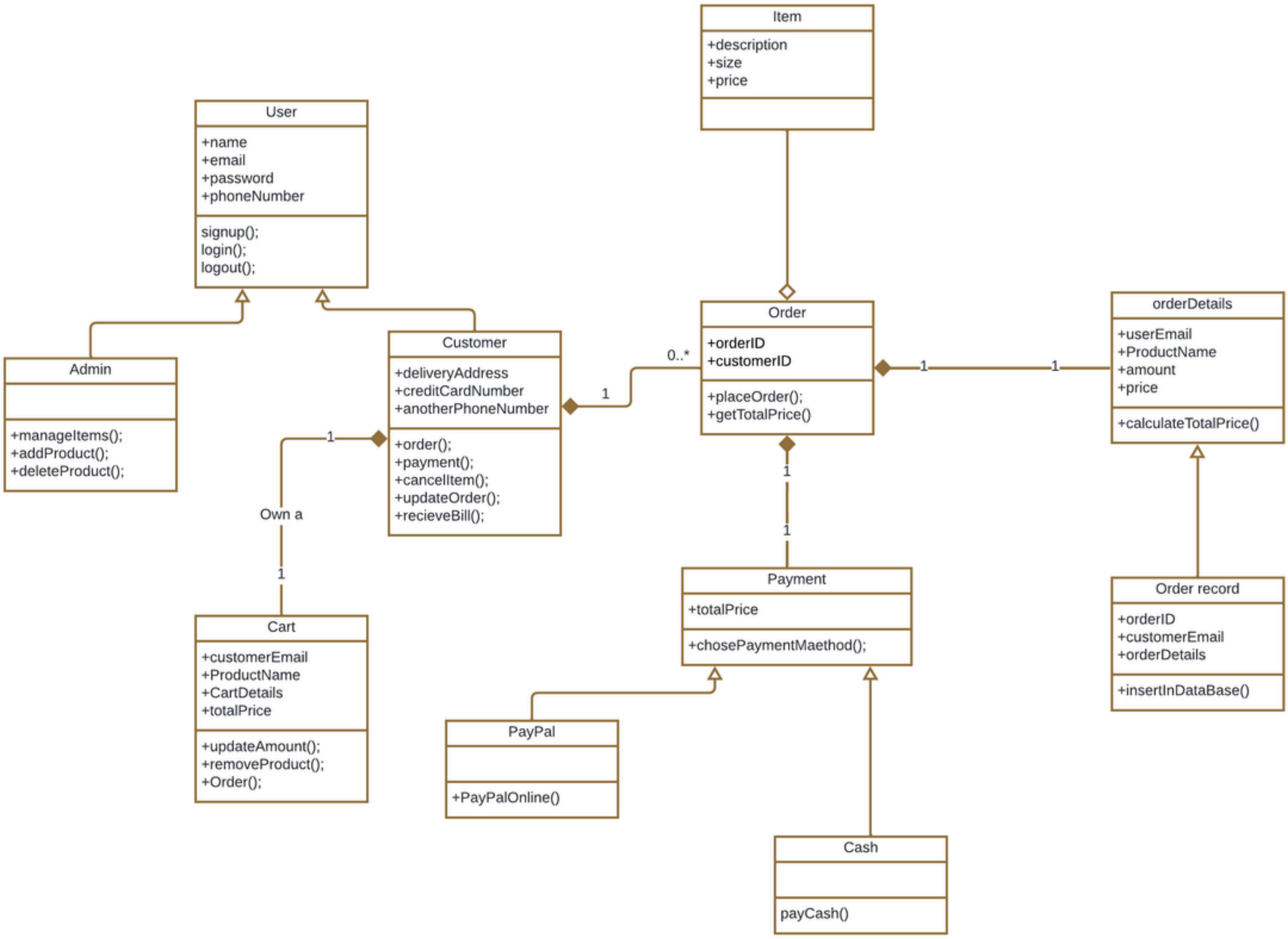
Use Case Diagram :



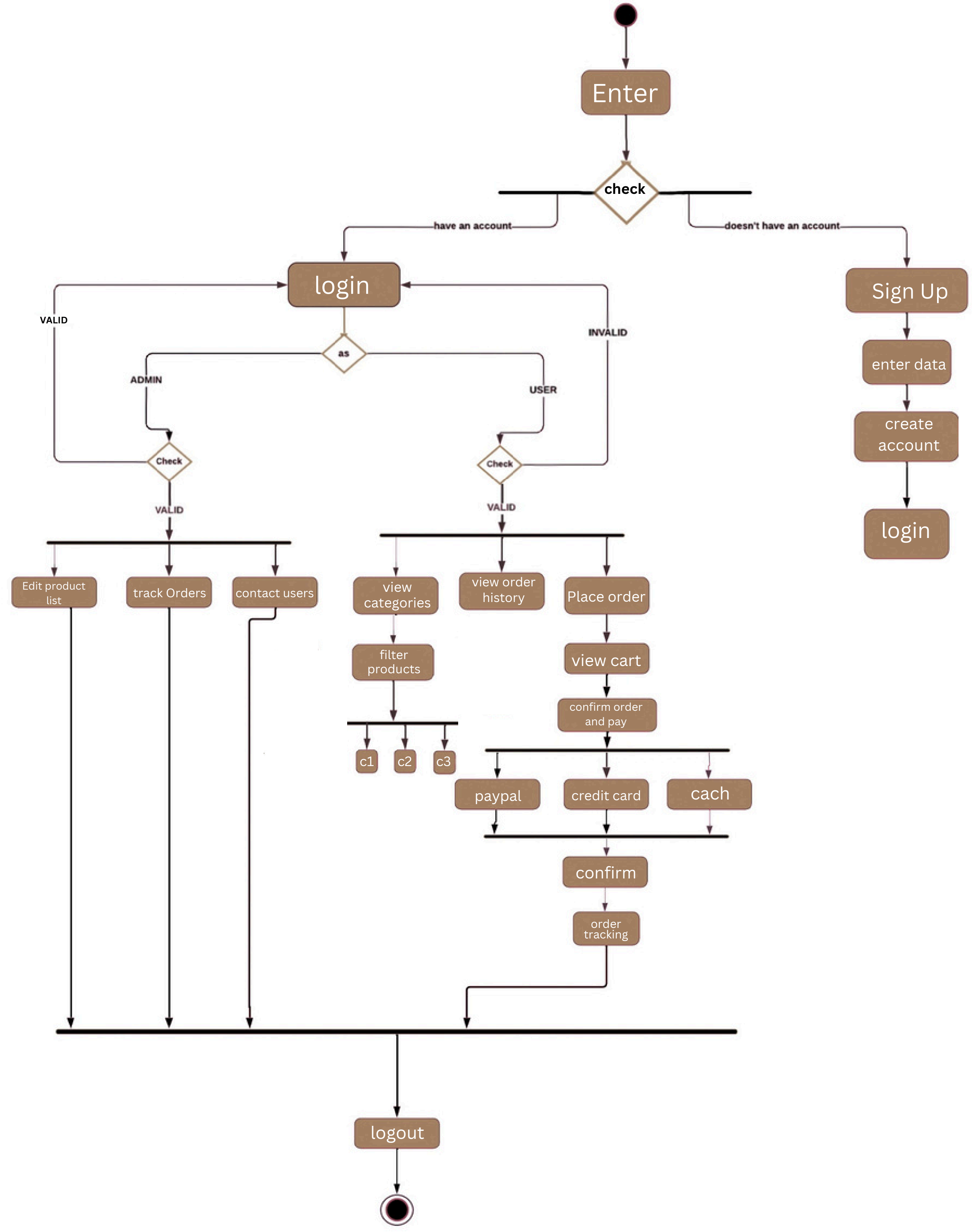
Context Diagram :



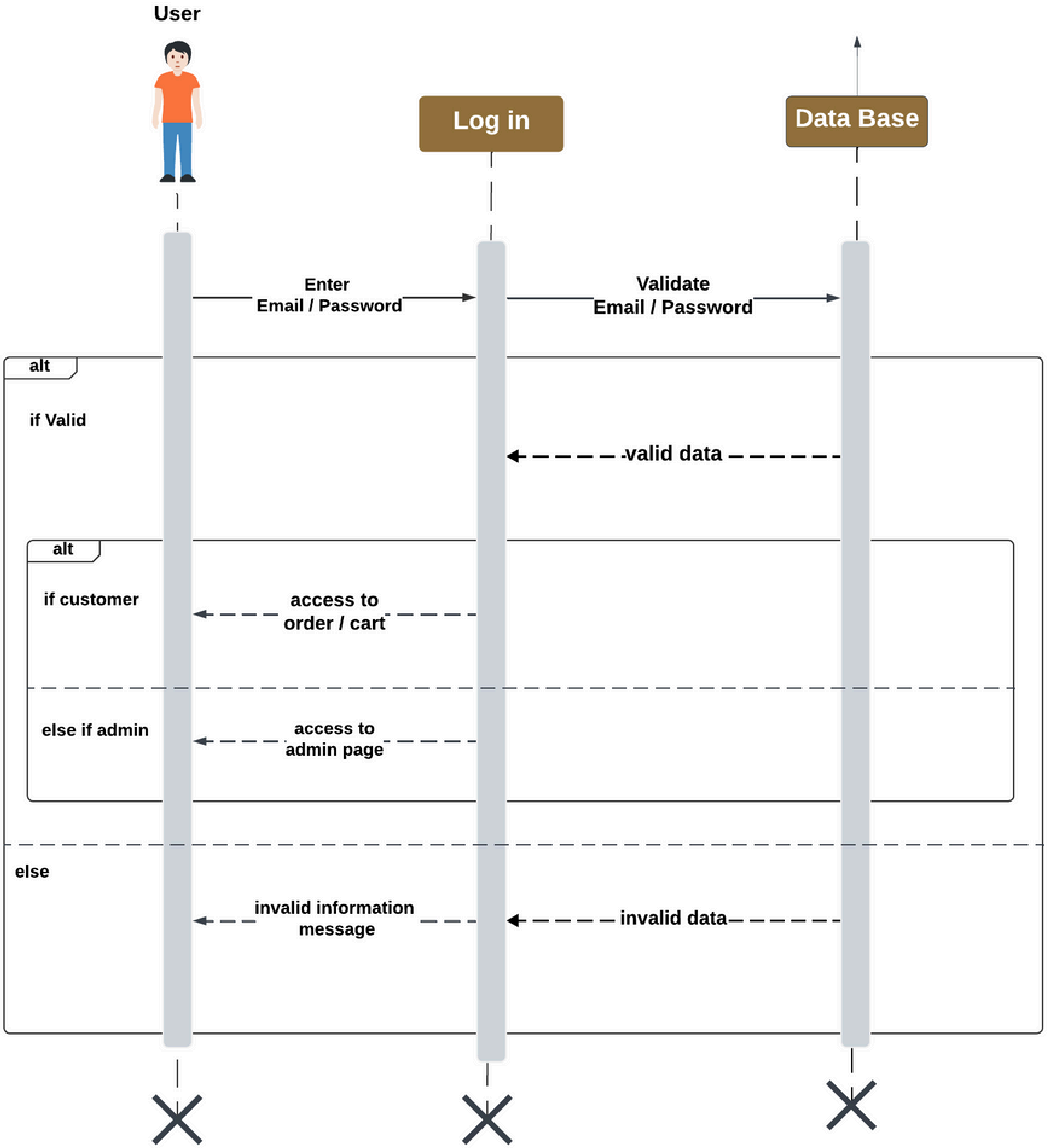
Class Diagram :

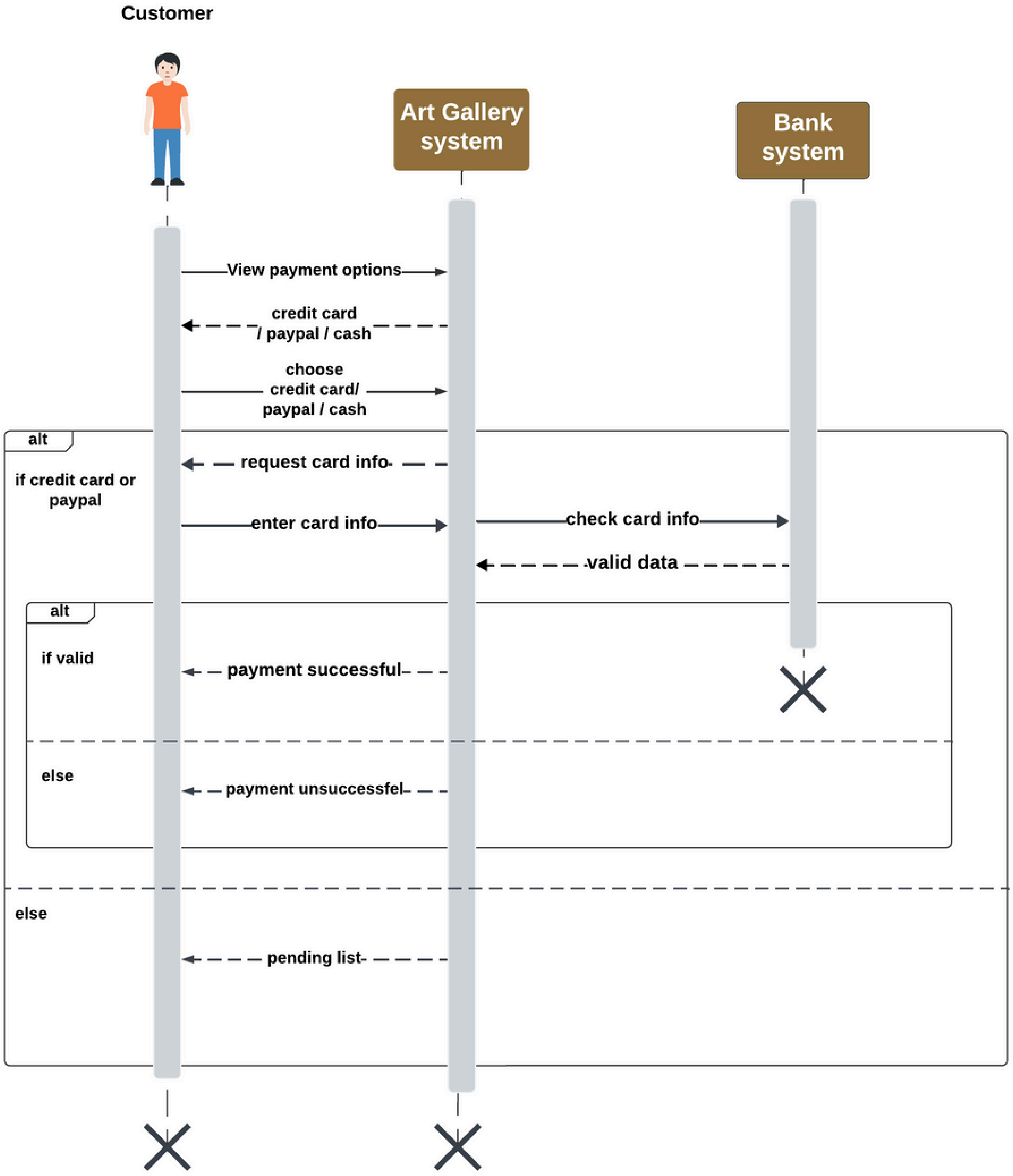


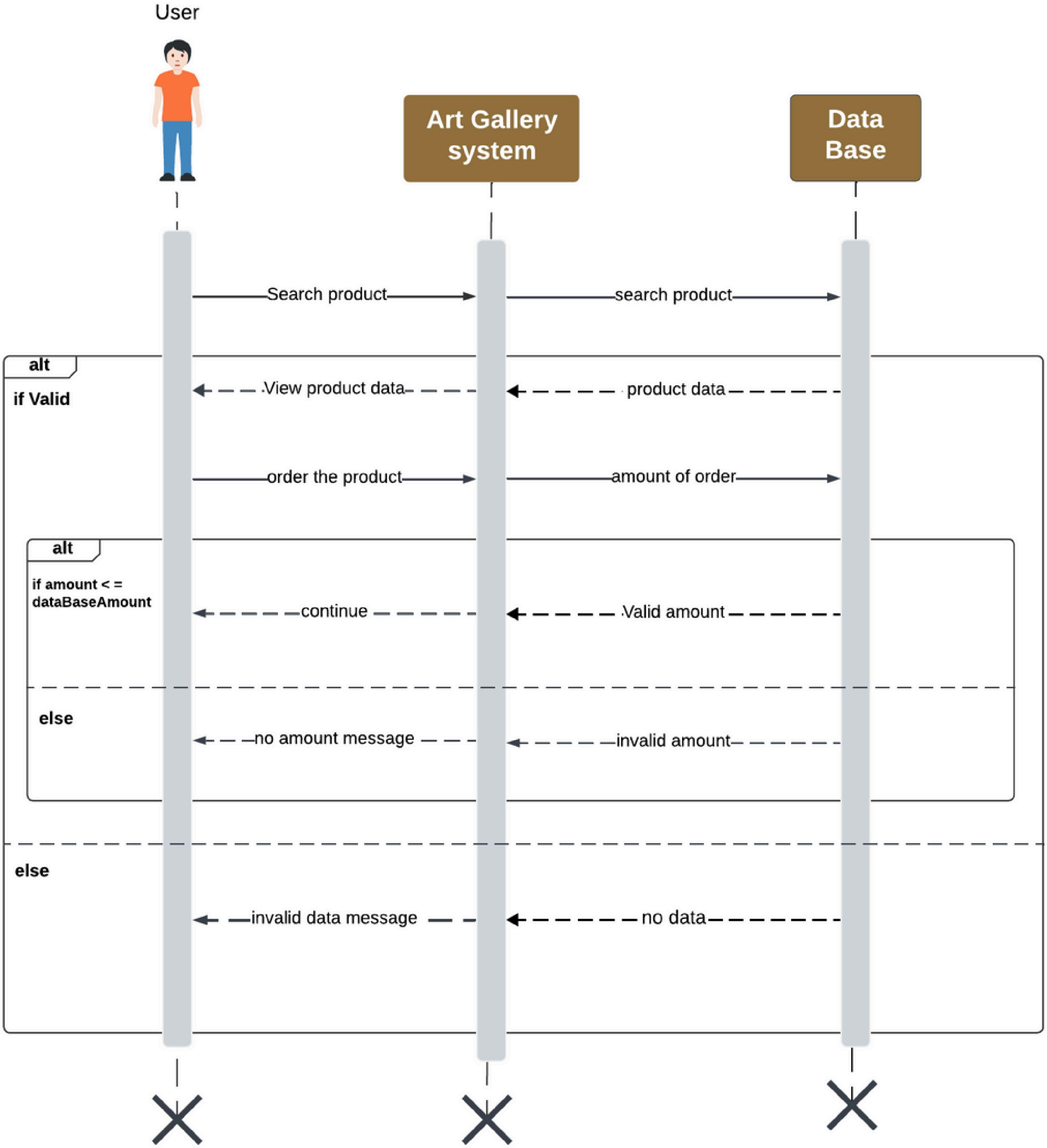
Activity Diagram :

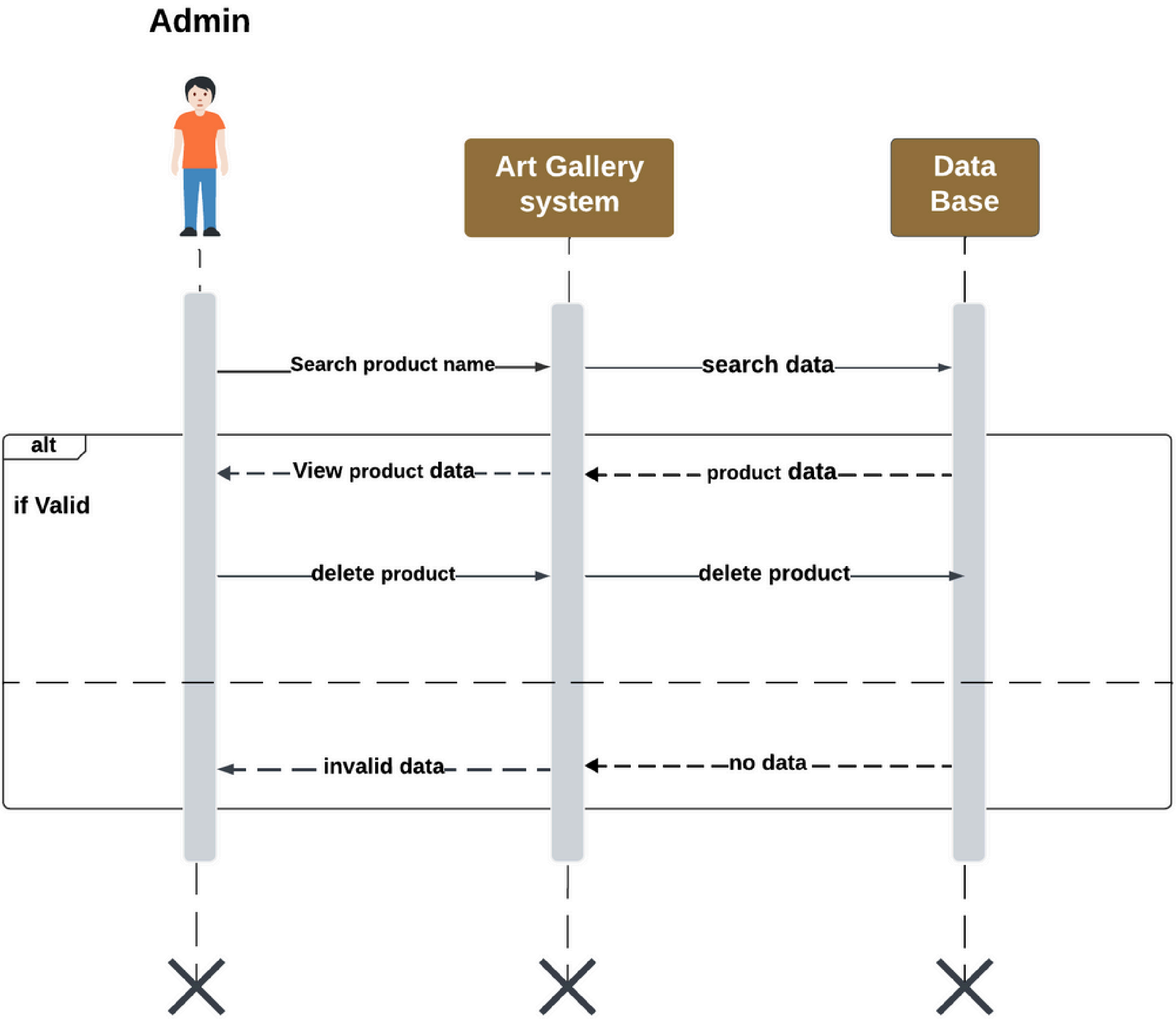


Sequence Diagrams :

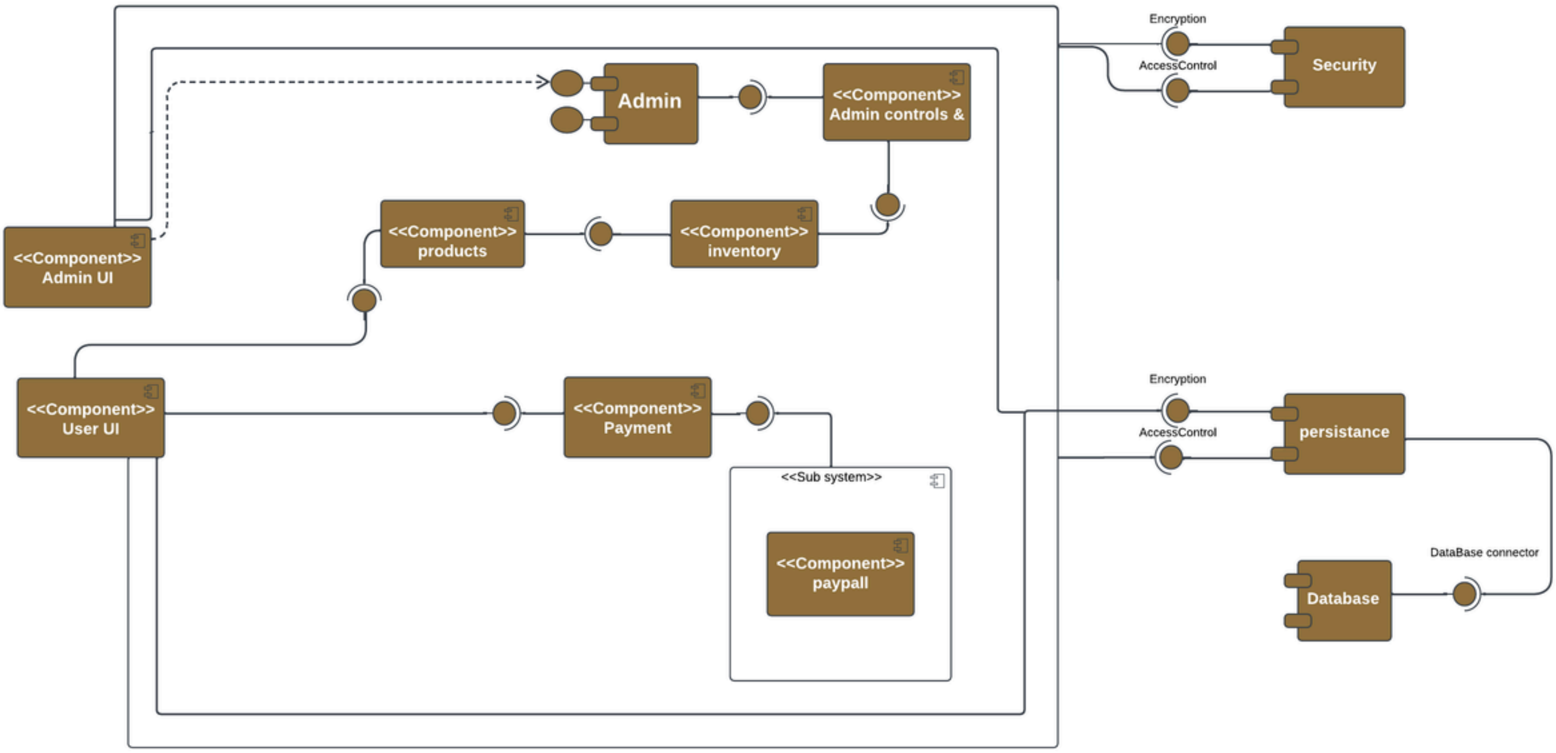



Customer
Art Gallery system
Bank system
View payment options
credit card / paypal / cash
choose credit card/ paypal / cash
alt
if credit card or paypal
request card info
enter card info
check card info
valid data
alt
if valid
payment successful
else
payment unsuccessfel
else
pending list


User
Art Gallery system
Data Base
Search product
search product
alt
if Valid
View product data
product data
order the product
amount of order
alt
if amount < = dataBaseAmount
continue
Valid amount
else
no amount message
invalid amount
else
invalid data message
no data


Admin
Art Gallery system
Data Base
Search product name
search data
alt
if Valid
View product data
product data
delete product
delete product
invalid data
no data

Component Diagram :


Admin
<<Component>>
Admin controls &
<<Component>>
Admin UI
<<Component>>
products
<<Component>>
inventory
<<Component>>
User UI
<<Component>>
Payment
<<Sub system>>
<<Component>>
paypall
Encryption
AccessControl
Security
Encryption
AccessControl
persistance
DataBase connector
Database