**SDD&SRS**

**Website Overview:**

Our e-commerce web application, built with Spring Boot, offers a seamless shopping experience for users. Customers can browse through a wide range of products, add items to their cart, and securely checkout. Admins have the ability to efficiently manage products, orders, and customer data, ensuring smooth operations on the platform.

With Spring Boot's robust framework, our e-commerce site is reliable, scalable, and easy to maintain. The platform prioritizes security, providing secure login mechanisms and leveraging JWT-based authentication and authorization systems. Detailed analytics are available for administrators to gain insights into order metrics, enhancing decision-making processes.

From product discovery to order management, our platform is designed to optimize the e-commerce experience for both customers and administrators. Whether it's effortless shopping or efficient product management, our platform delivers a user-friendly interface backed by powerful technology.

**Admin Module:**

**Secure Administrative Access:**

Administrators can securely log in using their credentials.

Generate tokens for accessing authenticated APIs.

Utilize a robust JWT-based authentication and authorization system for enhanced security.

**Category Management:**

Effortlessly post and retrieve product categories.

Maintain seamless administrative control over categories.

**Efficient Product Management:**

Administrators can efficiently perform CRUD operations (Create, Read, Update, Delete) on products.

Ensure a streamlined product management process for effective administration.

**Order Oversight:**

Monitor placed orders with ease.

Effortlessly manage order status updates such as shipped, delivered, or in progress.

**Advanced Analytics:**

Dive into detailed analytics through a dedicated analytics page.

Gain insights into placed, shipped, and in-progress orders.

View sales data for the current and previous months for informed decision-making.

**FAQs and Coupons:**

Admins can disseminate valuable product information through FAQs.

Manage discounts by extending them via a coupon management system.

**Customer Module:**

**Secure Customer Access:**

Customers can securely log in using their credentials.

Generate tokens for accessing authenticated APIs.

Leveraging a robust JWT-based authentication and authorization system for secure access.

**Product Discovery:**

Explore an extensive product selection conveniently displayed in the dashboard.

**Efficient Shopping:**

Seamlessly add single or multiple products to the cart for a smooth shopping experience.

Simplified Ordering:

Effortlessly place orders after adding products to the cart.

Apply discounts by entering coupon IDs during checkout.

**Share Your Feedback:**

Customers can post reviews for delivered products.

Facilitate informed decision-making for other users based on shared experiences.

**Wishlist:**

Create a personalized wishlist by adding favorite products.

**Product Insights:**

Obtain comprehensive details about specific products.

Access product details, reviews, and FAQs with a simple click on the product name.

**User Profile Management:**

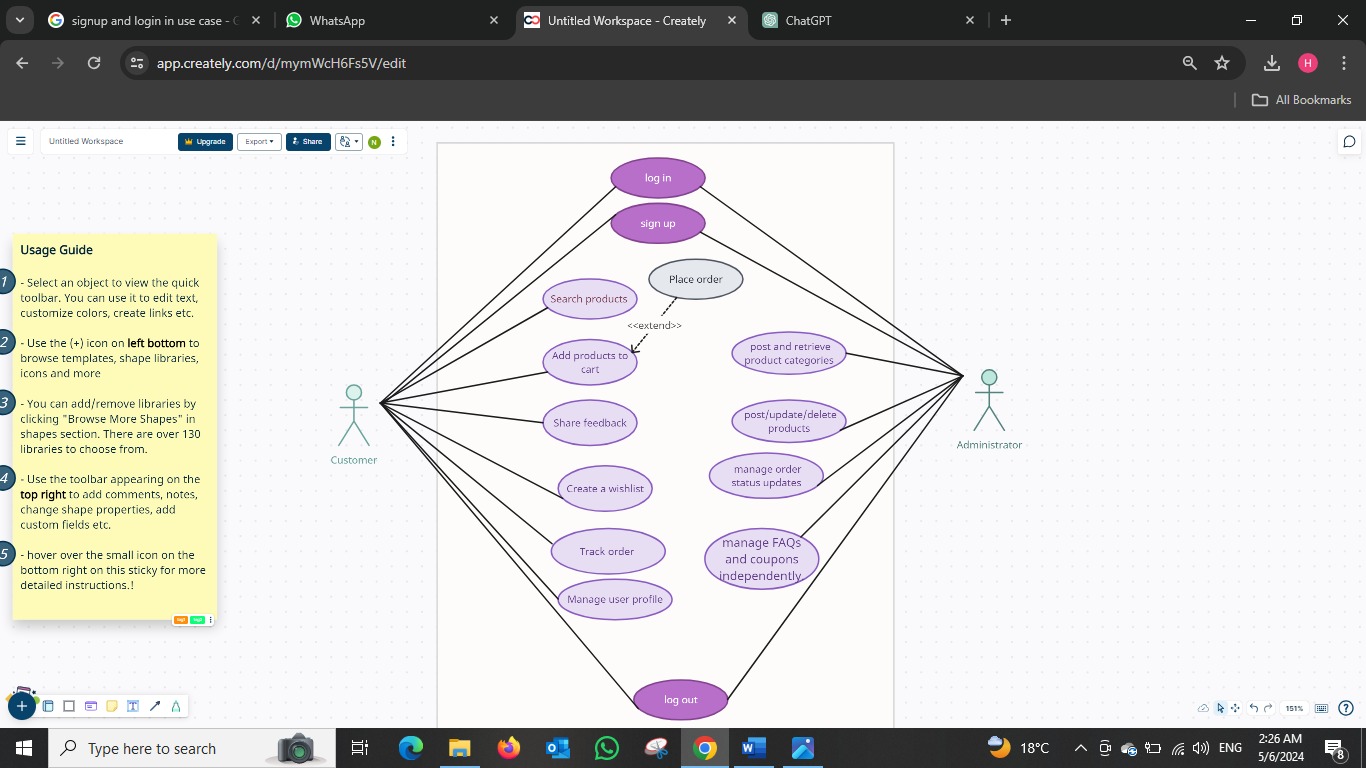
Easily update user profiles and change passwords for enhanced control over personal information.

**Order Tracking:**

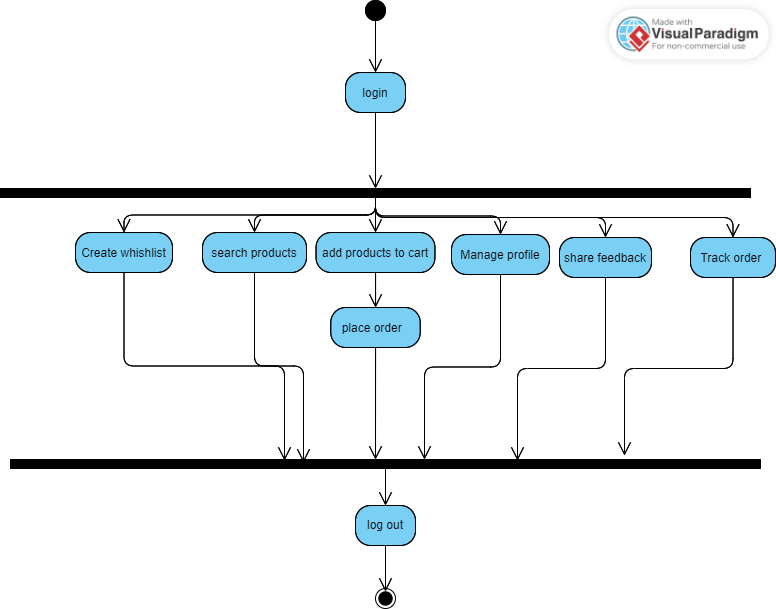
Conveniently track orders using tracking IDs. Access order tracking functionalities even without being logged in

**DIAGRAMS**

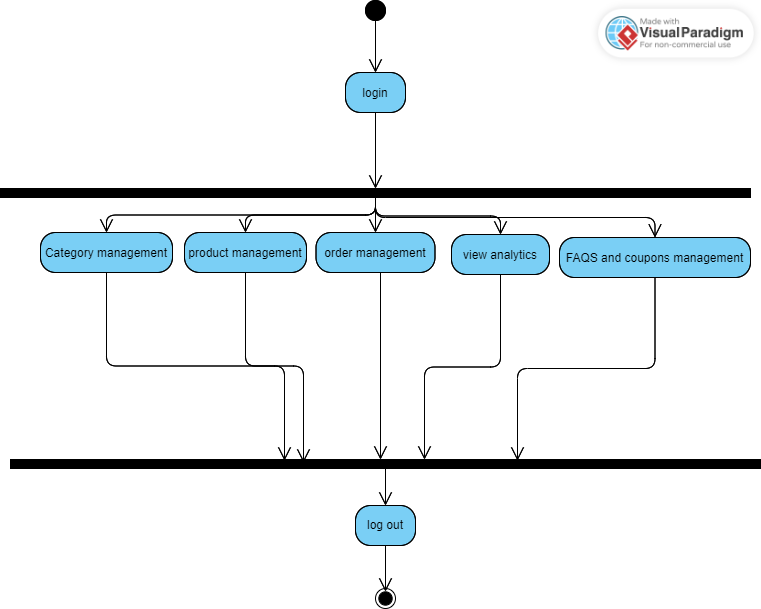
**Use case diagram:**

****

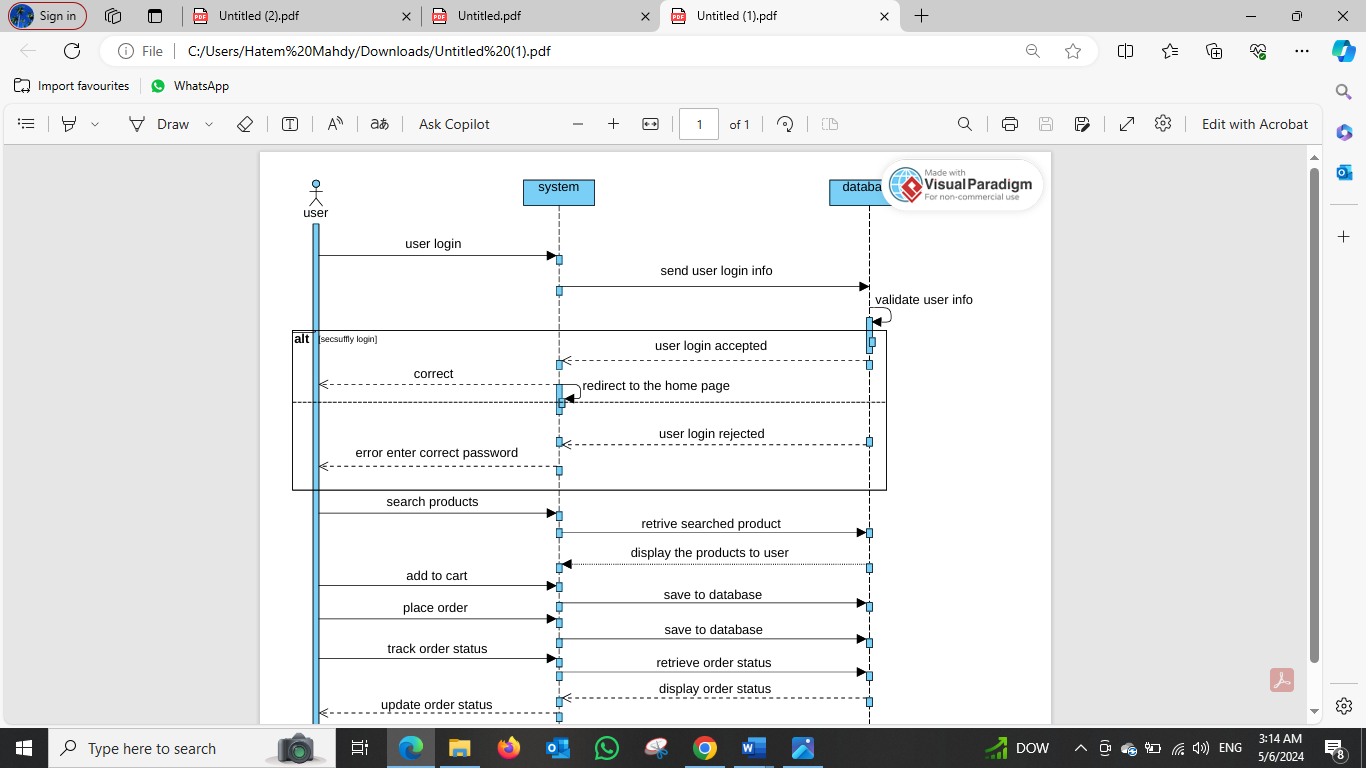
User activity diagram:



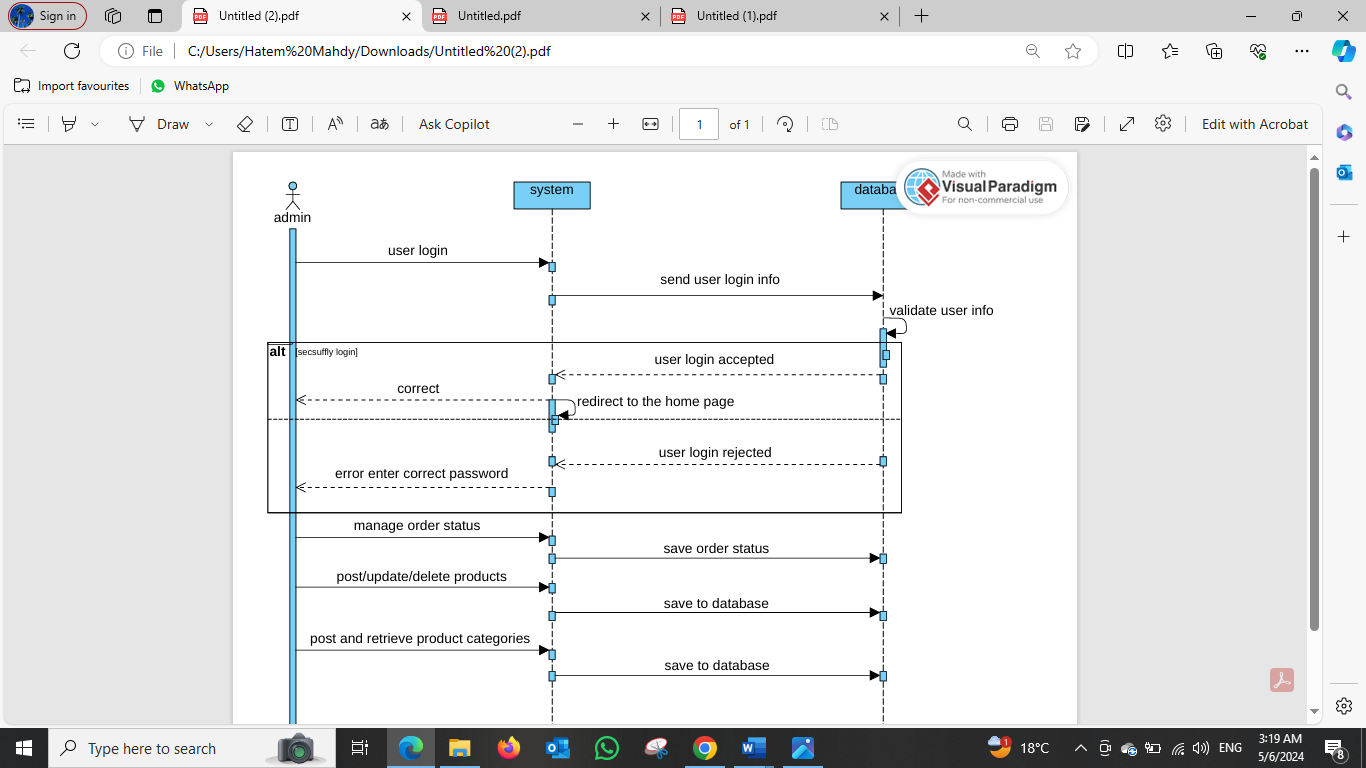
Admin activity diagram:



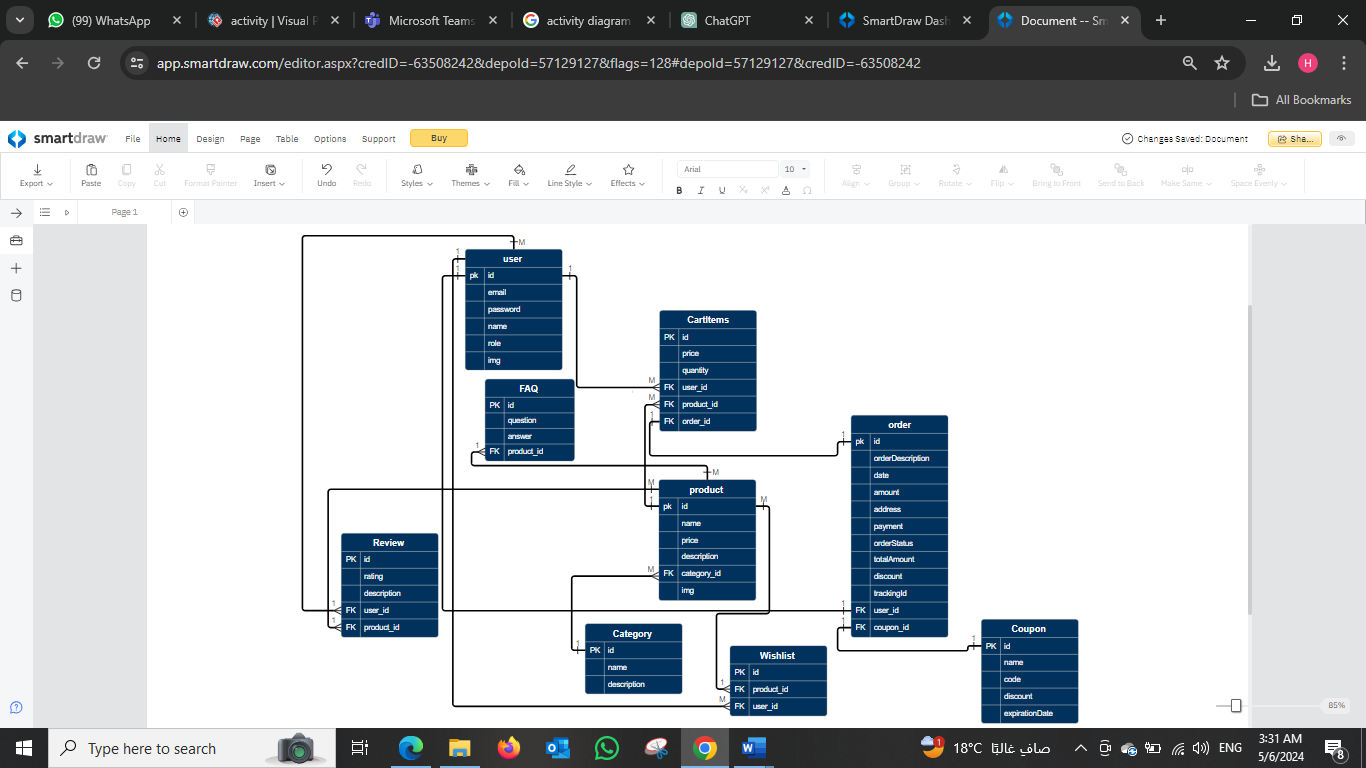
User sequence diagram:

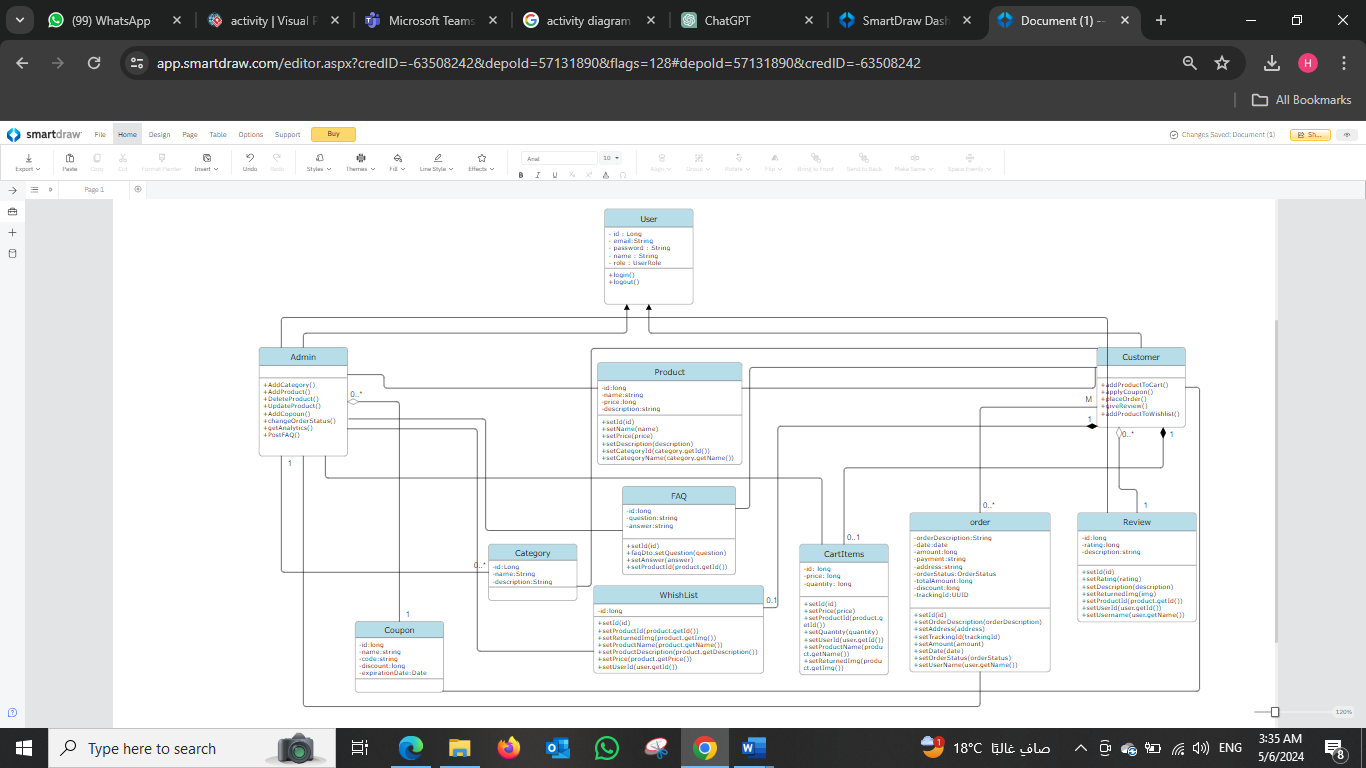


Admin sequence diagram:



ERD:



Class diagram :

OCL :

Constraint: Product price must be non-negative

context Product

inv: self.price >= 0

Constraint: Order must have at least one item

context Order

inv: self.items->notEmpty()

Constraint: Order total must be equal to the sum of item prices

context Order

inv: self.total = self.items->sum(price)

Constraint: Customer email must be unique

context Customer

inv: Customer.allInstances()->select(c | c <> self)->forAll(c | c.email <> self.email)

Constraint: User passwords must have at least 8 characters

context User

inv: self.password->notEmpty() and self.password.size() >= 8