Jaypee Institute of Information Technology, Noida

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AND INFORMATION TECHNOLOGY

Project Title: Ecommerce Website

Enroll. No.	Name of Student
SIMRAN	21103223
MOHIT GARG	21103229
AIBAD KHAN	21103218
CHIRANSHU AGRAWAL	21103220

Course Name: Software Engineering Lab
Course Code: 15B17CI573
Program: B. Tech. CS&E

3rd Year 6th Sem

2023 - 2024

Table of contents

Page No

1. Title 1
2. Acknowledgement 3
3. Customer Problem Statement: 4
   a) Problem statement
   b) Decomposition into sub problem
4. Goals, Requirements and Analysis 6
   a) Business Goals
   b) Enumerated Functional Requirements
5. Use cases 8
   a) Stakeholders
   b) Actors and Goals
   c) Use Case Diagram
6. Class diagram 12
7. Activity diagram 13
8. Sequence diagram 14
9. Implementation 15
10. Test case Report 17
11. Contribution of each team member 18
12. References 19

Acknowledgement:

We extend our heartfelt gratitude to **Mrs Sonal Saurah** and **Sumeshwar Sir** who have been the guiding lights throughout this project. Their unwavering support, invaluable insights, and dedication to our growth have been instrumental in shaping our journey. We are profoundly grateful for their patience, encouragement, and commitment to excellence, which have inspired us to reach new heights. This project stands as a testament to their profound impact on our lives, and we dedicate its success to them with deep respect and admiration.

Thank you for being the pillars of wisdom and inspiration in our educational voyage.

Customer Problem Statement:

Problem Statement:

As an avid online shopper, I often find myself facing frustrations when browsing through existing e-commerce websites. It's like entering a cluttered store where finding what I need feels like a treasure hunt, and I'm never quite sure if the products I'm considering are of good quality or if I can trust the sellers behind them.

Browsing these crowded websites can be confusing, with so many things to look at that I feel like I'm swimming in a big ocean of stuff.I long for a solution that simplifies the product search process. Scrolling through countless pages of listings is time-consuming and overwhelming. A software system that employs advanced search algorithms and personalized recommendations based on my preferences and past purchases would greatly enhance my shopping journey.

Moreover, I crave transparency and assurance regarding the quality of products and the credibility of sellers. Detailed product descriptions, authentic customer reviews, and verified seller credentials would go a long way in instilling confidence in the items I'm considering.

Additionally, the accessibility and usability of the platform across different devices are crucial. Whether I'm shopping on my laptop, tablet, or smartphone, I expect a seamless and consistent browsing experience without any technical glitches or usability issues.

Above all, trust is paramount in my online shopping experience. A software system that prioritizes transparency, reliability, and responsive customer support would reassure me that I'm making informed decisions and that assistance is readily available whenever needed.

What I truly desire is a streamlined and trustworthy online shopping experience, where finding the right products is effortless and I can shop with confidence, knowing that every item meets my standards of quality.

In short , the ideal software system would revolutionize my online shopping experience by addressing these pain points, offering convenience, confidence, and peace of mind every step of the way. With such a platform, shopping online would no longer be a chore but a delightful and rewarding experience

Decomposition into Sub Problem:

1) **Enhancing Product Presentation**:
    - Ensure clear and attractive product images and descriptions to help users make informed decisions.
    - Provide user-generated reviews and ratings to build trust and credibility

2) **Optimizing User Experience**:
    - Design a user-friendly interface that is easy to navigate and intuitive to use.
    - Ensure website responsiveness and compatibility across various devices and screen sizes.

3) **Streamlining Checkout Process**:
    - Simplify the checkout process with minimal steps and clear instructions.
    - Offer secure transactions.

4) **Ensuring Trustworthiness**:
    - Implement security measures to protect user data and ensure safe transactions.

5) **Transparent Product Information**:
    - Give customers all the details they need about products, like sizes, materials, and reviews from other buyers.

Goals requirements and analysis:

Business Goals:

1. **Increase Sales Revenue**
    - Enhance user experience to encourage more purchases.
    - Expand product offerings to attract a wider customer base.
2. **Improve Customer Satisfaction**
    - Provide reliable customer support services.
    - Ensure smooth and secure transactions.
3. **Fast Customer Service**
    - Address inquiries, concerns, and issues promptly to enhance customer satisfaction and loyalty.
4. **Huge discounts**
    - Bringing offers and discounts at regular periods to continue customer's interaction with the website.
    - Giving special coupons in festivals.
5. **Build Brand Loyalty**
    - Offer personalized recommendations based on user preferences.
    - Implement loyalty programs to reward repeat customers.

Enumerated Functional Requirements:

Requirement	Priority Weight	Description
REQ-1	high	User registration and login functionality.
REQ-2	high	Product browsing and search capability.
REQ-3	high	Shopping cart management for adding, removing, and editing items.

REQ-4	medium	Secure payment gateway integration.
REQ-5	high	Integration with customer feedback system for product reviews.
REQ-6	high	Real-time inventory updates to ensure product availability.
REQ-7	medium	Inventory management for products.
REQ-8	high	Integration with shipping services for order fulfillment.
REQ-9	medium	User account management (profile editing, password reset, etc.).

Enumerated Nonfunctional Requirements:

`10. Performance

- Priority: High
- Ensure fast loading times for web pages to prevent user frustration.

11. Security

- Priority: High
- Implement SSL encryption for secure transactions and protect user data.

12. Scalability

- Priority: Medium
- Design the website architecture to handle increasing traffic and product listings over time.

13. Reliability

- Priority: High
- Minimize downtime and ensure the website is accessible 24/7.

14. Usability

- Priority: High
- Design an intuitive user interface for easy navigation and seamless shopping experience.

15. Compatibility
    - Priority: Medium
    - Ensure compatibility with various web browsers and devices to reach a broader audience.

16. Accessibility
    - Priority: Medium
    - Ensure the website is accessible to users with disabilities, following WCAG guidelines.

17. Regulatory Compliance
    - Priority: High
    - Adhere to data protection regulations such as GDPR to protect user privacy.

Use Cases

Stakeholders:

1. **Customers:**
   Users who visit the website to browse products, make purchases, and interact with the system.

2. **Customer Service Representatives:**
   Employees responsible for addressing customer inquiries, issues, and providing support.

3. **Administrators:**
   Staff members who manage the website, handle inventory, process orders, and maintain the system.

4. **Shipping Partners:**
   Companies responsible for delivering orders to customers.

5. **Payment Gateway Providers:**
   Organizations that facilitate secure online payment transactions.

6. **Product Suppliers:**
   Businesses or individuals who supply products to be listed and sold on the platform

Actors and Goals:

1. **Customer (Initiating Actor)**
   - Goal: To browse products, add items to the cart, proceed to checkout, make payments, and track orders.

2. **Customer Service Representative (Participating Actor)**
   - Goal: To address customer inquiries, resolve issues with orders, and provide assistance throughout the purchase process.

3. **Administrator (Participating Actor)**
   - Goal: To manage product listings, update inventory, process orders, monitor website performance, and handle administrative tasks.

4. **Shipping Partner (Participating Actor)**
   - Goal: To receive order details, pick up packages from the warehouse, and deliver them to customers within the specified time frame.

5. **Payment Gateway Provider (Participating Actor)**
   - Goal: To securely process payments for orders placed on the website and facilitate transactions between customers and merchants.

6. **Product Supplier (Participating Actor)**
   - Goal: To provide product information, update inventory levels, and fulfill orders in a timely manner.

Derived Use Cases:

1. **Browse Products**: Customer searches and views products available on the website.

2. **Add to Cart**: Customer adds selected items to the shopping cart for purchase.

3. **Checkout:** Customer proceeds to checkout, enters shipping and payment details, and con

4. **Customer Support:** Customer service representative assists customers with inquiries, order issues, and general support.

5. **Manage Products:** Administrator manages product listings, updates inventory levels, and adds new items to the website.

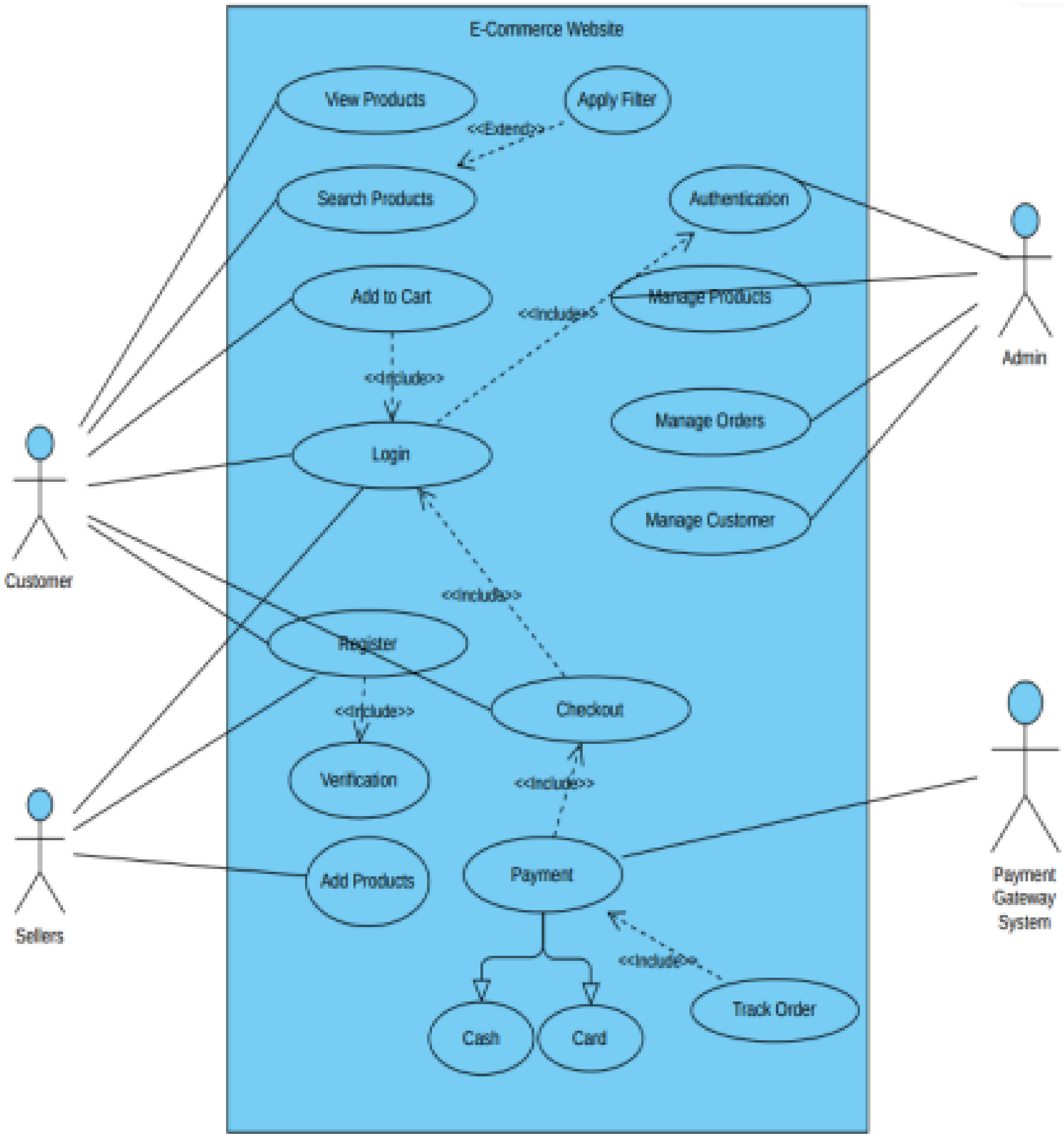
6. **Process Orders:** Administrator processes incoming orders, verifies payments, and prepares packages for shipping.

7. **Shipping and Delivery:** Shipping partner receives order details, picks up packages, and delivers them to customers.

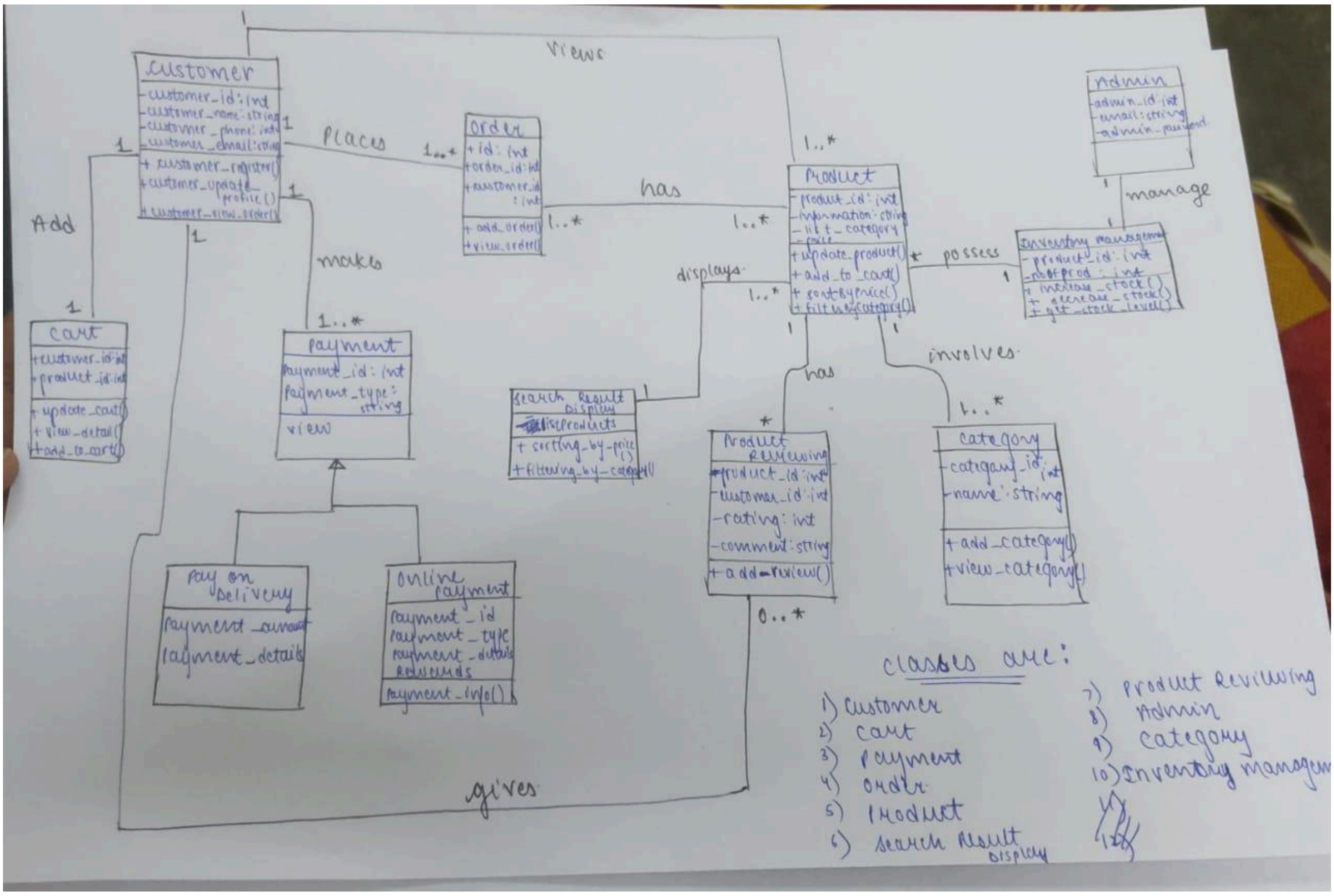
8. **Payment Processing:** Payment gateway provider securely processes customer payments for orders placed on the website.

9. **Inventory Management:** Product supplier updates inventory levels, provides stock updates, and ensures timely fulfillment of orders.

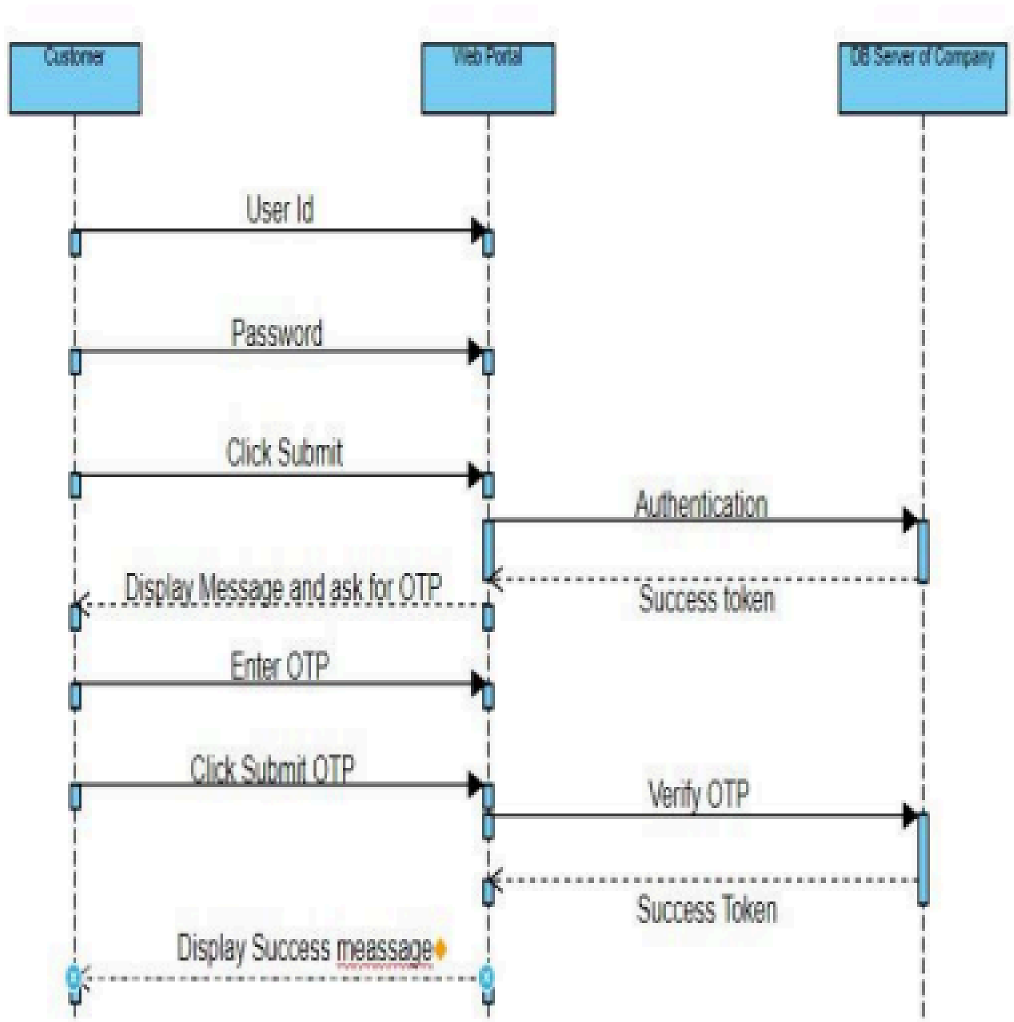
Use Case Diagram



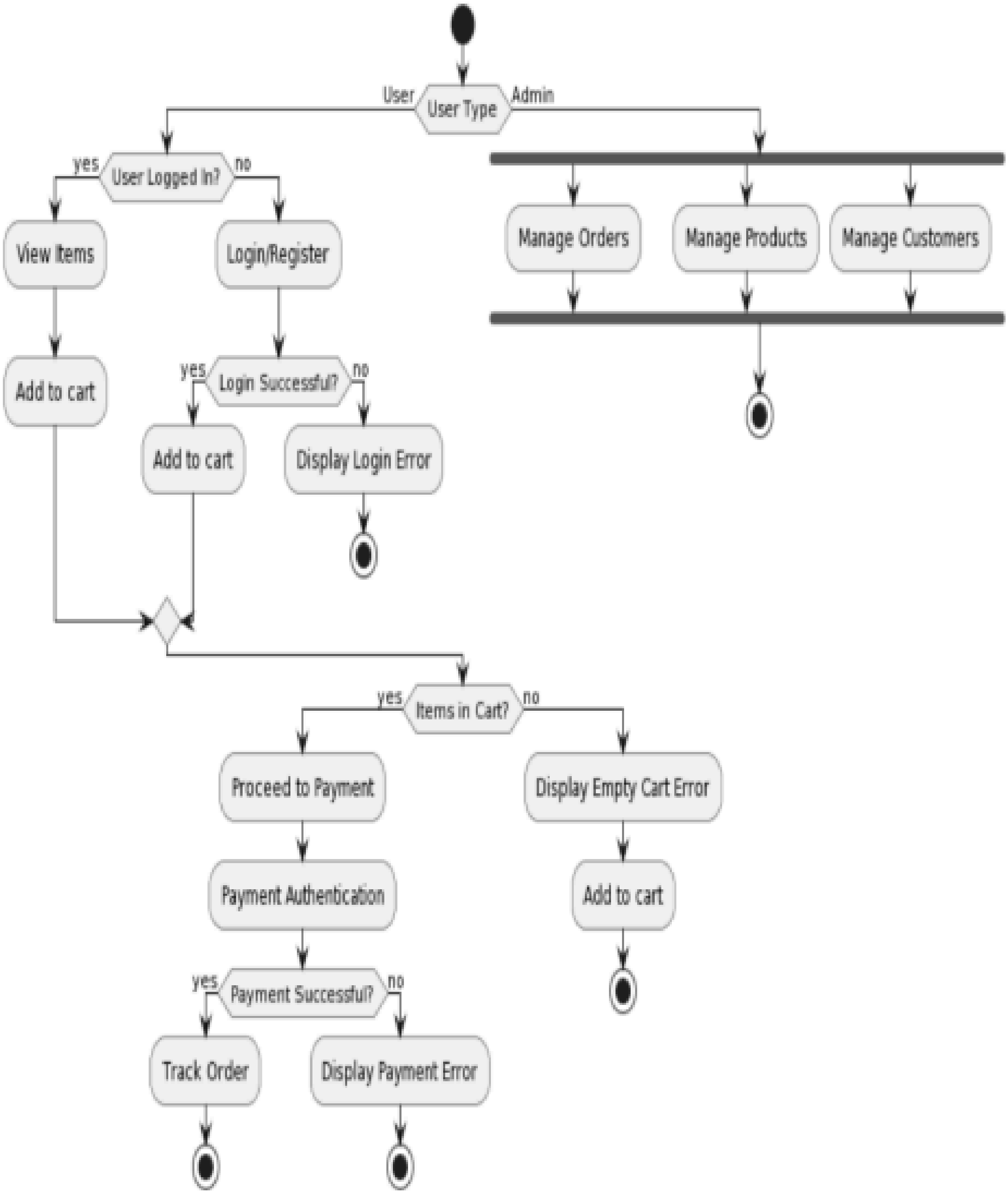
Class diagram



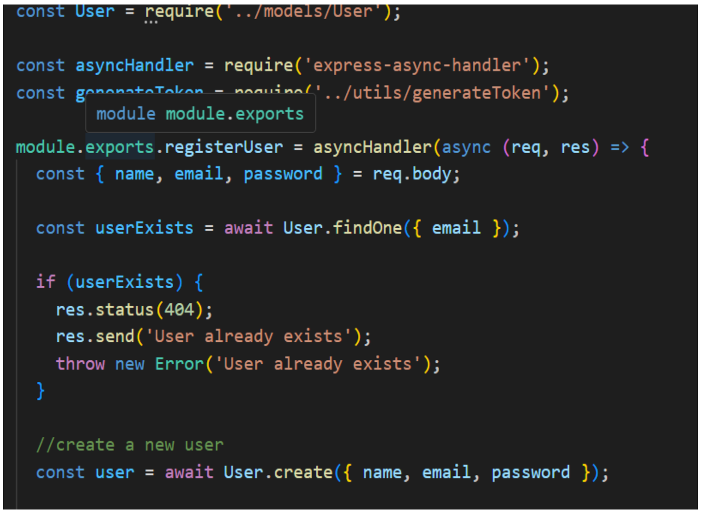
Activity diagram



Sequence diagram

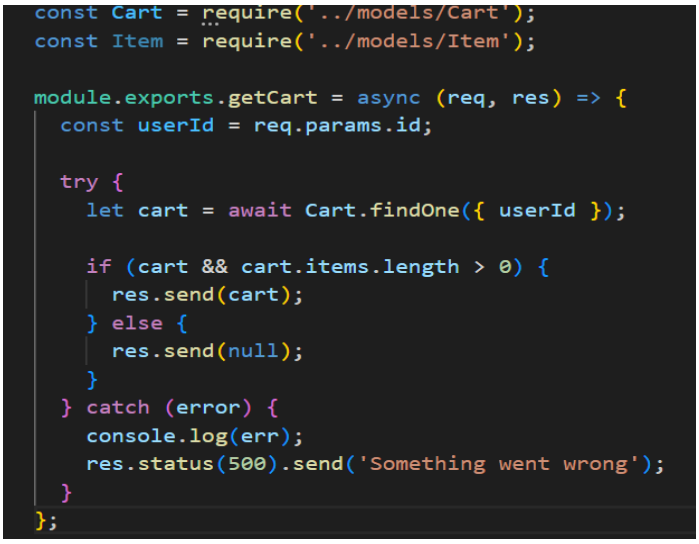


Implementation:



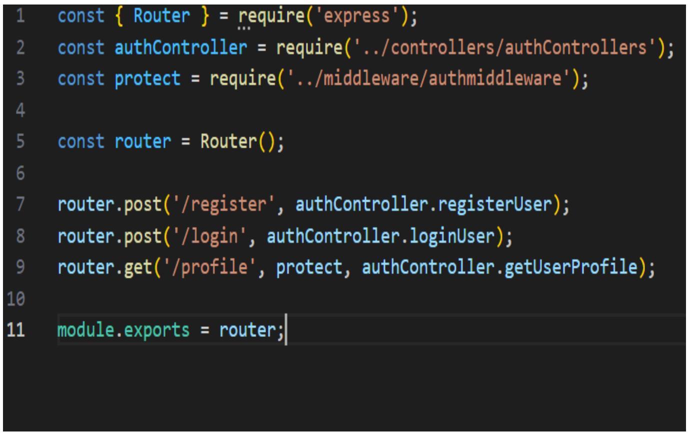
```
const User = require('../models/User');

const asyncHandler = require('express-async-handler');
const generateToken = require('../utils/generateToken');

module.exports.registerUser = asyncHandler(async (req, res) => {
  const { name, email, password } = req.body;

  const userExists = await User.findOne({ email });

  if (userExists) {
    res.status(404);
    res.send('User already exists');
    throw new Error('User already exists');
  }

  //create a new user
  const user = await User.create({ name, email, password });
```

This code is creating new user by using name,email and password from the request body,if the user already exists then it throws the message.



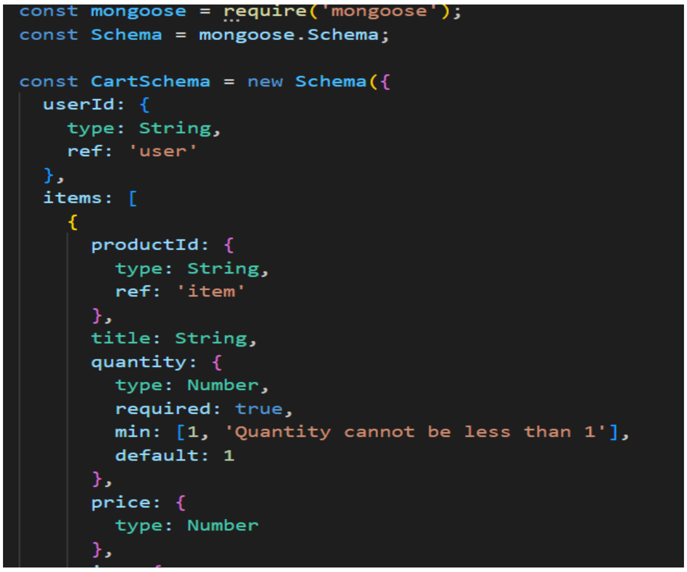
```
const Cart = require('../models/Cart');
const Item = require('../models/Item');

module.exports.getCart = async (req, res) => {
  const userId = req.params.id;

  try {
    let cart = await Cart.findOne({ userId });

    if (cart && cart.items.length > 0) {
      res.send(cart);
    } else {
      res.send(null);
    }
  } catch (error) {
    console.log(err);
    res.status(500).send('Something went wrong');
  }
};
```

This is finding item in card according to user id if it exists then it passes the cart details further



```
const { Router } = require('express');
const authController = require('../controllers/authControllers');
const protect = require('../middleware/authmiddleware');

const router = Router();

router.post('/register', authController.registerUser);
router.post('/login', authController.loginUser);
router.get('/profile', protect, authController.getUserProfile);

module.exports = router;
```

This code is creating routes that will route the user to register,login,profile based on relative URL.



```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const CartSchema = new Schema({
  userId: {
    type: String,
    ref: 'user'
  },
  items: [
    {
      productId: {
        type: String,
        ref: 'item'
      },
      title: String,
      quantity: {
        type: Number,
        required: true,
        min: [1, 'Quantity cannot be less than 1'],
        default: 1
      },
      price: {
        type: Number
      },
```

This code is creating schema for cart items,it is showing which things are stored in the database for cart. Eg. each item in cart should have user id, product id,quantity,price

Test Case Report

Test Case 1: Successful Registration

Test ID: TC-REG-001

Description: Verify that a user can successfully register with valid details.

Preconditions:

The user accesses the registration page.

Test Steps:

Enter valid name, email, password, and other required details.

Click on the "Register" button.

Expected Result:

Confirmation message: “Registration successful”.

Actual Result:

Confirmation message: “Registration successful”.

Status: Passed

Test Case 2: Invalid Email Format

Test ID: TC-REG-002

Description: Verify that an error message is displayed when an invalid email format is provided during registration.

Preconditions:

The user accesses the registration page.

Test Steps:

Enter an invalid email format (e.g., "test@example").

Click on the "Register" button.

Expected Result:

Error message: "Please enter a valid email address."

Actual Result:

Error message: "Please enter a valid email address."

Status: Passed

Test Case 3: Empty Password Field

Test ID: TC-REG-003

Description: Verify that an error message is displayed when the password field is left empty during registration.

Preconditions:

The user accesses the registration page.

Test Steps:

Enter valid name and email.

Leave the password field empty.

Click on the "Register" button.

Expected Result:

Error message: "Please enter a password."

Actual Result:

Error message: "Please enter a password."

Status: Passed

Test Case 4: Password Strength Validation

Test ID: TC-REG-004

Description: Verify that an error message is displayed when the password does not meet the minimum strength requirements during registration.

Preconditions:

The user accesses the registration page.

Test Steps:

Enter valid name and email.

Enter a weak password (e.g., "password").

Click on the "Register" button.

Expected Result:

Error message: "Password must be at least 8 characters long and contain at least one uppercase letter, one lowercase letter, one number, and one special character."

Actual Result:

Error message: "Password must be at least 8 characters long and contain at least one uppercase letter, one lowercase letter, one number, and one special character."

Status: Passed

References:

[1].https://www.geeksforgeeks.org/software-testing-basics

[2].https://en.wikipedia.org/wiki/Software_metric

[3].https://www.javatpoint.com/functional-vs-non-functional-requirements

[4].https://www.datensen.com/data-modeling/moon-modeler-for-databases.html