Software Requirements Specification

for  
**<MartCom>**

Prepared by <Muhammad Ahsan, Muhammad Shahaf, Ammar Bilal>  
< Date Created: November 30, 2024 >

Table of Contents

Table of Contents

[1. Introduction 1](#_Toc183910854)

[1.1 Purpose 1](#_Toc183910855)

[1.2 Document Conventions 1](#_Toc183910856)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc183910857)

[1.3 Product Scope 2](#_Toc183910858)

[1.4 References 2](#_Toc183910859)

[2. Overall description 2](#_Toc183910860)

[2.1 Product Perspective 2](#_Toc183910861)

[2.2 Product Functions 2](#_Toc183910862)

[2.3 User Classes and Characteristics 3](#_Toc183910863)

[2.4 Operating Environment 3](#_Toc183910864)

[2.5 User Documentation 4](#_Toc183910865)

[2.6 Design and Implementation Constraints 4](#_Toc183910866)

[2.7 Assumptions and dependencies 5](#_Toc183910867)

[2.8 Process Model 5](#_Toc183910868)

[2.8.1 Customer Process Flow 5](#_Toc183910869)

[2.8.2 Admin Process Flow 6](#_Toc183910870)

[2.9 Project Plan 6](#_Toc183910871)

[2.9.1 Project Plan 6](#_Toc183910872)

[3. External Interface Requirements 8](#_Toc183910873)

[3.1 User Interface 8](#_Toc183910874)

[3.2 Software Interfaces 9](#_Toc183910875)

[4. System Features 10](#_Toc183910876)

[4.1 Admin Features 10](#_Toc183910877)

[4.2 User Features 10](#_Toc183910878)

[5. Functional Requirements 10](#_Toc183910879)

[5.1 Admin Features 10](#_Toc183910880)

[5.2 User Features 11](#_Toc183910881)

[6. Non-Functional Requirements 11](#_Toc183910882)

[7.Conclusion 11](#_Toc183910883)

**Appendix B: Analysis Models**

# 1. Introduction

## 1.1 Purpose

This document describes the software requirements for the <MartCom>. It outlines the features and functionalities of the system for both admin and user roles. The system aims to provide an efficient way for managing products, orders, customers, and more.

## 1.2 Document Conventions

This document uses the following conventions:   
- Features are categorized into Admin and User features.   
- Lists are provided with bullet points or numbered lists where appropriate.   
- Section headers indicate different components of the software system.

## 1.3 Intended Audience and Reading Suggestions

- Users: Registered users, guest users, and administrators who interact with the application.

- Stakeholders: Business owners or project managers who need an overview of the features and functionalities.

## 1.3 Product Scope

- User-friendly interface for browsing and purchasing products.

- Product catalog management for small business owners.

- Basic order tracking for customers.

## 1.4 References

- Stack overflow.

- MDN Docs.

- GitHub.

# 2. Overall description

This section provides a broad overview of the “MartCom”, highlighting its purpose, key features, and general context.

## 2.1 Product Perspective

The e-commerce web application is designed as an end-to-end solution for online shopping, integrating functionalities for both customers and administrators. It serves as a standalone platform that includes:

* A user-friendly interface for customers to browse, search, and purchase products.
* Administrative tools for managing product listings, inventory, and orders.
* A responsive design, making it accessible across desktop and mobile devices.
* Integration with third-party payment gateways for seamless transactions.

## 2.2 Product Functions

The application includes the following core functionalities:

* **Customer Module:**
  + User registration and login.
  + Product browsing, search, and filtering.
  + Shopping cart and checkout process.
  + Order tracking and management.
  + Order history module.
* **Admin Module:**
  + Product management (add, edit, delete).
  + Inventory tracking and stock updates.
  + View and manage customer orders.
  + Analytics dashboard for sales and performance tracking.
* **Additional Features:**
  + Secure payment processing.
  + Notifications for order status updates.

## 2.3 User Classes and Characteristics

The primary users of the application are divided into the following categories:

1. **Customers:**
   * Casual and frequent shoppers.
   * Require an intuitive interface for browsing and purchasing.
   * Expect secure payment options and reliable order tracking.
2. **Administrators:**
   * Responsible for managing the product catalog, inventory, and orders.
   * Need an efficient dashboard for monitoring sales and system health.

## 2.4 Operating Environment

The application operates in the following environments:

* **Frontend:**
  + Developed using React.js for web application.
* **Backend:**
  + Built with Express.js and Node.js, connecting to a PostgreSQL database.
* **Minimum System Requirements:**
  + A device with an internet connection.
  + Browser with JavaScript enabled.

## 2.5 User Documentation

The application is accompanied by the following documentation:

* **User Guide:**
  + Detailed steps for customer account creation, browsing, and order placement.
  + FAQs addressing common user concerns.
* **Admin Manual:**
  + Instructions for managing product listings, monitoring orders, and generating reports.
* **API Documentation:**
  + Details for developers integrating third-party services or extending the platform.

## Design and Implementation Constraints

**1- Technology Constraints**

* The application must use React.js for the frontend and Node.js with Express.js for the backend.
* PostgreSQL is the chosen database; no other database technologies will be used.
* Third-party libraries like Axios and Redux must be compatible with the selected tech stack.

**2- Localization Constraints**

* The application must support English as the default language, with optional support for additional languages in the future.

## 2.7 Assumptions and dependencies

* **Assumptions:**
  + Customers have access to the internet and modern web browsers.
  + Admins are familiar with basic e-commerce operations.
* **Dependencies:**
  + Uses third-party libraries and frameworks, which need regular updates.

## Process Model

The **Process Model** defines the flow of data and control through the system and outlines the key processes involved in the MartCom e-commerce platform. The process model is divided into two primary categories: **Customer-related processes** and **Admin-related processes**.

## 2.8.1 Customer Process Flow

- **User Registration and Login**:

* The user provides their credentials, and the system validates them.
* On successful login, the user gains access to their personalized dashboard.
* **Product Browsing and Selection**:
* Users can filter products by categories or search by name.
* Product details (price, description, and reviews) are displayed.
* **Add to Cart**:
* Users can add products to their shopping cart.
* Cart displays the total number of items and cost.
* **Checkout and Payment**:
* The user reviews the cart, enters shipping information, and selects a payment method.
* **Order Confirmation and Tracking**:
* After successful payment, users receive an order confirmation.
* Users can track the status of their orders in real time.

## 2.8.2 Admin Process Flow

- **Product Management**:

* Admin adds, edits, or removes products from the inventory.
* The system updates the product catalog, reflecting changes in real-time.
* **Order Management**:
* Admin reviews incoming customer orders and processes them.
* Order status updates (e.g., shipped, delivered) are monitored.
* **Customer Management**:
* Admin can view and manage customer profiles and order histories.
* **Reporting**:
* Admin views and generates sales reports (e.g., weekly sales, popular products) from the dashboard.

## Project Plan

The **Project Plan** provides an outline for the stages, tasks, and timeline for developing the MartCom e-commerce platform.

## 2.9.1 Project Plan

- **Phase 1: Planning and Design (Duration: 4 Weeks)**

* **Task 1**: Requirement gathering and analysis.
* **Task 2**: Creating initial wireframes and system architecture.
* **Milestone 1**: SRS document and initial wireframes completed.
* **Phase 2: Frontend and Backend Development** (Duration: 8 Weeks)
* **Task 1**: Develop customer-facing features (product browsing, cart, checkout, order tracking).
* **Task 2**: Develop admin-facing features (product management, order management, reporting).
* **Milestone 2**: Basic functional system prototype completed (customer and admin dashboards, product catalog).
* **Phase 3: Integration and Testing** (Duration: 2 Weeks)
* **Task 1**: Integrate third-party services (payment gateways, email notifications, Cloudinary).
* **Task 2**: Perform unit testing, integration testing
* **Milestone 3**: Fully integrated system with successful tests.
* **Phase 4: Post-Launch Support** (Ongoing)
* **Task 1**: Monitor system performance and address user feedback.
* **Task 2**: Regular updates and feature enhancements.
  1. **Feasibility report**
     1. **Technical Feasibility**
* **Technology Stack:**
* Frontend: React.js (modern, responsive UI/UX)
* Backend: Express.js (robust API development)
* Database: PostgreSQL (reliable relational database)
* Payment Gateway: Stripe (secure and easy to integrate)
* Hosting: AWS or Vercel (scalable cloud hosting)
* **System Requirements:**
* Development Tools: VS Code, Postman, Git
* Deployment: Linux-based VPS or cloud infrastructure
* User Devices: Compatible with browsers on desktops, tablets, and smartphones
  + 1. **Operational Feasibility**
* **Team Composition:**
* **Development Team:** 3 developers (frontend, backend, full-stack).
* **UI/UX Designer:** Focused on user interface and experience design.
* **Quality Assurance (QA):** Ensuring application stability and usability.
* **Customer Support:** Handling customer inquiries, returns, and complaints.
* **Resource Availability:**
* The project has an experienced team and access to necessary tools.
* Partnerships with payment gateways and hosting services are in place.

# 3. External Interface Requirements

## 3.1 User Interface

The application provides the following interfaces for end-users:

1. **Customer Interface:**
   * **Login/Register Page:** Users can sign in, register, or reset their password.
   * **Home Page:** Displays featured products and categories.
   * **Product Detail Page:** Includes product images, descriptions, pricing, and reviews.
   * **Cart Page:** Lists selected items with options to update quantities or remove items.
   * **Checkout Page:** Captures user address, payment information, and confirms orders.
   * **Order Tracking Page:** Displays current order status and estimated delivery time.
2. **Admin Interface:**
   * **Dashboard:** Overview of sales, orders, and user activity.
   * **Product Management:** CRUD operations for product listings.
   * **Order Management:** View, update, and process customer orders.

## 3.2 Software Interfaces

The e-commerce platform integrates with several software systems:

1. **Frontend Framework:**
   * React.js for the user interface, bundled with Webpack.
2. **Backend API:**
   * API’s are developed using Node.js and Express.js.
3. **Database:**
   * PostgreSQL as the primary relational database for storing product, user, and order data.
4. **Third-Party Services:**
   * **Cloudinary** for image hosting and optimization.
   * **NodeMailer** for sending notifications (SMS/Email).
5. **Operating System Compatibility:**
   * Compatible with modern browsers on Windows, macOS, iOS, and Android.

# 4. System Features

## 4.1 Admin Features

- View weekly sales reports with visual charts.

- Easily sort and manage products, categories, and orders.

- Add or update products and categories seamlessly.

- View customer profiles and manage account settings.

## 4.2 User Features

- Quickly search for products and categories.

- Add items to the cart and preview before purchase.

- Apply Coupon to your cart and avail discounts.

- Track orders in real-time and access order history.

# 5. Functional Requirements

## 5.1 Admin Features

- View a chart report based on the sales of the previous week

- View products and sort them efficiently based on parameters such as category, product name, product id, status (stock/out of stock)

- Add or Update product

- View orders and sort them efficiently based on parameters such as order id, customer id, from date, to date, status (shipping/delivered)

- View any specific order along with its specific details such as shipping details, payment details, item details, order details

- Add categories and have the option to sort them on the basis of your category id

- View all customers and have the option to search or sort on the basis of customer id or customer name

- Manage your profile, update your name, email, or password

## 5.2 User Features

- Search by product name and category name

- Preview items you want to buy in the cart

- See currently the most sold and best reviewed items in the store

- Can avail coupons of different tier that give you various different discounts

- Track your order currently, allowing you to see its current status

- See your order history, showing you all the orders you have previously placed

- Manage your profile, allowing you change your name, email, password and phone number

- Option to pay amount with three different card options (Stripe, PayPal, Razorpay)

# 6. Non-Functional Requirements

The system must meet the following non-functional requirements to ensure reliability, performance, and security:   
  
- Security: All user data should be encrypted in transit and at rest. User passwords should be hashed using industry-standard algorithms.

- Availability: The system should have 99.9% uptime.

- Usability: The user interface should be easy to navigate, with minimal learning curve for both admins and customers.

- Scalability: The system should support future growth in terms of users, products, and orders.

# 7.Conclusion

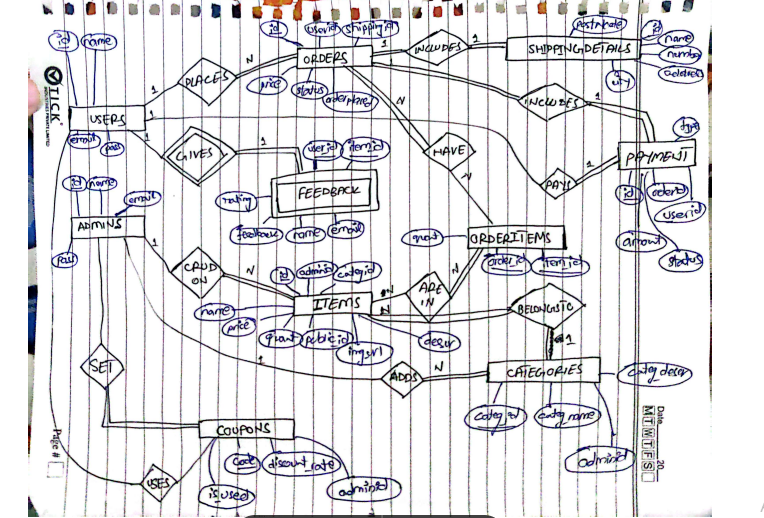
The Software Requirements Specification for **MartCom** provides a comprehensive overview of the system's purpose, functionality, and technical requirements. By addressing both functional and non-functional aspects, this document serves as a blueprint for the development team and a reference for stakeholders.

The outlined requirements ensure that **MartCom** will deliver a seamless shopping experience for customers and an efficient management platform for administrators. The integration of modern technologies, scalability, and secure practices ensures the system will meet current demands and adapt to future growth.

By following this SRS, the development team aims to deliver a high-quality e-commerce platform that exceeds expectations, ensuring a successful launch and ongoing support for **MartCom.**

**Appendix B: Analysis Models**

**Entity Relationship Diagram**

****

**Sequence Activity systems architecture**

* **Three-Tier Architecture:**

A three-tier architecture divides the system into three layers: presentation, logic, and data.

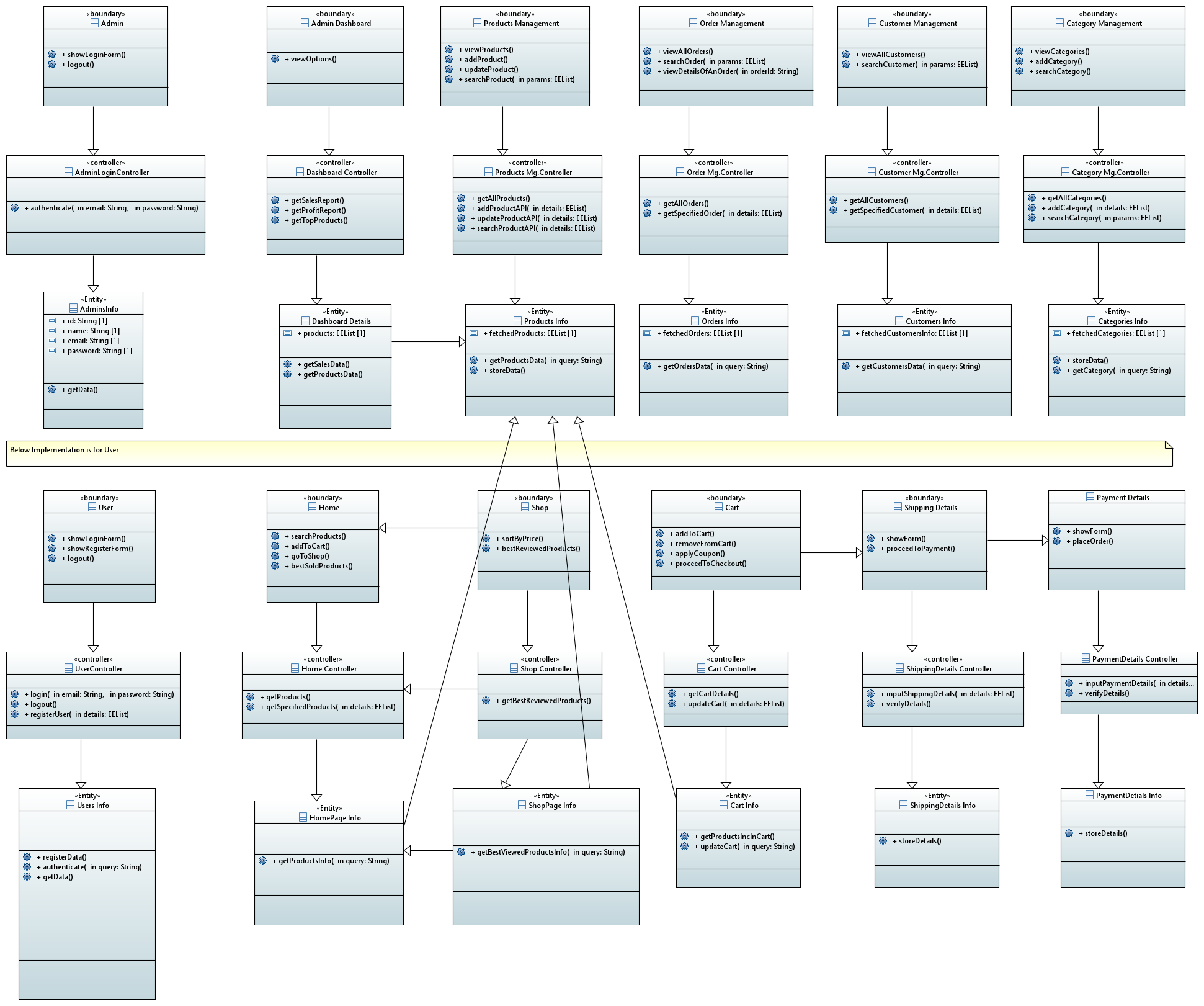
**How it Applies to MartCom:**

* **Presentation Tier**: The **React application** that serves the user interface, displaying products, cart, and other UI components to the user.
* **Logic Tier**: The **API layer** (Express.js or Node.js back-end), which handles the business logic (user authentication, cart management, order processing, etc.).
* **Data Tier**: The **database** (PostgreSQL) that stores and retrieves data like user information, product details, and order histories.

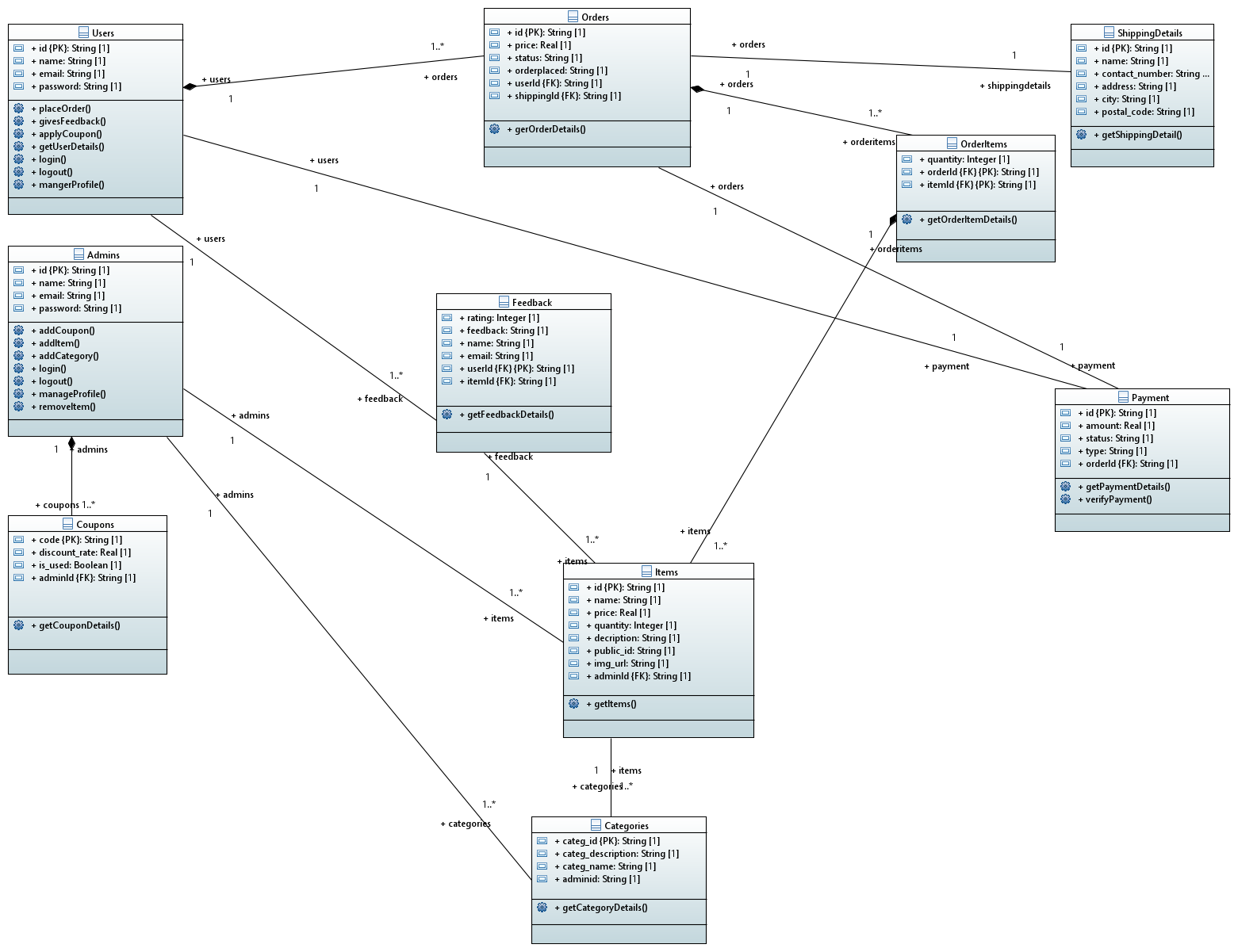
**Flow:**

1. The **client** (React app) sends requests to the **logic tier** (API).
2. The **logic tier** processes the request and communicates with the **data tier** (database) to retrieve or store information.
3. The **logic tier** sends back the processed data to the **presentation tier** (React app) for the user to interact with.

**Entity Control Boundary Class Diagram**

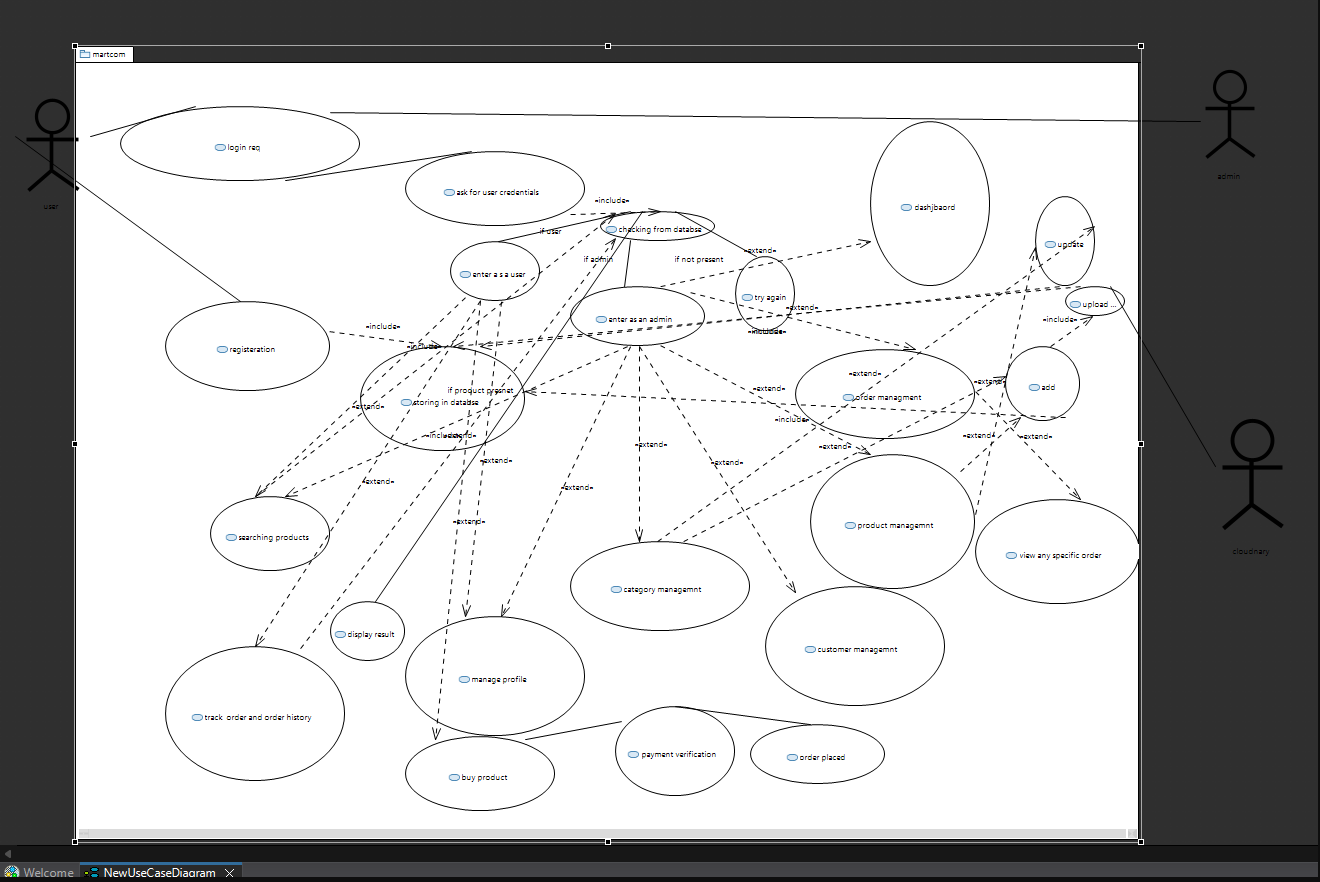


**Class Diagram**



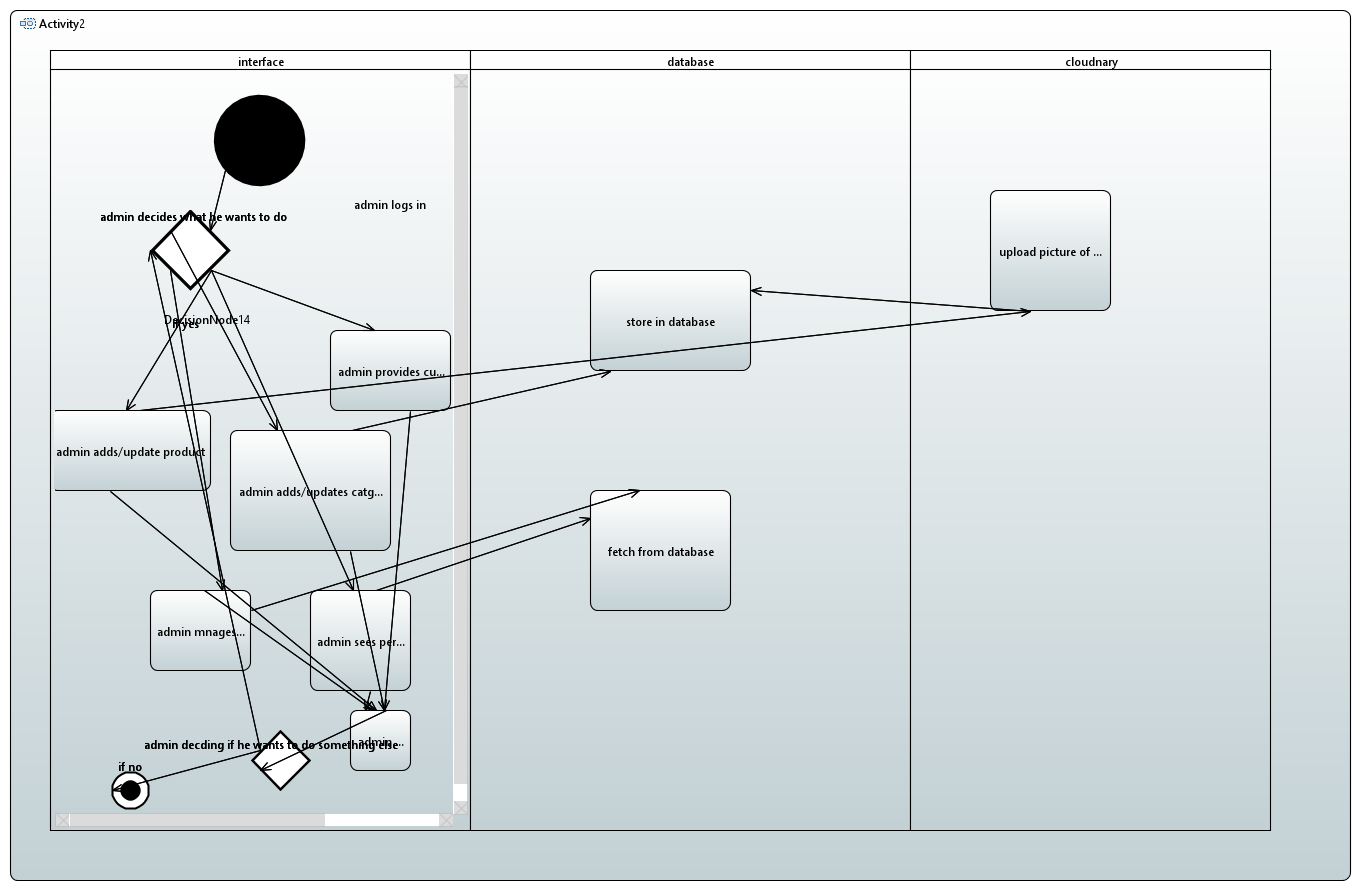
**Component Diagram**

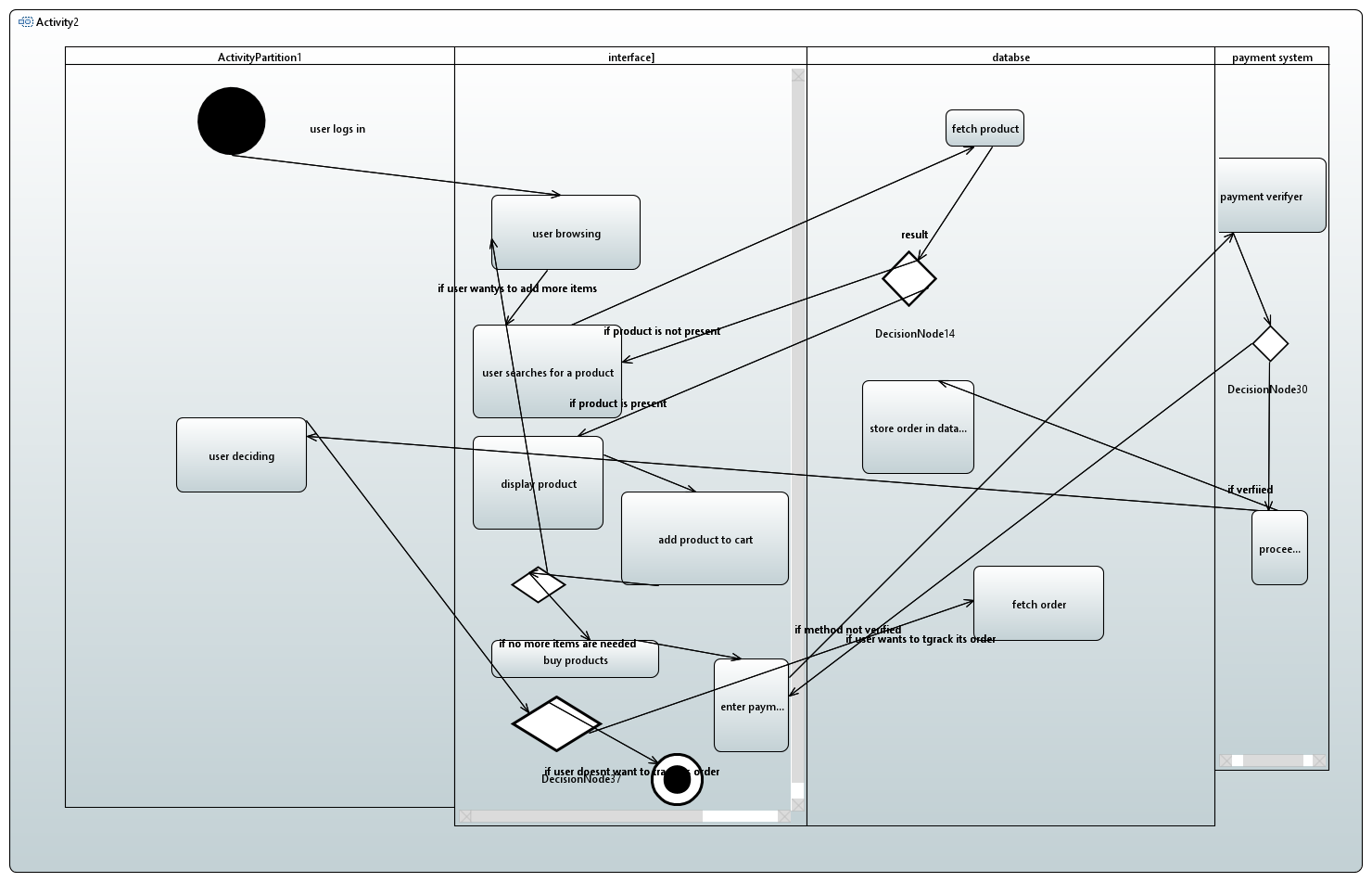
**Use Case diagram**



**Activity Diagram:**

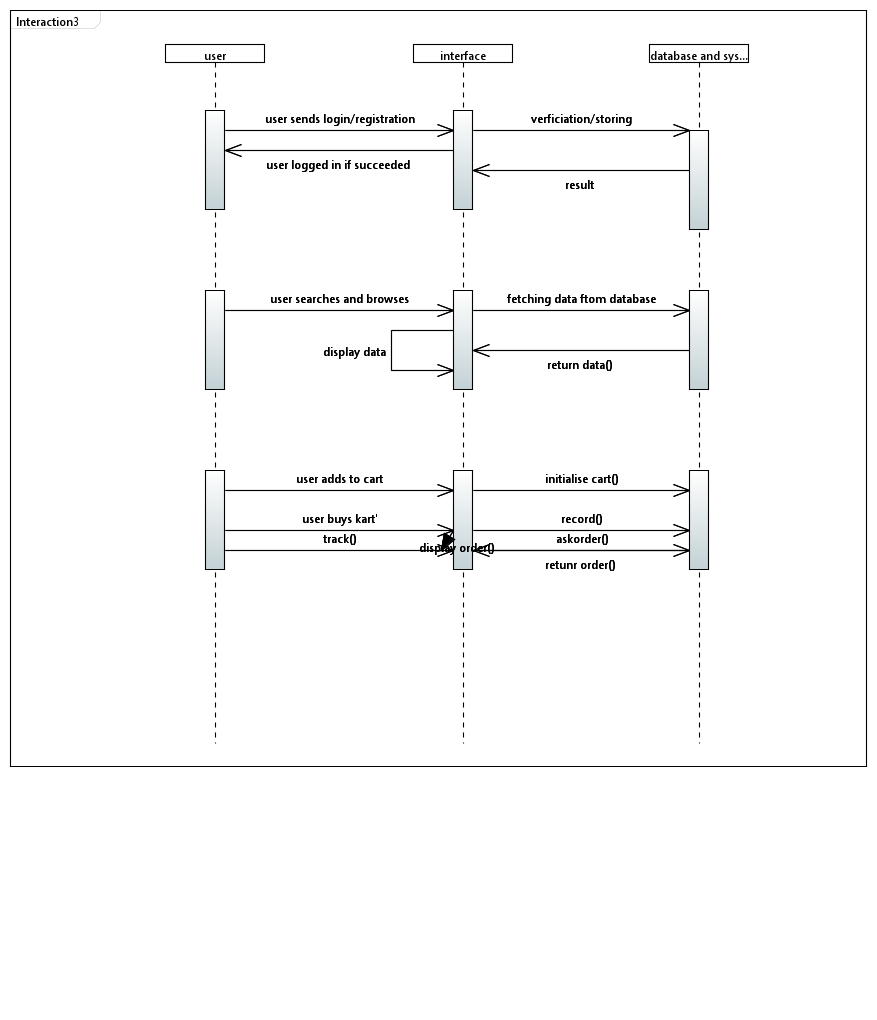
**Admin**:



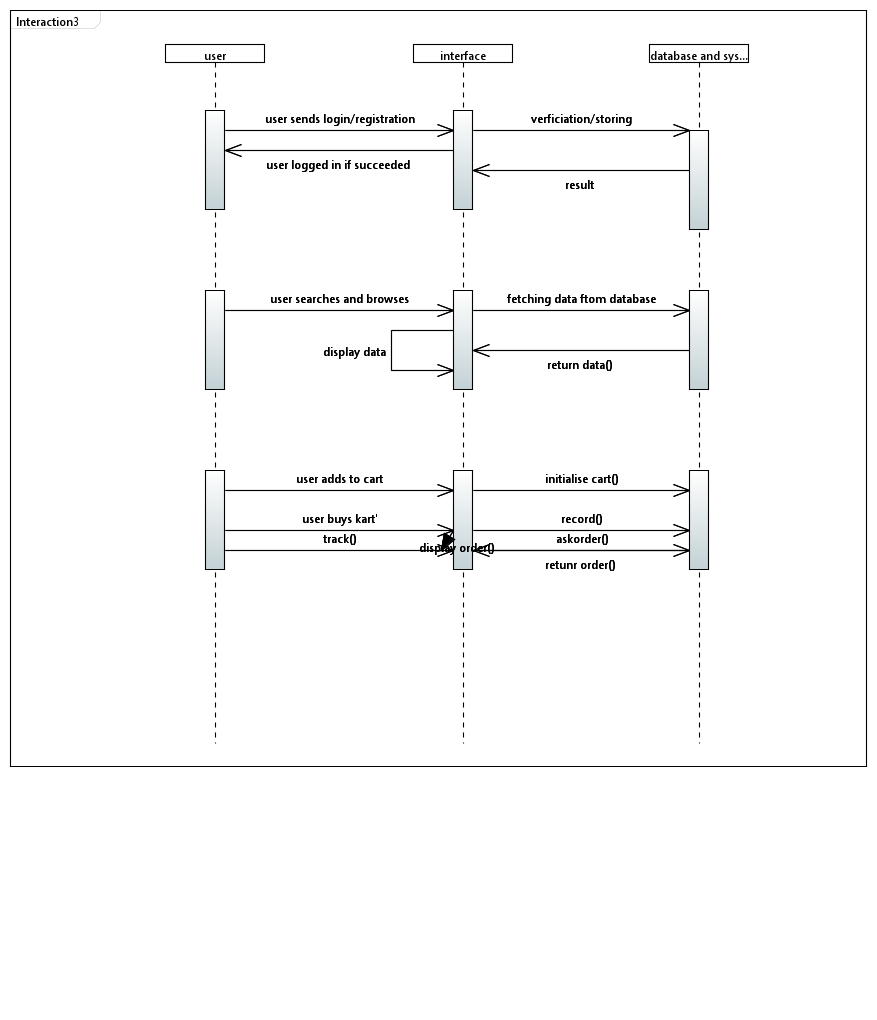
**User**: 

**Sequence Diagram:**

**Admin:**



**User:**

****

**Databse diagram:**

