Software Requirements Specification

For

Product Recommendation System

Version 1.0 approved/rejected

Prepared by: Muhammad Sohaib Raza,

Submitted to: Miss Iqra Qayyum

University of Punjab Gujranwala Campus

05-Jan-2025

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Purpose

The PRS is designed to enhance the user shopping experience by offering personalized product recommendations based on individual preferences, browsing history, and purchasing behavior. This system aims to improve customer satisfaction, increase sales, and support business objectives by delivering relevant product suggestions in real-time.

1.2 Product scope

The Product Recommendation System aims to analyze user preferences and behaviors to generate personalized product suggestions. It helps businesses enhance customer satisfaction, boost sales, and improve decision-making. The system integrates with e-commerce platforms and offers real-time recommendations, data insights, and exportable reports.

1.3 Intended audience and document overview

This document is intended for various stakeholders, including data scientists, business analysts, e-commerce managers, system developers, and testers. By reading this document, readers can understand the purpose of the Product Recommendation System and gain insights into its core concepts and functionality.

1.4 References

(Plant UML Text Editor, 2009)

1. PlantUML Text Editor. (2025, Jan 10). Retrieved from PlantUML Text Editor: https://www.planttext.com/

2 Overall description

2.1 Product perspective

The Product Recommendation System is designed as an intelligent solution that integrates seamlessly with existing e-commerce platforms. It analyzes user preferences, purchase history, and browsing behavior to generate personalized product recommendations.

2.2 Product function

The functionality of the product is as follows:

1. Data Ingestion and Preprocessing
2. Personalized Product Recommendations
3. Real-time Recommendation Updates
4. User Preference Analysis
5. Performance Monitoring
6. Data Visualization
7. Custom Reporting
8. Integration with E-commerce Platforms

2.3 User class and characteristics

The primary users of the Product Recommendation System include e-commerce managers, who oversee customer engagement metrics and use recommendations to drive sales and enhance user experience. Business analysts analyze the recommendation data to identify trends and opportunities for personalized marketing strategies. Additionally, data scientists, interact with the system to refine recommendation algorithms and enhance the accuracy and relevance of suggestions.

2.4 Operating environment

Operating System: Minimum requirements include Windows XP or Windows Vista. A better environment includes Windows 7, 8, 8.1, or 10.

Language: Python

The Product Recommendation System will be deployed in a cloud-based environment, ensuring scalability and flexibility to handle varying user demands. It will be accessible through web browsers such as Chrome, Firefox, and Edge on desktop and mobile devices. The system will integrate seamlessly with existing e-commerce platforms (e.g., Shopify, WooCommerce) through APIs. It will be compatible with data sources from various relational and non-relational databases to enable efficient data processing and recommendation generation.

2.5 Design and implementation constraints

These are some constraints that could increase costs for the system. Overcoming these constraints would result in optimal system performance. These are as follows:

1. **Data Quality:** Requires accurate and consistent user behavior and product data to generate relevant recommendations.
2. **Integration:** Must seamlessly integrate with various e-commerce platforms and third-party services.
3. **Scalability:** Needs to efficiently handle growing volumes of user and product data as the system expands.
4. **Accuracy:** Relies on precise recommendation algorithms to ensure relevant and personalized suggestions.
5. **Security:** Must protect sensitive user data and comply with privacy regulations.
6. **Budget:** Limited resources may impact the ability to implement advanced algorithms or scale the system effectively.

2.7 User documentation

It will provide specific guidelines for users on how to utilize the Product Recommendation System. Slide shows will be provided to demonstrate the system's features and how it works effectively for recommending products.

2.8 Assumptions and Dependencies

The Product Recommendation System relies on clean, structured, and up-to-date user behavior and product data to deliver effective recommendations. It requires integration with e-commerce platforms and third-party services for seamless data flow. The system depends on reliable cloud infrastructure for data storage and processing, as well as the continuous availability of third-party APIs for user data, product catalogs, and external sources to enhance recommendations.

3 External interface requirements

There are many types of interfaces as such supported by this software system namely;

User interface, software interface and hardware interface.

3.1 User interfaces

The system will provide an intuitive web-based interface for users, allowing them to view and interact with personalized product recommendations.

3.2 Hardware interfaces

The system will require minimal hardware interaction, as it is cloud-based and relies on the infrastructure of the hosting environment.

3.3 Software interfaces

The system will integrate with various e-commerce platforms (e.g., Shopify, WooCommerce) via APIs, enabling seamless data exchange and recommendation generation.

3.4 Communication interfaces

The system will use HTTPS and RESTful APIs for secure communication and data transfer between users, the recommendation engine, and integrated e-commerce platforms.

4 System Features

System features contain different functional requirements of the system

4.1 Functional Requirements

1. **Data Ingestion:** The system must be able to import and process user behavior and product data from various e-commerce platforms and databases.
2. **Product Recommendations:** The system must generate personalized product recommendations for users based on their behavior, preferences, and past interactions using machine learning algorithms.
3. **Recommendation Optimization:** The system must provide optimized product suggestions to maximize sales and customer satisfaction by considering factors like inventory levels, trending products, and user preferences.
4. **Reporting:** The system must generate customizable reports and data visualizations for product recommendation performance, including conversion rates and user engagement.
5. **User Management:** The system must allow administrators to manage user roles and permissions for secure access to recommendation system features.
6. **Data Export:** The system must support exporting recommendation data and reports in various formats, such as CSV, PDF, and Excel.
7. **Real-Time Updates:** The system must provide real-time updates on user interactions and product recommendations, allowing businesses to adapt and personalize offerings dynamically.

4.1.1 USE CASE MODEL OF Product Recommendation System

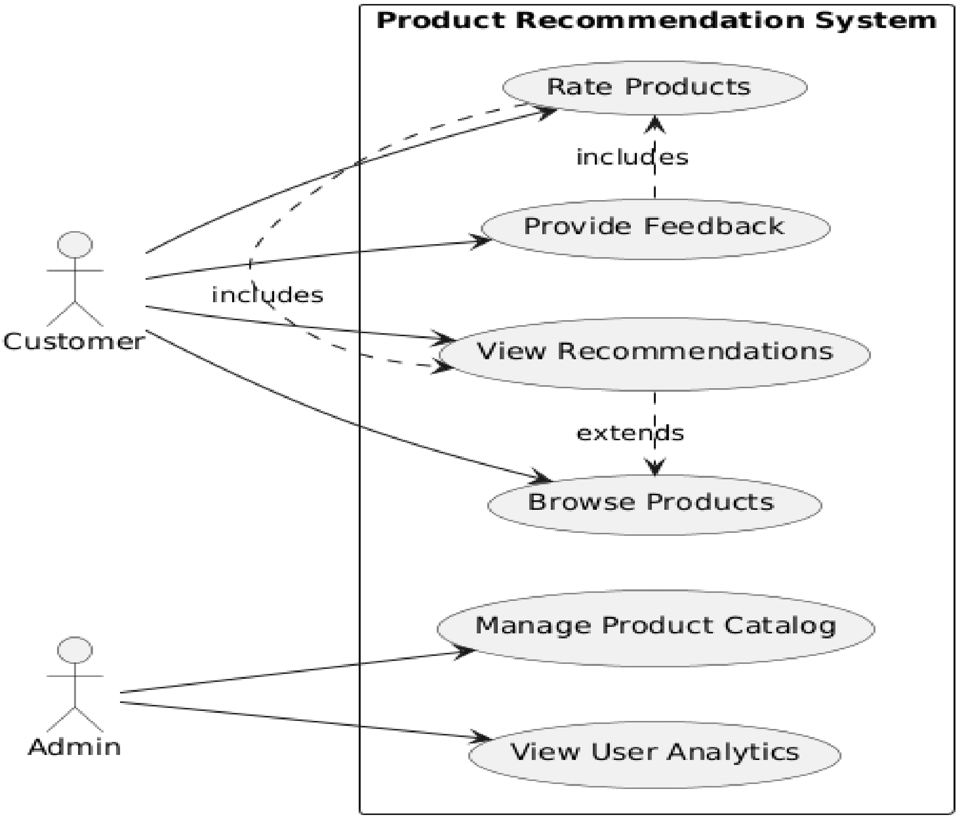


Figure 1: Use case Model

4.1.2 Scenarios of Use Case

Table 1: Import User and Product data

Use case:	Import User and Product data
Actor	System
Pre-condition	Data from external platforms and user interaction history is available and accessible.
Normal Flow	The system imports user behavior, product interaction, and inventory data from integrated e-commerce platforms and databases.
Post condition	User and product data is successfully imported into the system for generating recommendations.

Table 2: Generate Product Recommendation

Use case:	Generate Product Recommendations
Actor:	System
Pre-condition:	Historical user and product data is available and processed.
Normal flow:	The system analyzes user data (purchases, browsing history) and applies machine learning models to generate personalized product recommendations.
Post-condition:	Personalized product recommendations are generated and presented to the user

Table 3: Personalize Recommendation-Based on Context

Use case:	Personalize Recommendations Based on Context
Actor:	System

Pre-condition:	User behavior data (current session) and product data are available.
Normal flow:	The system adapts the product recommendations in real-time based on factors like user browsing, cart additions, and external influences (e.g., seasonal trends).
Post-condition:	Dynamic, personalized recommendations are presented based on real-time data.

Table 4: Manage Recommendations

Use case:	Manage Recommendations
Actor:	Admin
Pre-condition:	Admin has appropriate permissions to modify recommendation settings.
Normal flow:	The admin reviews system performance, updates recommendation algorithms, and integrates new product data to improve recommendations.
Post-condition:	System settings and models are updated, ensuring more effective recommendations.

Table 5: Monitor Recommendation Performance

Use case:	Monitor Recommendation Performance
Actor:	System, Admin
Pre-condition:	Recommendation model is deployed and generating suggestions.
Normal flow:	The system tracks metrics such as click-through rates, purchase behavior, and user engagement. Admin monitors performance and adjusts the model as needed.
Post-condition:	Performance metrics are logged, and the recommendation model is refined based on results.

Table 6: Cross-Selling

Use case:	Payment
Actor:	User (Customer)
Pre-condition:	User adds an item to their cart.
Normal flow:	User adds a product to their cart. System suggests complementary products (e.g., charger for a laptop). User may choose to add suggested items to the cart.
Post-condition:	Complementary product is added to the cart (if accepted).

Table 7: Upselling

Use case:	Upselling
Actor:	User (Customer)
Pre-condition:	User is browsing or viewing a product.
Normal flow:	User views a product page. System identifies and recommends a higher-end version of the product. User may choose to switch to the premium product.
Post-condition:	Premium product is added to the cart (if accepted).

Table 8: Recently Viewed Products

Use case:	Recently Viewed Products
Actor:	User (Customer)
Pre-condition:	User has browsed one or more products.

Normal flow:	1. User navigates to the homepage or product category. 2. System displays a carousel of recently viewed items. 3. User may revisit a product or proceed to purchase.
Post-condition:	User re-engages with previously viewed items.

Table 9: New Arrivals and Trending Products

Use case:	New Arrivals and Trending Products
Actor:	User (Customer)
Pre-condition:	User visits the homepage or product category.
Normal flow:	1. User opens the homepage. 2. System highlights new arrivals or trending products. 3. User clicks on an item for more details or purchase.
Post-condition:	User interacts with the highlighted products.

Table 10: Wishlist-Based Recommendations

Use case:	Wishlist-Based Recommendations
Actor:	User (Customer)
Pre-condition:	User has items saved in their wishlist.
Normal flow:	1. User logs in and accesses their wishlist. 2. System recommends related or updated products based on wishlist items. 3. User selects items for purchase.
Post-condition:	Recommended items are added to the cart (if accepted).

4.1.3 Sequence Diagrams

Sequence diagrams of Product Recommendation system are as follows:

Figure 2: Import User and Product Data

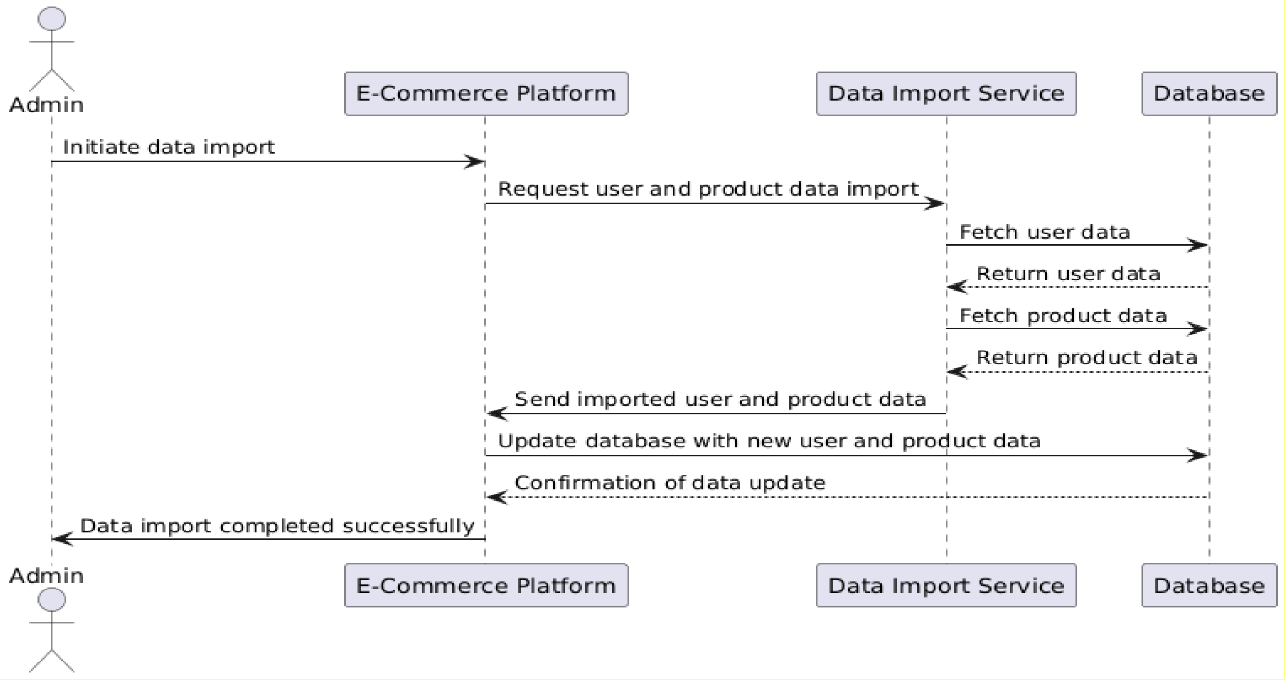


Figure 3: Generate Product Recommendations

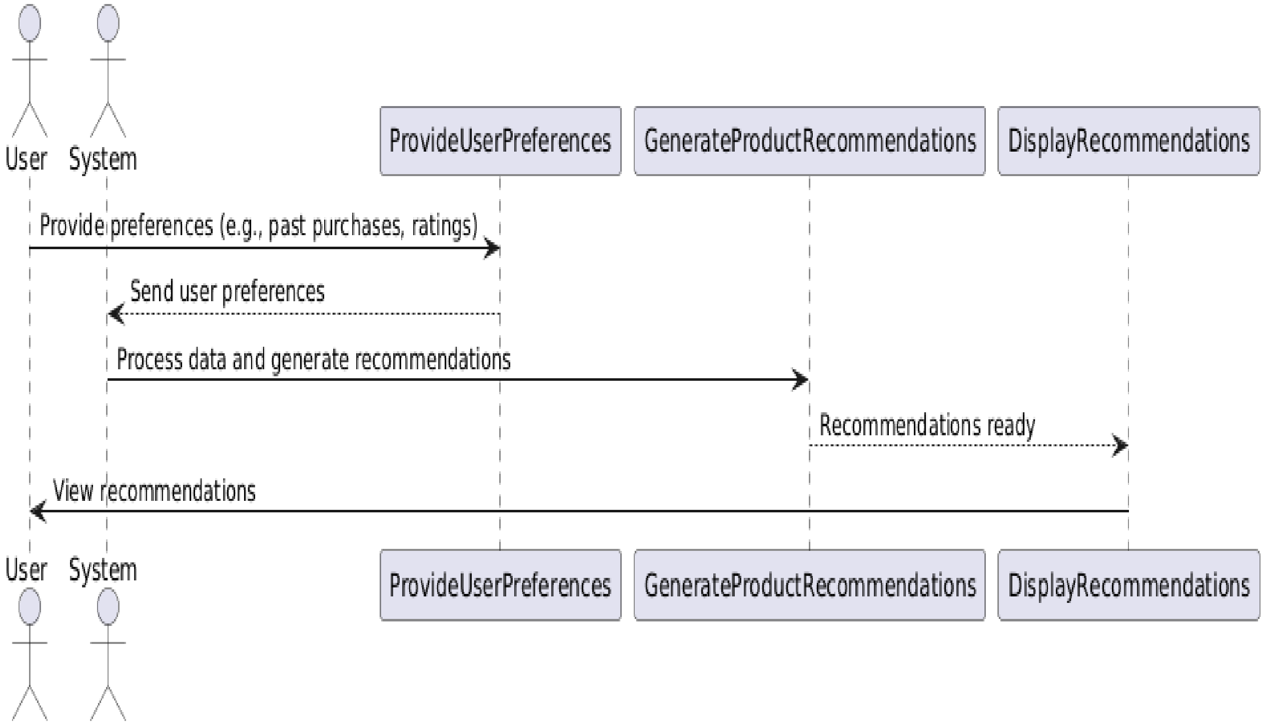


Figure 4: Personalize Recommendation Based on Context

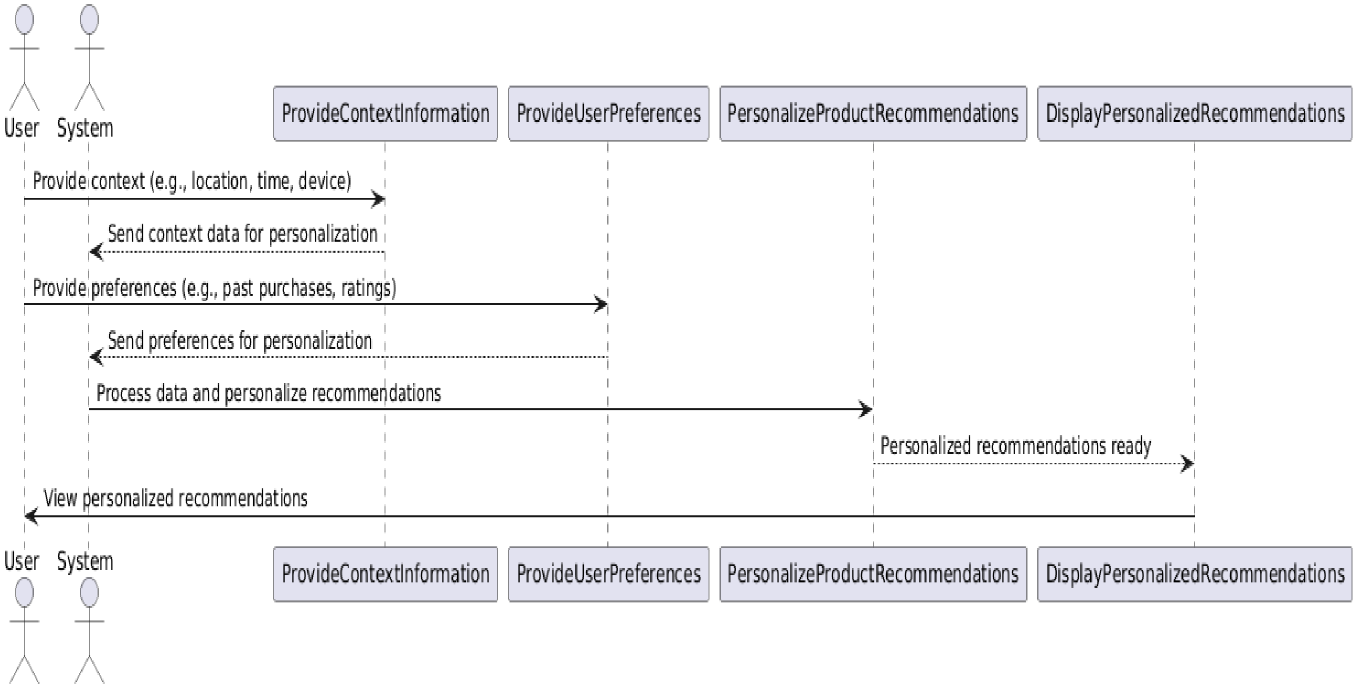

User
System
ProvideContextInformation
ProvideUserPreferences
PersonalizeProductRecommendations
DisplayPersonalizedRecommendations
Provide context (e.g., location, time, device)
Send context data for personalization
Provide preferences (e.g., past purchases, ratings)
Send preferences for personalization
Process data and personalize recommendations
Personalized recommendations ready
View personalized recommendations
User
System
ProvideContextInformation
ProvideUserPreferences
PersonalizeProductRecommendations
DisplayPersonalizedRecommendations

Figure 5: Manage Recommendations

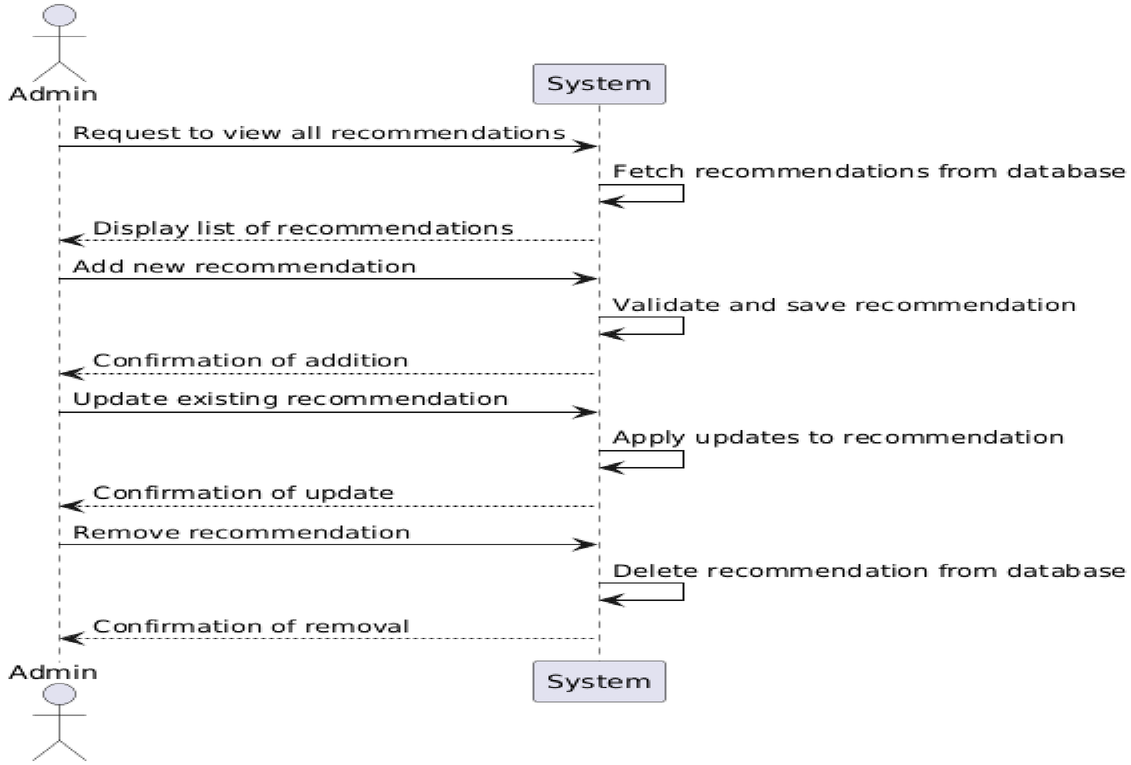

Admin
System
Request to view all recommendations
Fetch recommendations from database
Display list of recommendations
Add new recommendation
Validate and save recommendation
Confirmation of addition
Update existing recommendation
Apply updates to recommendation
Confirmation of update
Remove recommendation
Delete recommendation from database
Confirmation of removal
Admin
System

Figure 6: Monitor Recommendation Performance

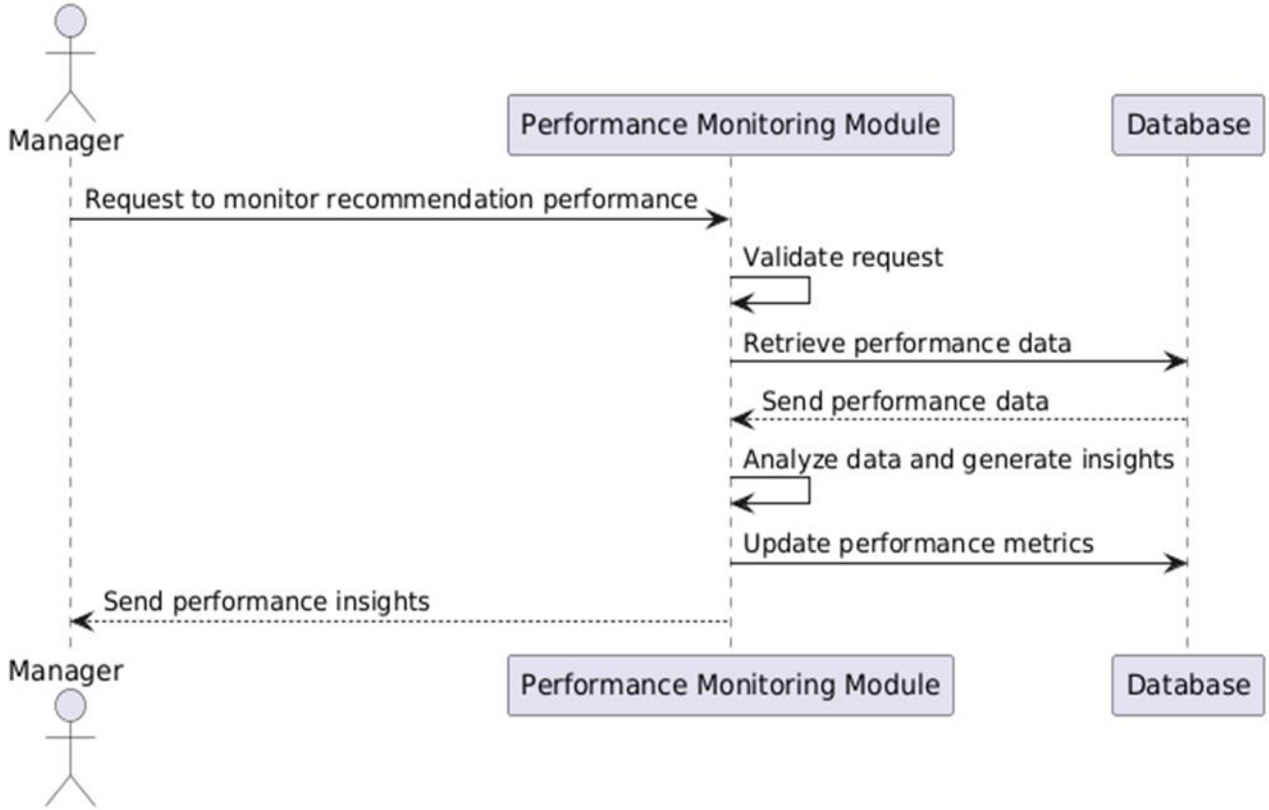

Manager
Performance Monitoring Module
Database
Request to monitor recommendation performance
Validate request
Retrieve performance data
Send performance data
Analyze data and generate insights
Update performance metrics
Send performance insights
Manager
Performance Monitoring Module
Database

Figure 7: Cross Selling

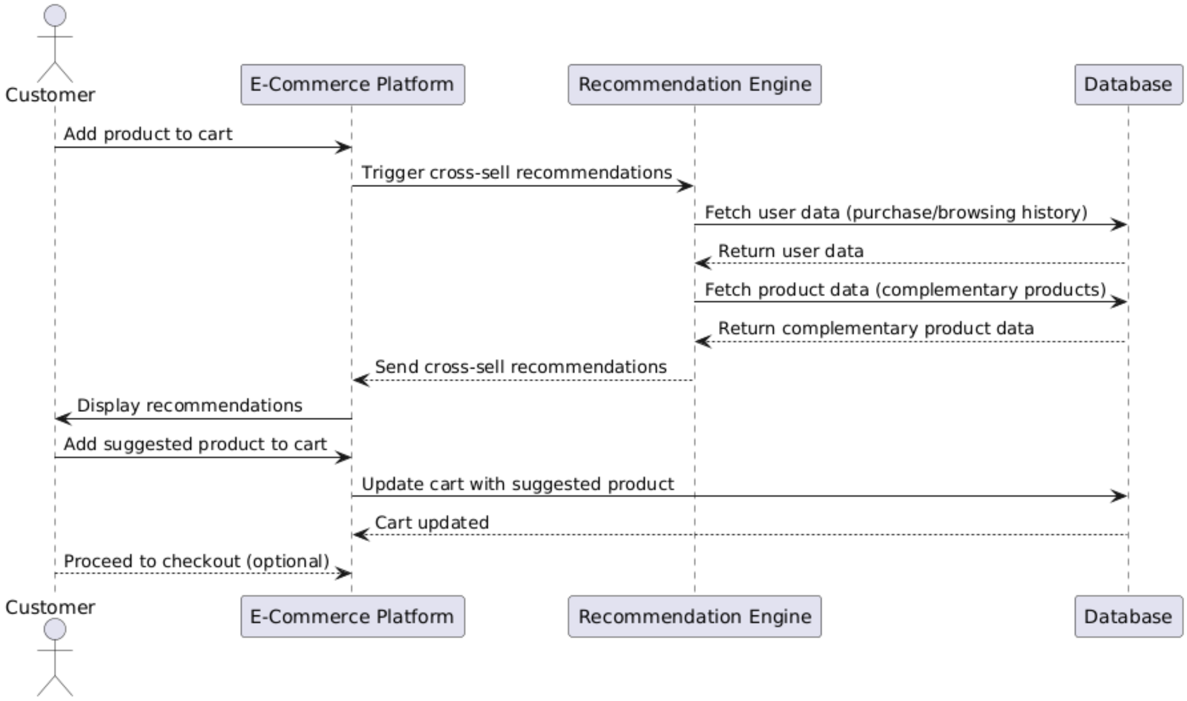

Customer
E-Commerce Platform
Recommendation Engine
Database
Add product to cart
Trigger cross-sell recommendations
Fetch user data (purchase/browsing history)
Return user data
Fetch product data (complementary products)
Return complementary product data
Send cross-sell recommendations
Display recommendations
Add suggested product to cart
Update cart with suggested product
Cart updated
Proceed to checkout (optional)
Customer
E-Commerce Platform
Recommendation Engine
Database

Figure 8: Upselling

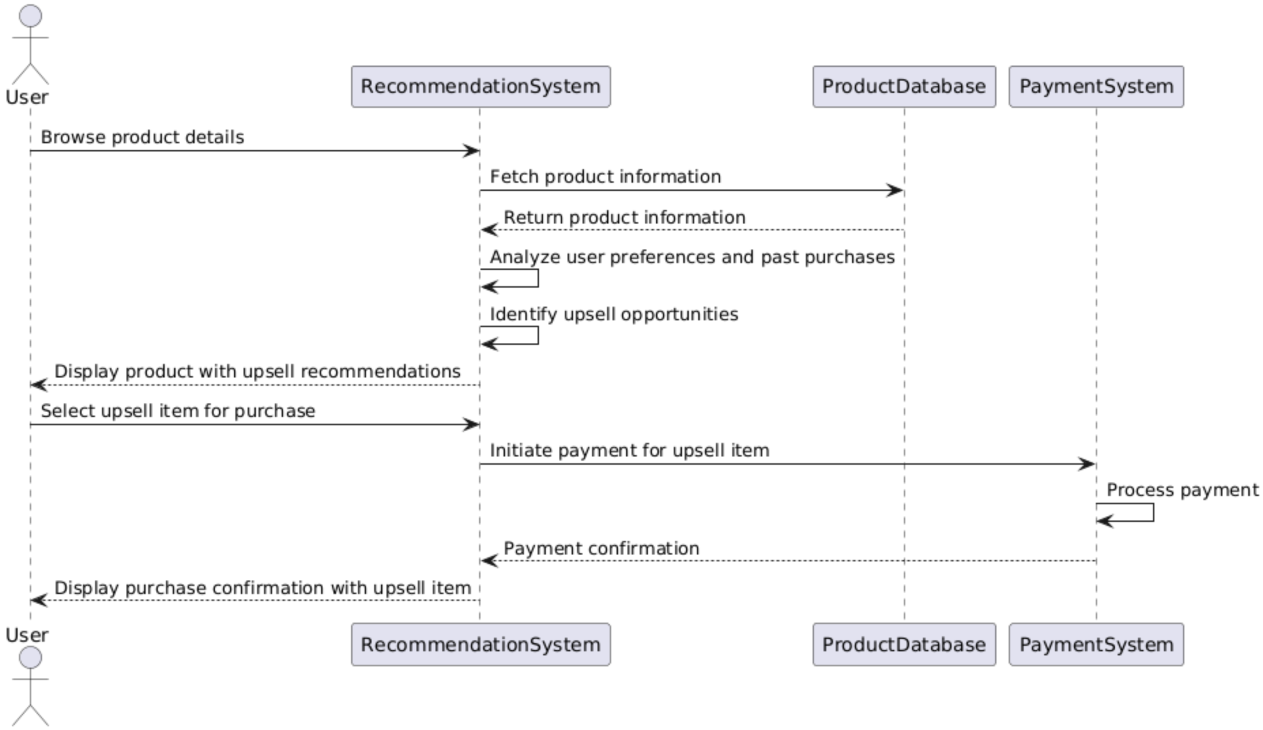

User
RecommendationSystem
ProductDatabase
PaymentSystem
Browse product details
Fetch product information
Return product information
Analyze user preferences and past purchases
Identify upsell opportunities
Display product with upsell recommendations
Select upsell item for purchase
Initiate payment for upsell item
Process payment
Payment confirmation
Display purchase confirmation with upsell item
User
RecommendationSystem
ProductDatabase
PaymentSystem

Figure 9: Recently Viewed Products

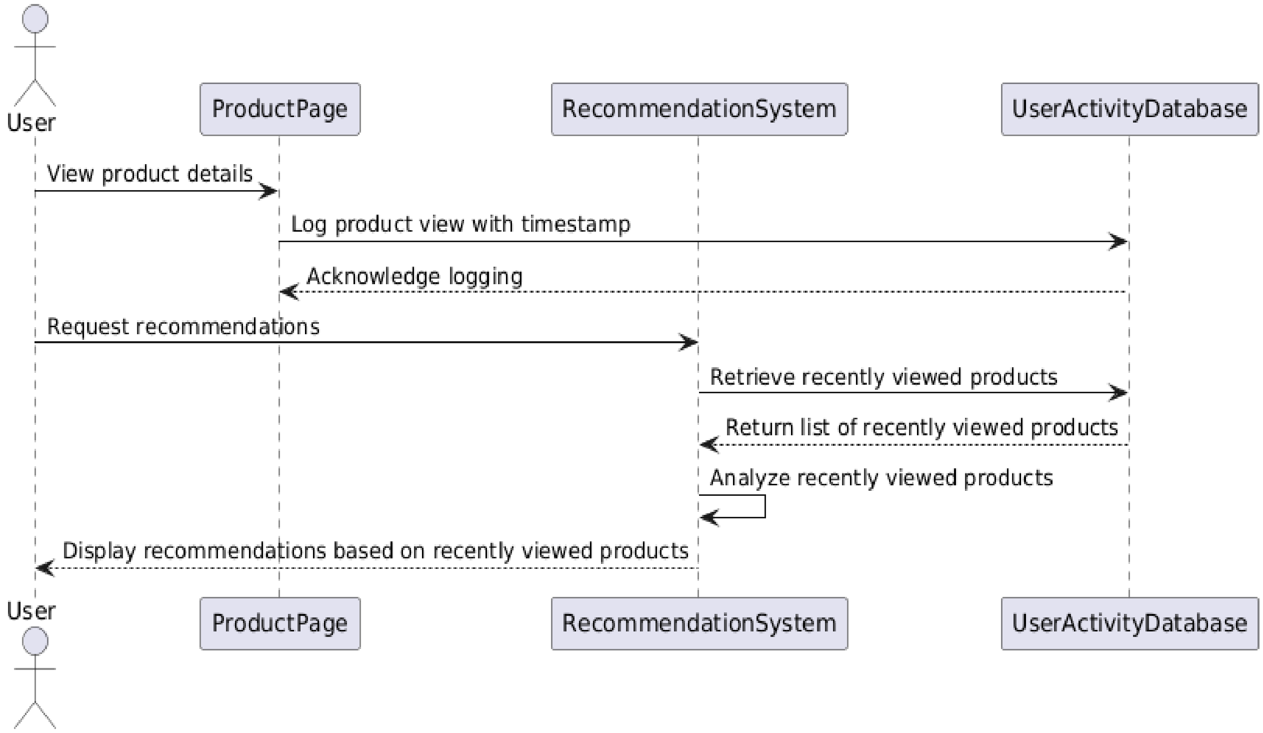

User
ProductPage
RecommendationSystem
UserActivityDatabase
View product details
Log product view with timestamp
Acknowledge logging
Request recommendations
Retrieve recently viewed products
Return list of recently viewed products
Analyze recently viewed products
Display recommendations based on recently viewed products
User
ProductPage
RecommendationSystem
UserActivityDatabase

Figure 10: New Arrivals and Trending Products

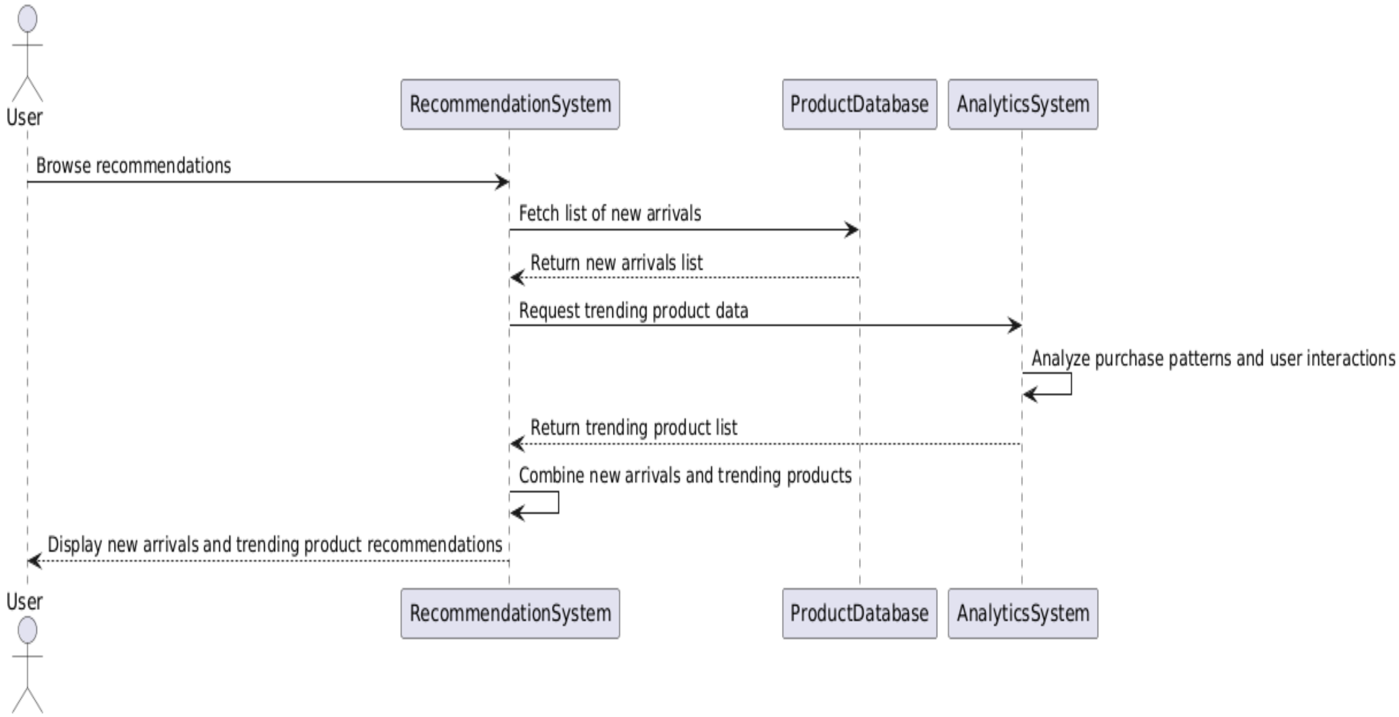
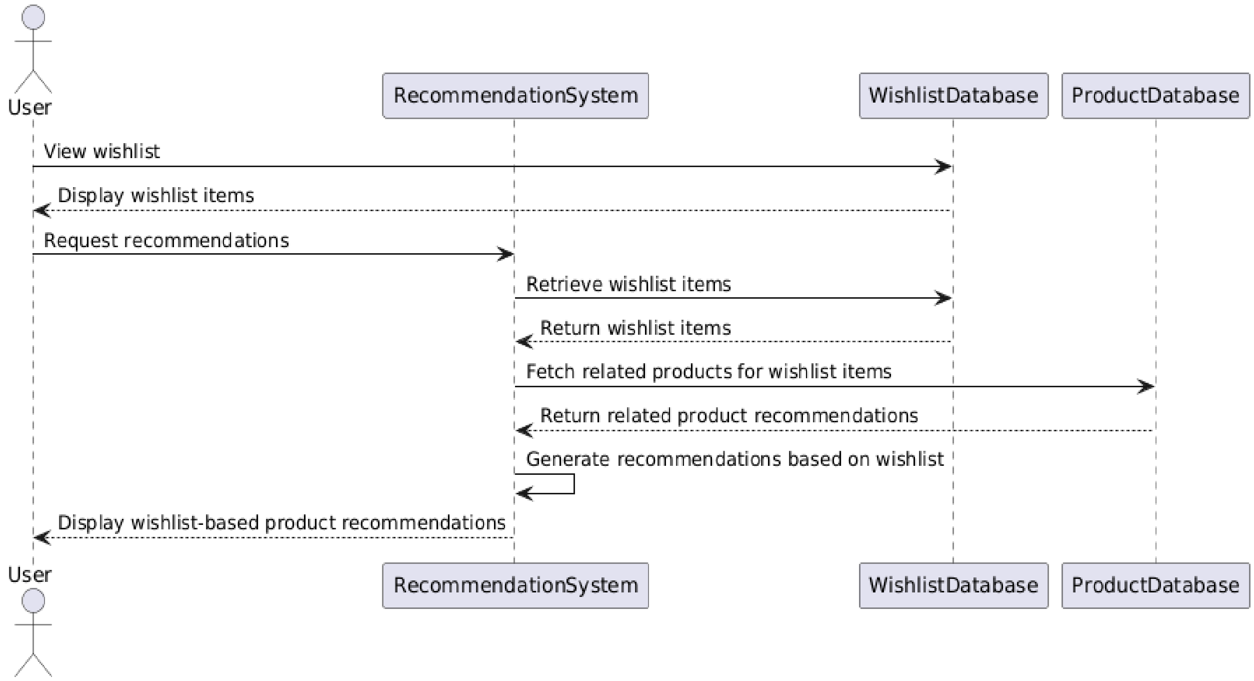

User
RecommendationSystem
ProductDatabase
AnalyticsSystem
Browse recommendations
Fetch list of new arrivals
Return new arrivals list
Request trending product data
Analyze purchase patterns and user interactions
Return trending product list
Combine new arrivals and trending products
Display new arrivals and trending product recommendations
User
RecommendationSystem
ProductDatabase
AnalyticsSystem

Figure 11: Wishlist Based Recommendations



4.1.4 Context Flow Diagram

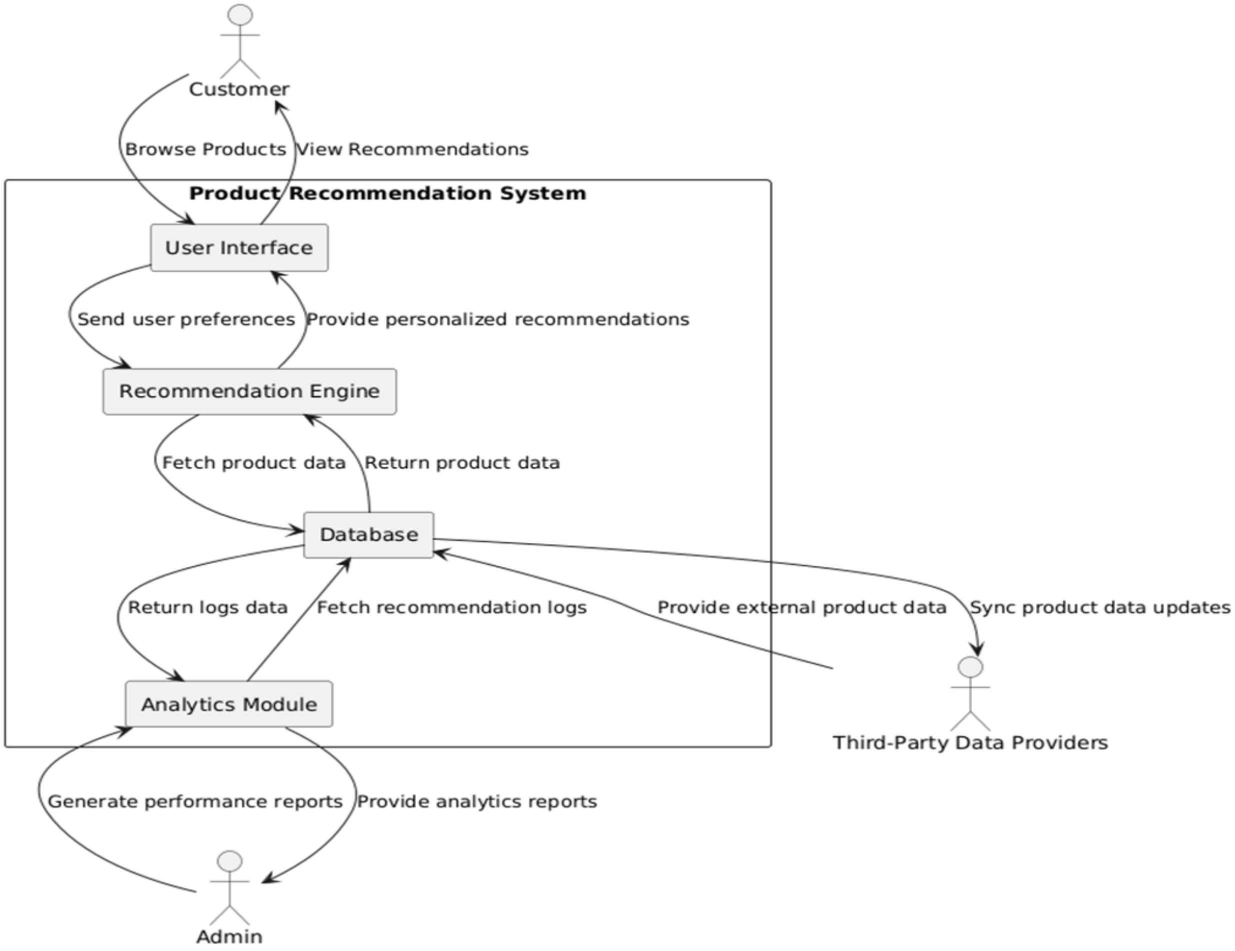

Customer
Browse Products
View Recommendations
Product Recommendation System
User Interface
Send user preferences
Provide personalized recommendations
Recommendation Engine
Fetch product data
Return product data
Database
Return logs data
Fetch recommendation logs
Provide external product data
Sync product data updates
Analytics Module
Third-Party Data Providers
Generate performance reports
Provide analytics reports
Admin

Figure 12: CFD of Product Recommendation System

The Product Recommendation System depends on high-quality, clean, and structured user behavior and product data, as the effectiveness of its recommendations is directly tied to the quality of this input. It requires integration with various e-commerce platforms and third-party services to ensure seamless, real-time data flow for accurate product suggestions.

The diagram outlines the workflow of a **Product Recommendation System**:

- **Customer**: Interacts with the system to browse products and view recommendations.
- **User Interface (UI)**: Collects user preferences and displays personalized recommendations.
- **Recommendation Engine**: Processes user data and retrieves product information from the database.
- **Database**: Stores product data and logs, enabling data retrieval and updates.
- **Analytics Module**: Analyzes logs to generate performance and analytics reports.
- **Admin**: Accesses analytics reports for decision-making.
- **Third-Party Data Providers**: Sync and update external product data in the database.

4.1.5 Activity Diagram

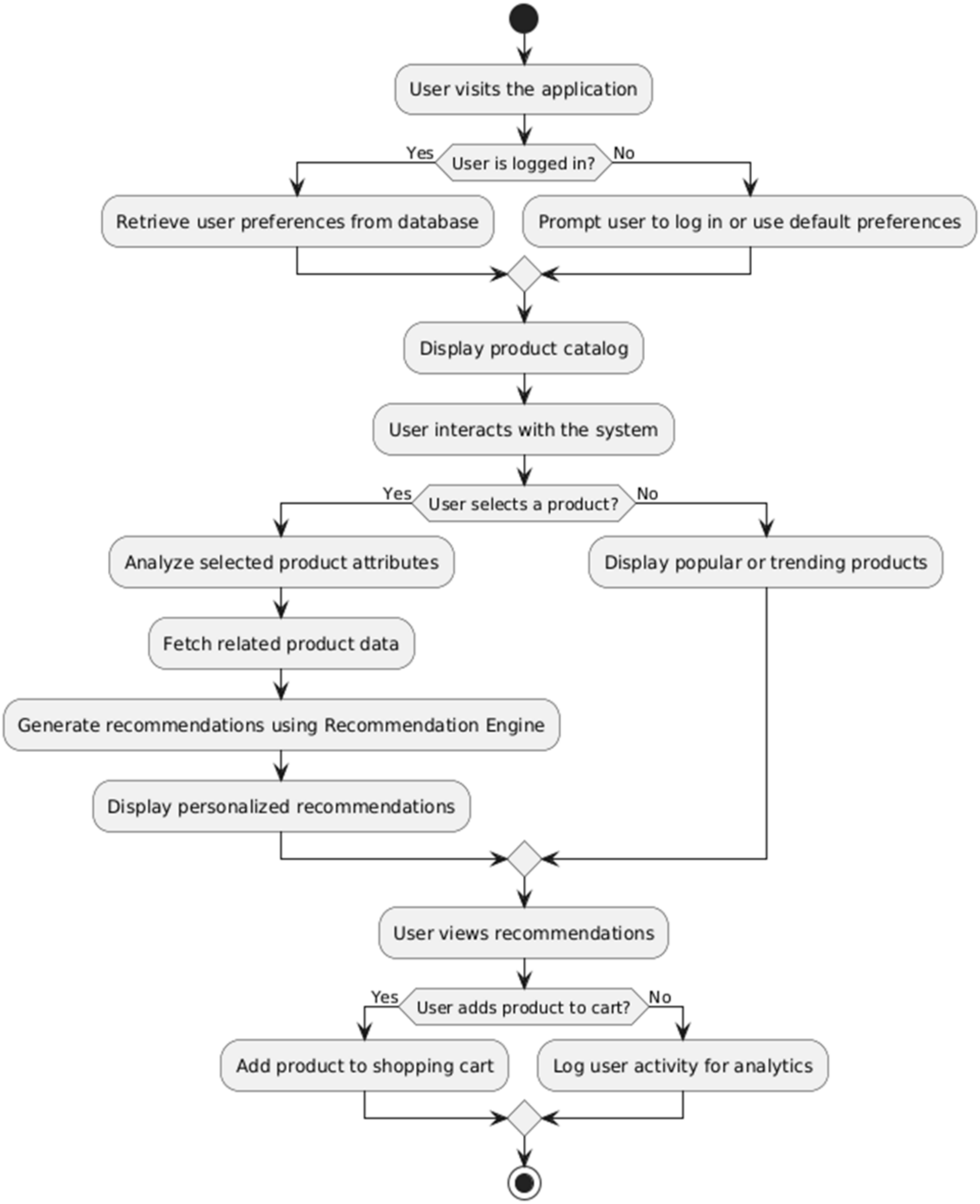

User visits the application
User is logged in?
Yes
No
Retrieve user preferences from database
Prompt user to log in or use default preferences
Display product catalog
User interacts with the system
User selects a product?
Yes
No
Analyze selected product attributes
Display popular or trending products
Fetch related product data
Generate recommendations using Recommendation Engine
Display personalized recommendations
User views recommendations
User adds product to cart?
Yes
No
Add product to shopping cart
Log user activity for analytics

Figure 13: Activity Diagram of PRS

4.1.6 Class Diagram

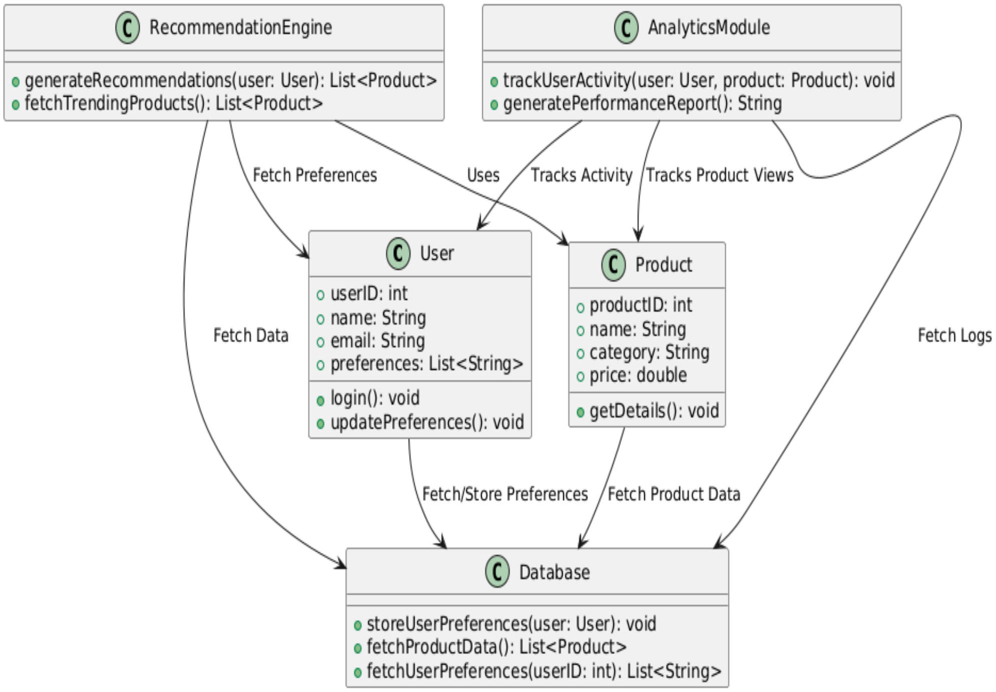

RecommendationEngine
generateRecommendations(user: User): List<Product>
fetchTrendingProducts(): List<Product>
AnalyticsModule
trackUserActivity(user: User, product: Product): void
generatePerformanceReport(): String
Fetch Preferences
Uses
Tracks Activity
Tracks Product Views
User
userID: int
name: String
email: String
preferences: List<String>
login(): void
updatePreferences(): void
Product
productID: int
name: String
category: String
price: double
getDetails(): void
Fetch Data
Fetch Logs
Fetch/Store Preferences
Fetch Product Data
Database
storeUserPreferences(user: User): void
fetchProductData(): List<Product>
fetchUserPreferences(userID: int): List<String>

Figure 14: Class Diagram Of PRS

4.2 Behavior requirements

4.2.1 Use case view:

The use case for each of the actor is described below:

Actors:

i. **E-commerce Manager:**
Interacts with the system to view product recommendations, optimize inventory, and analyze
recommendation performance reports.

ii. **Business Analyst:**
Interacts with the system to generate and analyze reports based on product recommendation data.

System:

The central system responsible for processing user behavior, purchase history, and product data to

generate personalized product recommendations and manage relevant data.

External API (e.g., Product Ratings or Market Trend Data):

Provides supplementary data, such as product ratings, reviews, and market trends, to enhance the accuracy and relevance of product recommendations.

4.2.2. Use Cases:

Import User and Product Data:

a) **Description**:
The system retrieves user behavior data (e.g., browsing history, purchase history) and product
Details (e.g., descriptions, categories, ratings) from external sources (e.g., e-commerce platforms).

b) **Actor**:
System

c) **Pre-condition**:
The external system or platform is available for data retrieval.

d) **Post-condition**:
User and product data are stored in the system and ready for analysis by the recommendation
engine.

Generate Product Recommendation:

a) **Description**:
The system processes the user and product data, applies recommendation algorithms (e.g.,
collaborative filtering or content-based filtering), and generates personalized product recommendations.

b) **Actor**:
E-commerce Manager, Business Analyst

c) **Pre-condition**:
Valid user and product data are available.

d) **Post-condition**:
Product recommendations are generated and stored in the system for display or analysis.

Personalize Recommendation based on Context:

a) **Description**:
The system dynamically adjusts and personalizes product recommendations based on the user's
current context (e.g., location, time of day, device type, or ongoing promotions).

b) **Actor**:
E-commerce Manager

c) **Pre-condition**:
User and product data are available, and contextual data (e.g., session details or external factors)
has been retrieved.

d) **Post-condition**:
The personalized product recommendations are updated in real-time and displayed to the user.

Generate Report

a. Description:
Business Analysts generate detailed reports based on product recommendation performance,
such as click-through rates, conversion rates, and user engagement metrics.

b. Actor:
Business Analyst

c. Pre-condition:
User, product, and recommendation data are available.

d. Post-condition:
Reports are generated and stored for review and strategic decision-making.

Manage Recommendation

a. Description:
The system allows the E-commerce Manager to manage and fine-tune the product recommendation settings. This includes configuring recommendation algorithms, adjusting
weights for specific factors (e.g., user behavior, product popularity), and monitoring performance metrics.

b. Actor:
E-commerce Manager

c. Pre-condition:
The recommendation system is operational, and valid user and product data are available.

d. Steps:

i. The E-commerce Manager accesses the recommendation management interface.

ii. The system displays available recommendation algorithms and settings (e.g., collaborative filtering, content-based filtering).

iii. The manager adjusts settings or chooses a new algorithm to implement.

iv. The system applies the changes and updates recommendations in real-time.

v. The manager monitors the performance metrics to evaluate the effectiveness of the recommendations.

e. Post-condition:

The recommendation system is fine-tuned and optimized based on the manager’s inputs,
improving the relevance and accuracy of recommendations.

Monitor Recommendation performance

a. Description:

The system allows the Business Analyst or E-commerce Manager to monitor and evaluate the performance of the product recommendation system through detailed metrics and reports,
such as click-through rates, conversion rates, engagement levels, and sales influenced by recommendations.

b. Actor:

E-commerce Manager, Business Analyst

c. Pre-condition:

The recommendation system has been operational, and performance data has been collected.

d. Steps:

i. The actor logs into the system and navigates to the performance monitoring dashboard.

ii. The system displays key performance metrics, such as:

Click-through rate (CTR) for recommendations

Conversion rates for recommended products

Revenue generated from recommendations

Engagement metrics (e.g., average time spent on recommended pages)

iii. The actor selects specific time periods or filters (e.g., by product category or user segment) to analyze performance.

iv. The system generates and displays detailed graphs, reports, or trends based on the selected

v. criteria.

vi. If anomalies or areas for improvement are identified, the actor notes changes to be

vii. implemented (e.g., adjusting algorithms or recommendation parameters).

e. Post-condition:

The actor gains actionable insights into the recommendation system's performance and identifies opportunities for optimization.

5 Non-functional Requirements

5.1 Performance Requirements:

- **Response Time**: The system should provide real-time or near-real-time recommendations with a

low latency (e.g., < 1 second for product recommendations after a user action).

- **Scalability**: The system must scale to handle increasing numbers of users, products, and data

points. It should be able to handle traffic spikes (e.g., during sales or promotional events).

- **Throughput**: The system should handle a high volume of requests without degradation in

performance. This includes handling thousands of product recommendations per second.

5.2 Security:

- **Data Privacy**: User data, such as browsing behavior and purchase history, must be handled in

accordance with relevant data privacy regulations (e.g., GDPR, CCPA).

- **Access Control**: Sensitive recommendation settings and performance data must be protected with

role-based access control to ensure only authorized users can make adjustments or view reports.

5.3 Availability:

- **Uptime:** The system should have high availability, ensuring that the recommendation engine is

online and accessible 99.9% or higher.

- **Fault Tolerance:** The system should be resilient to failures, with failover mechanisms in place to minimize downtime or data loss.

5.4 Reliability Requirements:

- **Consistency:** Recommendations must remain consistent across user sessions, meaning a user

should receive similar recommendations based on their behavior and profile.

- **Accuracy:** The recommendation system should produce accurate recommendations based on user

preferences, purchase history, and behavior.

5.5 Usability:

- **Ease of Use**: The interface for managing the recommendation system should be user-

friendly
for both technical and non-technical users, like the E-commerce Manager or Business Analyst.

- **Customization**: Users should be able to personalize or modify the recommendation settings,
such as adjusting the algorithm's sensitivity to certain factors (e.g., recency or popularity).

5.6 Cost Efficiency Requirements:

- **Resource Utilization**: The system should optimize resource usage, such as computational power, memory, and storage, to maintain performance without excessive operational costs.
- **Cloud Cost Management**: If using cloud infrastructure, the system should efficiently manage
cloud resources and minimize unnecessary expenditures.

5.7 Maintainability

- **Modularity:** The system should be modular, with separate components for recommendation generation, data processing, and reporting. This allows for easier updates and improvements.
- **Error Handling:** The system should be able to detect and report errors in the recommendation process (e.g., if the model fails to generate recommendations) and allow quick troubleshooting.
- **Documentation:** Proper documentation should be available for both the recommendation algorithms and the system architecture for easier maintenance and updates.

6 Other Requirements

- **Licensing Requirements:**
Not applicable.
- **Legal, Copyright, and Other Notices:**
All rights reserved by our team.
- **Applicable Standards:**
The system must comply with relevant data protection laws, such as GDPR, to ensure privacy and secure handling of user data. The recommendation algorithms should adhere to industry standards
for fairness, transparency, and inclusivity, minimizing biases in recommendations.
- **Hardware and Software Requirements:**
  - **i.** **Hardware**: The system should support high-performance servers for handling large-scale user data and real-time recommendations. Depending on the deployment, cloud services may be used for scaling.
  - **ii.** **Software**: The recommendation engine should be built on robust frameworks

(e.g., TensorFlow, PyTorch, Scikit-learn) and deployed using scalable cloud platforms e.g., AWS, Google Cloud, Azure).

- **Ethical and Social Requirements:**
  The recommendation system should be designed to prioritize user privacy and data security.
- **Training Requirement:**
  Staff and relevant stakeholders should be trained on how to interpret the recommendations, manage the system, and handle user queries.
- **Support and Maintenance Requirement:**
  Ongoing technical support should be provided to ensure system reliability, including regular updates, bug fixes, and improvements to the recommendation algorithms based on user feedback and new data.