The Islamia University of Bahawalpur

Department of Software Engineering

SOFTWARE REQUIREMENTS SPECIFICATION

(SRS DOCUMENT)

for

Multi-Vendor E-Commerce Web App with LLM Integration

Version 1.0

By

Haider Ali

S22BSEEN1E02078

Session Spring 2022 – 2026

Supervisor

Ms. Alisha Fida

Bachelor of Software Engineering

Table of Contents

Revision History

Name	Date	Reason for changes	Version

Application Evaluation History

Comments (by committee) ***include the ones given at scope time both in doc and presentation**	**Action Taken**

Supervised by

Ms. Alisha Fida

Signature_______________

Introduction

Time has definitely witnessed the rise of digital commerce and with every passing day more and more is being popularized as companies as well as customers are all influenced. An E-Commerce Web Application is simply where customers can shop, use to acquire products, complete purchase order, and track orders online. In this paper, we identify the specifications needed to build such an application in Django Framework with sound programming principles and for securing and scalability. It has two sections the customer interface and the admin interface. Customers will be able to view an online catalogue, add items to the shopping cart, pay via secure payment gateways. Amongst other things, owners and administrators will be able to manage products and orders with an easy-to-use administration panel so they can also control user accounts. Features such as product search, order tracking, notifications will assist in avoiding the creation of complex problems related to the shopping process being introduced in the system. The platform will be designed today using current technologies and will be fast, secure and will be available to users on all types of devices. This SRS specifies exactly how to build the system according to requirements, and in turn is based on them.

Purpose

This document is written in order to provide software requirements of an E-Commerce Web Application. With this system a user is able to browse through products, put products in the cart, make secure payments and also manage the order. Django is a Python based web framework which will be used for developing this system to ensure scalability, security as well as ease of user-friendly interaction.

Scope

This Multi-Vendor E-Commerce web application will provide the following features:

- Separate registration and dashboard for Customers and Vendors.
- Vendor registration approval by Admin.
- Vendors can add/manage their own products, view orders, and monitor earnings.
- Customers can browse, filter, and purchase from a shared product catalog.
- Secure payments processed per vendor (split payment architecture via Stripe/PayPal).
- Admin can manage all users, products, orders, and commissions.
- Full LLM-based chatbot powered by GPT for personalized support and product recommendations.
- Mobile responsive and scalable for high user traffic.

Product Perspective

This application serves as a multi-vendor marketplace where:

- Multiple vendors can sell products via their vendor-specific dashboards.
- Customers can purchase products from one or more vendors in a single order.
- Admin manages users, vendors, commissions, and platform-wide policies.
- AI-powered chatbot (based on LLM like GPT) assists users in real-time for queries, FAQs, and smart product suggestions.

Operating Environment

- Frontend: HTML, CSS, JavaScript (Bootstrap for responsiveness).
- Backend: Django Framework (Python).
- AI chatbot: LLM
- Database: PostgreSQL/MySQL.
- Deployment: Cloud-based (AWS or Digital Ocean).
- Payment Integration: Stripe, PayPal,etc
- Browser Compatibility: Google Chrome, Mozilla Firefox, Safari, Edge.

Design & Implementation Constraints

- System must be developed using the Django Framework.
- Frontend must be mobile responsive (Bootstrap or similar).
- Must support multiple browsers (Chrome, Firefox, Safari, Edge).
- Integrate with PCI-DSS compliant payment gateways (e.g., Stripe, PayPal).
- Ensure secure user data handling with SSL/TLS and hashed passwords.
- Role-based access control for customers, vendors, and admins.
- Stable internet connection required for seamless functionality.
- Payment APIs and email services must be functional for transactions and notifications.

Overall Description

User Classes and Characteristics

- Customers: Browse and purchase products, manage profiles, chat with AI assistant.
- Vendors: Register as sellers, list products, view/manage orders, withdraw earnings.
- Admins: Approve vendors, manage products/orders/users, view platform analytics.
- AI Assistant (LLM): Responds to user queries, recommends products, handles FAQs, escalates to human if needed.

Customers

- Browse products from multiple vendors and add them to a shared shopping cart.
- Place orders containing products from one or more vendors.
- View and manage profile details including shipping address and payment history.
- Track order status (Pending, Shipped, Delivered) per item/vendor.
- Submit product ratings and written feedback after purchase.
- Get smart assistance from the LLM-powered AI chatbot (e.g., product recommendations, return policy).

Vendors

- Register and wait for admin approval before becoming active sellers.
- Manage their own product listings (add/update/delete).
- View and manage orders related to their products.
- Track earnings, request withdrawals, and view commission deductions.
- Update store profile (e.g., contact details, store logo).
- Receive notifications about new orders and low stock alerts.
- Receive customer feedback and respond to product queries (optional).

Administrators

- Manage all platform users: customers, vendors, and guests.
- Approve/reject vendor registration requests.
- Moderate and manage the entire product catalog across vendors.
- Process vendor withdrawal requests and monitor commission reports.
- Manage chatbot content, escalation triggers, and logs.
- Oversee the system’s performance, backups, and maintenance activities.

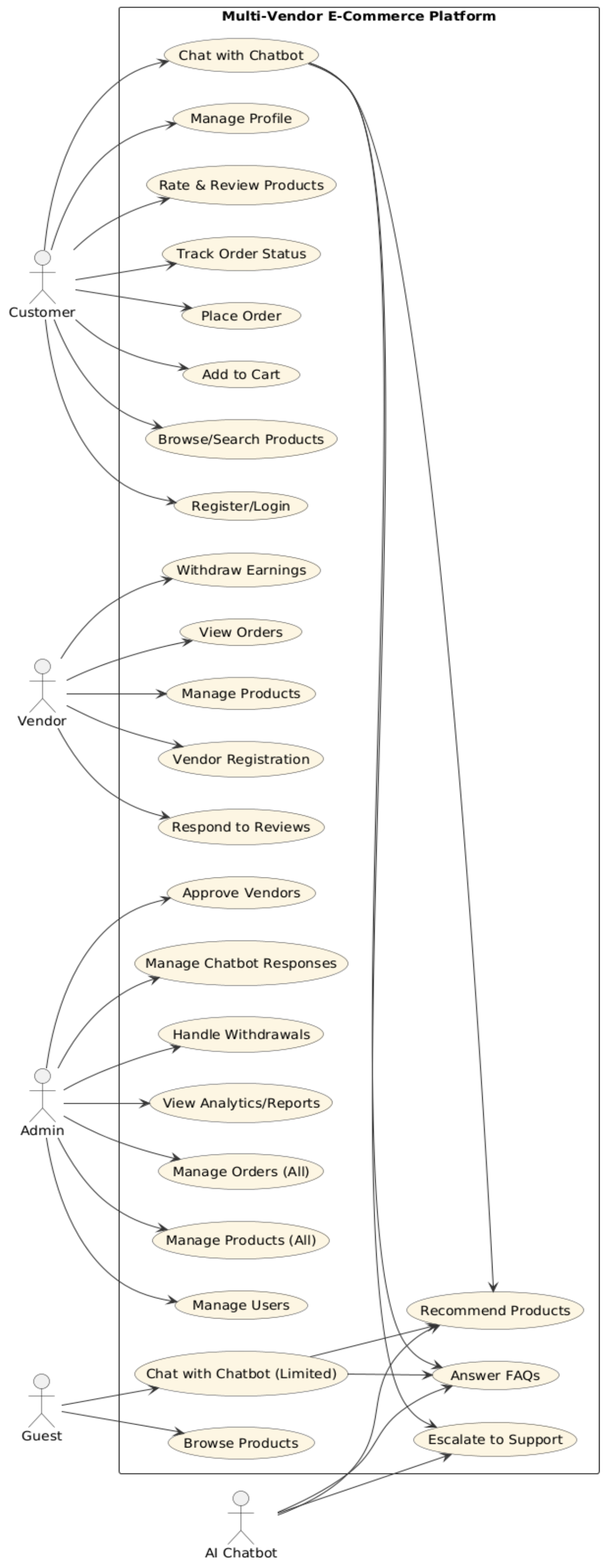
Guest Users

- Browse the full product catalog across vendors.
- Use filters, search, and categories to discover products.
- Interact with the AI chatbot for basic queries and product recommendations.
- Must sign up to place orders or access personal features.

AI Chatbot (LLM-Powered)

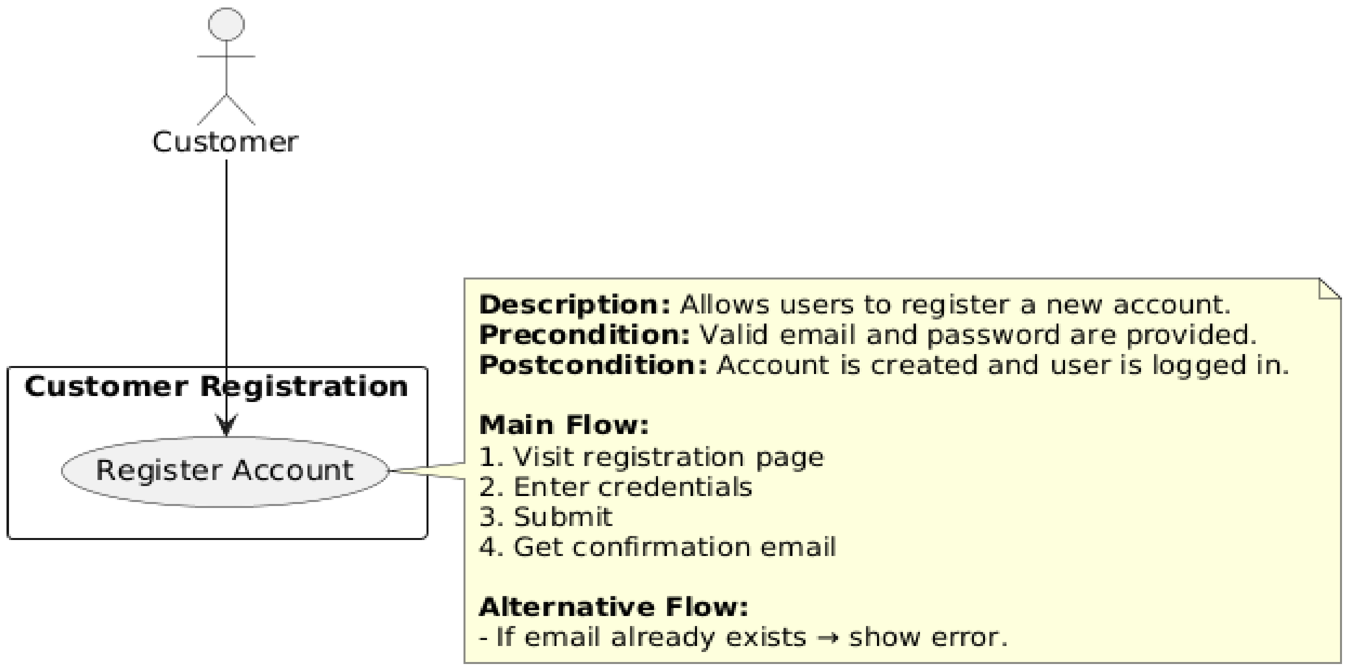
- Responds to natural language queries such as:
    - "What’s the return policy for Vendor X?"
    - "Recommend a budget laptop under 100,000 PKR."
- Helps users locate specific products or categories.
- Offers personalized suggestions based on browsing history or preferences.
- Escalates complex issues to human support (e.g., order disputes).
- Accessible to both guests and registered users across all devices.

Use Case Diagram



Use Case 1: Customer Registration

Use Case	**Customer Registration**
Actor	**Customer**
Description	Allows users to register a new account.
Precondition	User provides valid email and password.
Postcondition	Account is created and user is logged in.
Main Flow	1. Visit registration page 2. Enter credentials 3. Submit 4. Get confirmation email
Alternative Flows	Email already exists → show error.



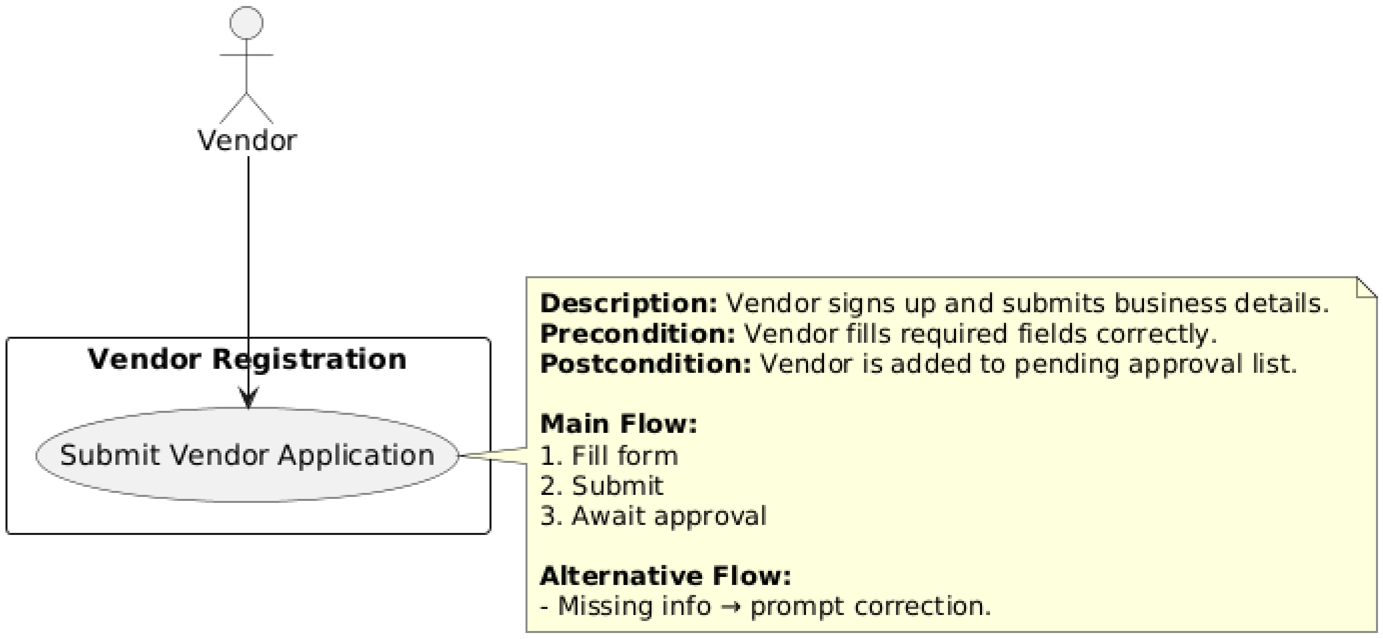
Description: Allows users to register a new account.
Precondition: Valid email and password are provided.
Postcondition: Account is created and user is logged in.

Main Flow:
1. Visit registration page
2. Enter credentials
3. Submit
4. Get confirmation email

Alternative Flow:
- If email already exists → show error.

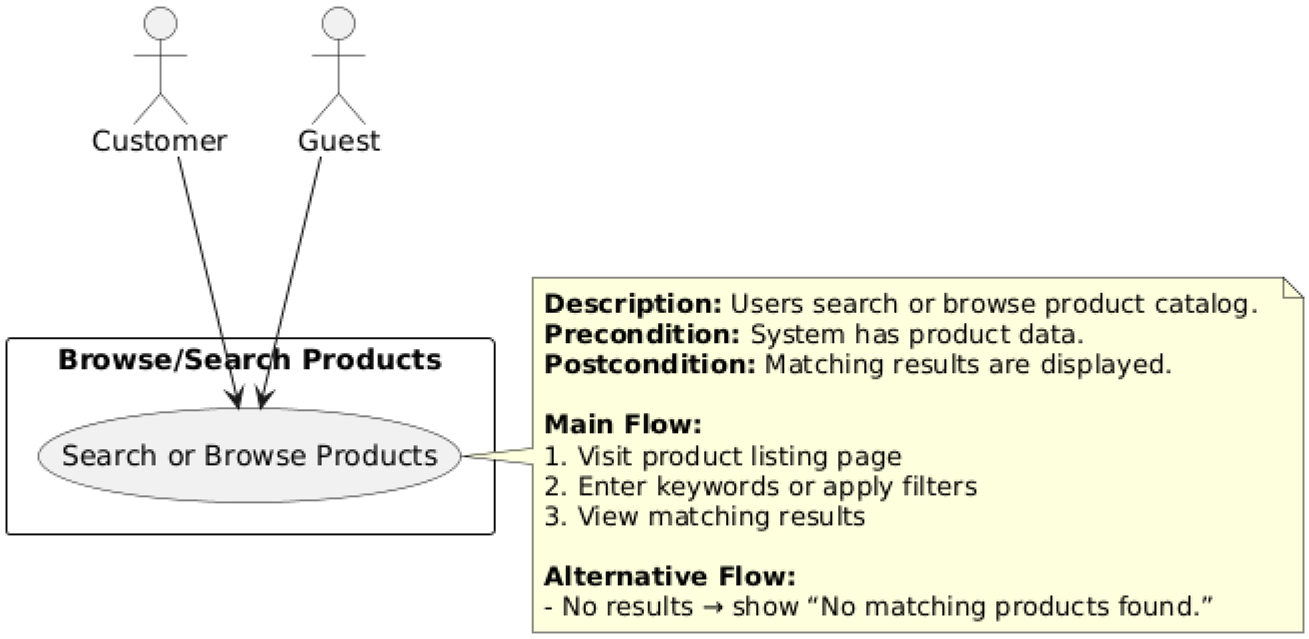
Use Case 2: Vendor Registration

Use Case	Vendor Registration
Actor	**Vendor**
Description	Vendor signs up and submits business details.
Precondition	Vendor fills required fields correctly.
Postcondition	Vendor is added to pending approval list.
Main Flow	1. Fill form 2. Submit 3. Await approval
Alternative Flows	Missing info → prompt correction.


Vendor
Vendor Registration
Submit Vendor Application
Description: Vendor signs up and submits business details.
Precondition: Vendor fills required fields correctly.
Postcondition: Vendor is added to pending approval list.
Main Flow:
1. Fill form
2. Submit
3. Await approval
Alternative Flow:
- Missing info → prompt correction.

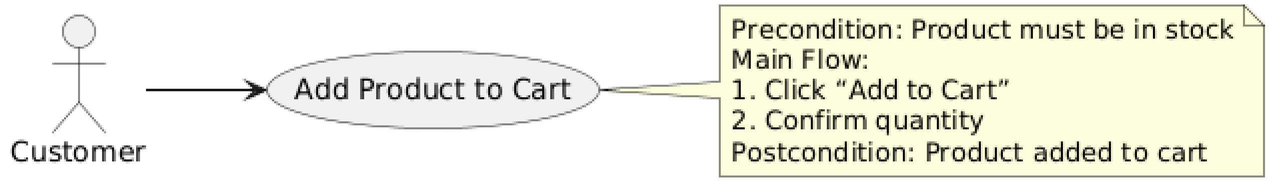
Use Case 3: Browse/Search Products

Use Case	**Browse/Search Products**
Actor	**Customer, Guest**
Description	Users search or browse product catalog.
Precondition	System has product data.
Postcondition	Matching results are displayed.
Main Flow	1. Type in search or apply filter 2. Results appear


Customer
Guest
Browse/Search Products
Search or Browse Products
Description: Users search or browse product catalog.
Precondition: System has product data.
Postcondition: Matching results are displayed.
Main Flow:
1. Visit product listing page
2. Enter keywords or apply filters
3. View matching results
Alternative Flow:
- No results → show "No matching products found."

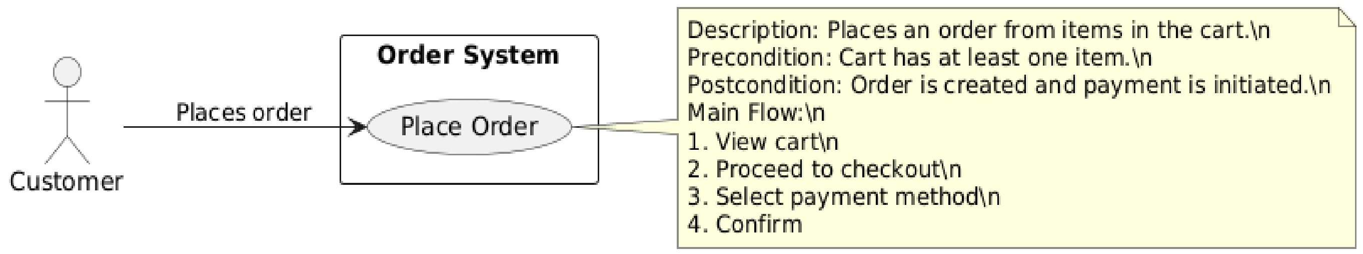
Use Case 4: Add to Cart

Use Case	**Add Product to Cart**
Actor	**Customer**
Description	Adds selected product to shopping cart.
Precondition	Product must be in stock.
Postcondition	Product added to cart.
Main Flow	1. Click “Add to Cart” 2. Confirm quantity



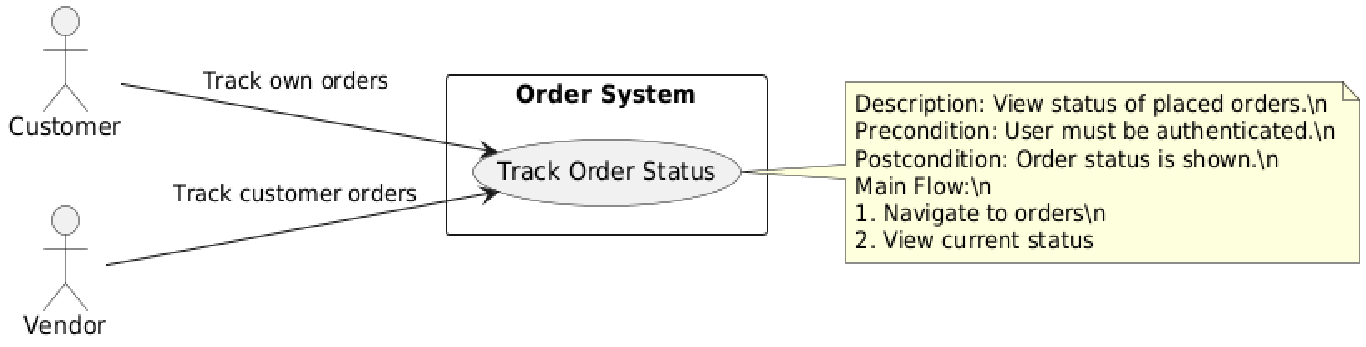
Use Case 5: Place Order

Use Case	**Place Order**
Actor	**Customer**
Description	Places an order from items in the cart.
Precondition	Cart has at least one item.
Postcondition	Order is created and payment is initiated.
Main Flow	1. View cart 2. Proceed to checkout 3. Select payment method 4. Confirm


Customer
Places order
Order System
Place Order
Description: Places an order from items in the cart.\n
Precondition: Cart has at least one item.\n
Postcondition: Order is created and payment is initiated.\n
Main Flow:\n
1. View cart\n
2. Proceed to checkout\n
3. Select payment method\n
4. Confirm

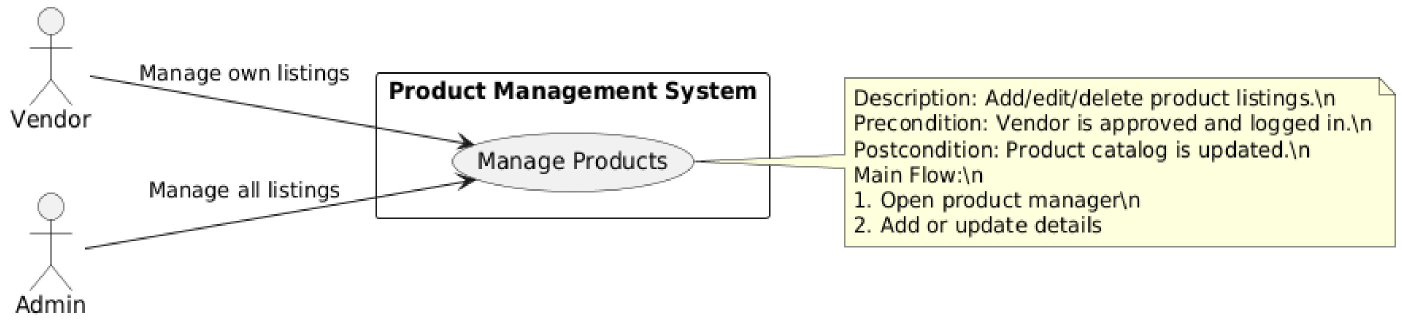
Use Case 6: Track Order

Use Case	Track Order Status
Actor	**Customer, Vendor**
Description	View status of placed orders.
Precondition	User must be authenticated.
Postcondition	Order status is shown.
Main Flow	1. Navigate to orders 2. View current status



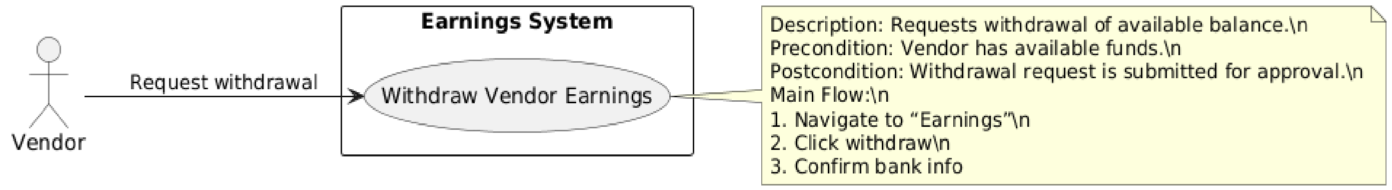
Use Case 7: Manage Products

Use Case	**Manage Products**
Actor	**Vendor, Admin**
Description	Add/edit/delete product listings.
Precondition	Vendor is approved and logged in.
Postcondition	Product catalog is updated.
Main Flow	1. Open product manager 2. Add or update details


Vendor
Manage own listings
Admin
Manage all listings
Product Management System
Manage Products
Description: Add/edit/delete product listings.\n
Precondition: Vendor is approved and logged in.\n
Postcondition: Product catalog is updated.\n
Main Flow:\n
1. Open product manager\n
2. Add or update details

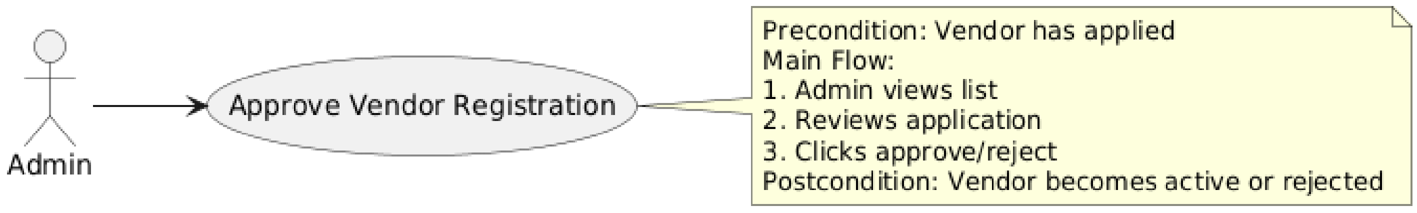
Use Case 8: Withdraw Earnings

Use Case	**Withdraw Vendor Earnings**
Actor	**Vendor**
Description	Requests withdrawal of available balance.
Precondition	Vendor has available funds.
Postcondition	Withdrawal request is submitted for approval.
Main Flow	1. Navigate to “Earnings” 2. Click withdraw 3. Confirm bank info


Earnings System
Request withdrawal
Withdraw Vendor Earnings
Vendor
Description: Requests withdrawal of available balance.\n
Precondition: Vendor has available funds.\n
Postcondition: Withdrawal request is submitted for approval.\n
Main Flow:\n
1. Navigate to “Earnings”\n
2. Click withdraw\n
3. Confirm bank info

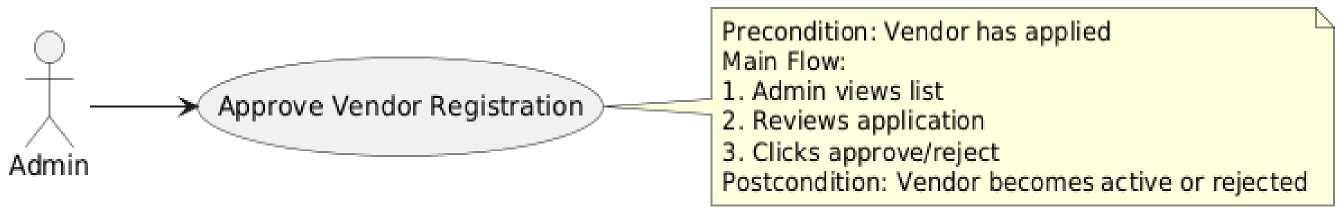
Use Case 9: Approve Vendor

Use Case	Approve Vendor Registration
Actor	Admin
Description	Approves or rejects vendor applications.
Precondition	Vendor has applied.
Postcondition	Vendor becomes active or rejected.
Main Flow	1. Admin views list 2. Reviews application 3. Clicks approve/reject


Admin
Approve Vendor Registration
Precondition: Vendor has applied
Main Flow:
1. Admin views list
2. Reviews application
3. Clicks approve/reject
Postcondition: Vendor becomes active or rejected

Use Case 10: Chat with Chatbot

Use Case	**Chat with AI Chatbot**
Actor	**Customer, Guest**
Description	Interact with chatbot for assistance.
Precondition	Chatbot is available and trained.
Postcondition	User receives help or escalates to support.
Main Flow	1. Click chat icon 2. Ask question 3. Get intelligent response


Admin
Approve Vendor Registration
Precondition: Vendor has applied
Main Flow:
1. Admin views list
2. Reviews application
3. Clicks approve/reject
Postcondition: Vendor becomes active or rejected

Functional Requirements

Identifier	**FR-1**
Title	Register as Vendor
Requirement	The system shall allow users to register as vendors by submitting necessary business and contact information.
Source	UC-Register Vendor
Rationale	Enables onboarding of new sellers to the platform.
Business Rule	BR-1: Vendor applications must be approved by admin.
Dependencies	None
Priority	High

Identifier	**FR-2**
Title	List Product
Requirement	Approved vendors shall be able to add new product listings with images, descriptions, prices, and stock.
Source	UC-ManageProducts
Rationale	Supports catalog growth and product availability.
Business Rule	BR-2: Products must be reviewed for compliance before being published.
Dependencies	FR-1
Priority	High

Identifier	FR-3
Title	Place Order
Requirement	Customers shall be able to place orders for items added to their shopping cart.
Source	UC-PlaceOrder
Rationale	Core functionality to enable purchases.
Business Rule	BR-3: Orders must have at least one valid item and a confirmed payment method.
Dependencies	FR-2
Priority	High

Identifier	FR-4
Title	Track Order Status
Requirement	Customers and vendors shall be able to view the current status of orders.
Source	UC-TrackOrderStatus
Rationale	Improves transparency and user experience.
Business Rule	BR-4: Only authenticated users can access order history.
Dependencies	FR-3
Priority	Medium

Identifier	**FR-5**
Title	Manage Product Listings
Requirement	Vendors and admins shall be able to add, edit, or delete product listings.
Source	UC-ManageProducts
Rationale	Keeps the catalog accurate and up to date.
Business Rule	BR-5: Only approved vendors or admins can modify listings.
Dependencies	FR-1
Priority	High

Identifier	**FR-6**
Title	Withdraw Vendor Earnings
Requirement	Vendors shall be able to request withdrawal of their available earnings.
Source	UC-WithdrawVendorEarnings
Rationale	Supports vendor revenue management.
Business Rule	BR-6: Withdrawals require verified bank details and admin approval.
Dependencies	FR-3
Priority	Medium

Identifier	FR-7
Title	AI Chatbot Support
Requirement	The system shall provide a chatbot interface to assist users with questions or issues.
Source	UC-ChatWithAIChatbot
Rationale	Improves support accessibility and reduces response time.
Business Rule	BR-7: Chatbot must provide option to escalate to live support.
Dependencies	None
Priority	Medium

1.User Registration, User Authentication

Users must be able to register using their email and password. The system must also support secure log in and log out. They must also be able to reset their password through email. The access to managing the system is role based.

2. Product Management

Adding, updating, and deleting products have to be possible for admins. Name, price, description, category, stock, images are the details we need to include in an item, and products should have those details. Users can search and filter products by names, by category and price. It should be displayed the product ratings and reviews.

3. Shopping Cart

We need to'think of product addition to cart or product removal from cart' as a function for the users. The product quantity in the cart must be updateable by users. The total price must be dynamically displayed on the cart.

4. Order Management

After payment confirmation users should be able to place orders. The business needs to generate order receipts and send them via email. Order status users should be able to track (Pending, Shipped, Delivered). Admins must be able to change order status.

5. Payment Integration

The system has to work with payment gateways (such as Stripe, PayPal). The payment details must be processed securely and securely stored. Payment and transaction receipts must be sent to users.

6. Notifications

The system must send email notifications for:

Order confirmation.

Payment receipts.

Order status updates. New orders should notify admins.

7. User Profile Management

Users must be able to change their personal details (name, address, contact info), 7.2 Users must be able to view their order history.

8. Feedback and Ratings

Users are to rate and give feedback about product, 8.2 Ratings should be visible on the product page.

Non-Functional Requirements

1.Performance Requirements

It has to respond to the user within 3 to 5 seconds. It must be handled by the system of 1000+ concurrent users without performance degradation.

2.Security Requirements

All sensitive data has and must be encrypted using SSL/TLS. Password are hashed to and stored securely. In reality, PCI-DSS standards must be adhered to by payment transactions. To implement role based access control, the system must be such.

3.Usability Requirements

For all devices the system needs to have an intuitive and responsive design. It must also support multiple languages (optional).

4. Availability Requirements

The system needs to have 99.9% uptime. Data loss is to be prevented by doing regular backups.

5. Maintainability Requirements

And, to permit easy updates and maintenance, the system must be modular. Developers and their admins need to have comprehensive documentation.

(

1. References

Django Official Documentation: https://docs.djangoproject.com/

Stripe API Documentation: https://stripe.com/docs