Software Requirements Specification

for

Bazar Kori – An Online Grocery Retail and Delivery Platform

Must. Yasmin Sultana Emu, Sadia Reza, Talha Muammar,
Fabliha Anbar Riadh, Fariha Tasnim Pragga

October 2025

Contents

Chapter 1

Introduction

1.1 Purpose

“Bazar Kori” is a unified online grocery purchasing solution designed to simplify and enhance the way users shop for everyday goods and essentials. Being built using FastAPI, the platform offers a modern, efficient, and scalable architecture that enhances the performance and reliability for all users.

The system enables customers to browse, compare, and purchase grocery products from a wide range of categories; from freshly produced packages of goods to all forms of convenience, all can be achieved from their home with a few taps and scrolls on their phone in our app. Users can easily add to their cart, manage orders, and track every delivery they make in real time. To accommodate various user needs, “Bazar Kori” caters to both casual buyers and premium subscribers, offering personalized recommendations, priority delivery, and exclusive discounts for premium users, making a user-friendly, seamless, and secure experience.

This Software Requirements Specification (SRS) defines the functional and non-functional requirements needed for “Bazar Kori” to achieve its goal of providing a comprehensive and efficient online grocery shopping experience.

1.2 Target Audience

Software Requirements Specification (SRS) is done to meet a wide variety of stakeholders involved in the development, operation, and use of our system. The groups are significant to the platform to ensure it meets the functional, technical, and business goals. The key stakeholders are:

- Administrators: They will oversee the system, its performance, and ensure a secure and smooth running.
- Project Managers (PMs): These individuals monitor the general development process, coordinate among different teams, and make sure that

project objectives are achieved within the limits of the scope and time.

- Developers: Participated in the design, coding, and implementation of the core functionality of the platform in Python, FastAPI, and others.
- Testers and Quality/Quality Control Engineers: Test the system to ensure that it performs to its specifications by performing testing, debugging, and quality assurance.
- Clients and Investors: Assess the viability of the project, its scalability, and business opportunity, as well as make sure that it is consistent with commercial objectives.
- End Users (Customers): Use the system to browse, buy, and administer grocery products easily.
- Sellers/Administrators: Handle the listing of the products, stocks, and fulfillment of the orders with the aid of the seller interface.
- Delivery Agents: Receive, handle assigned deliveries, update the status of orders, and finalize delivery operations on time.
- External Stakeholders (e.g., Payment Gateway Providers, Logistics Partners): Provide third-party integrations and provide the Grocery Market ecosystem with seamless interoperability.

Conforming to the content of the document with the special requirements of each group of audience members, this SRS strives to create a common ground among all stakeholders, which will lead to efficient cooperation throughout the entire development and deployment life of the system.

1.3 Application Use Context

This section displays the relationship of various stakeholders with the Grocery Market system. All roles are important to the lifecycle of the system, both during the development and management phases, up to the operation of the platform by the end-users, and are therefore a guarantee that the platform provides a smooth, safe, and effective experience of maintaining the grocery store operations.

1.3.1 System Administrators

The entire system is operated by administrators who handle all matters such as user accounts, product listings, roles, and managing the database. They also act as the sellers where all the sales, inventory, and pricing are done via the admin dashboard. It has two types of users and does not have a single seller log-in: Admin and Customer.

1.3.2 Business Analysts (BAs)

Essentially, those analytics dashboards allow BAs to explore sales data, customer behavior, and market trends. They take the feedback of the users and propose improvements to ensure that the system remains competitive and the users remain satisfied.

1.3.3 Project Managers (PMs)

PMs are the equivalents of the project directors since they ensure that the dev squad remains in sync with requirements and what the users really want. They liaise with clients, devs, and stakeholders and ensure that deadlines and deliverables are tight.

1.3.4 Developers

Dev folks write code and keep the system operational, primarily Python FastAPI, and connect it to services such as payments and tracking of deliveries. They work in pairs, practice, and monitor deployments to ensure that everything is firm.

1.3.5 QA/QC Engineers

QA/QC personnel will run functional tests, performance tests, and security tests to ensure that things are dependable. They record findings and bugs in the system to ensure quality rolling continuously and uniformly across the board.

1.3.6 End Users (Buyers and Admins/Sellers)

Buyers and admins as end users can register, shop, manage their carts, pay, save addresses, as well as leave a review. High-value customers receive shipping at no cost through a subscription. Admins (sellers) deal with the listings, the pricing, stock, checkout analytics, and purchase history.

1.3.7 Key Stakeholders (Leadership, Sales, Marketing)

The app is used by leadership, sales, and marketing teams to receive sales reports, customer engagement statistics, and trend data. They extract info created by the system to guide strategy, review marketing, and measure platform growth. Although we do not connect the outside marketing tools, the data provides important information on the business planning.

1.3.8 Test Engineers

Test engineers, in collaboration with QA/CQ, are important in ensuring that the app is capable of withstanding real-life loads. They do load/stress/end-to-end tests just to be certain that it can sustain the various user demands until we deploy it.

1.3.9 Investors / Sponsors

Funding and monitoring the progress of the “Bazar Kori” project is done by investors and sponsors. They receive frequent finance, usage, feature updates, and ROI reports. Although they do not strike a live dashboard, they are provided with comprehensive summaries with which to monitor the progress and identify opportunities. They are also able to check on deliverables and provide feedback on the same to ensure that we achieve business objectives.

1.4 Product Scope

“Bazar Kori” is basically a unified online grocery shopping system where customers can find all daily essentials under one store. This platform aims to provide a seamless and reliable shopping experience for both the customers and administrators. This section will provide an overview of the system’s goals, boundaries, and core functionalities.

1.4.1 Objective Overview

The system aims to simplify and digitalize our daily shopping experience by providing a smooth, easy interface to navigate through and place orders for the customers. Customers will be able to browse products through categories, manage carts, place orders, and secure payments through a third-party gateway. The administrator will be able to manage inventory, sort orders, support customers through direct messaging, and monitor sales all through a centralized dashboard. The system is developed using FastAPI to ensure scalability, responsiveness, and performance efficiency.

1.4.2 Expected Benefits and Key Goals

This section outlines how users and the business will benefit from our system. The key goal is to establish a dependable and scalable grocery solution that ensures user satisfaction and operational efficiency. Key benefits include:

- Convenient and time-saving online shopping experience.
- Easy categorized product browsing, search, and filtering options.
- Secure payment and order tracking functionalities.
- Secure user authentication.
- Organized inventory for efficient operation.
- Reduction of manual errors.

- Enhanced customer service with feedback.
- Data-driven insights for better business decisions.

1.4.3 Alignment with Corporate Vision

The app supports organizational or commercial goals by encouraging a broader vision in the digital retail and grocery market. With the help of technology, businesses will be able to create a sustainable and customer-centric model that will be both easily accessible and convenient. It also promotes the e-commerce system among the local vendors, contributing to the modern economy.

1.4.4 Integration with Business Model

"Bazar Kori" is such an application that will help retail businesses expand by providing a seamless online shopping and delivery system. It connects broader business operations by automating inventory management, order tracking, and payment processing. While the customer can place orders without store visits, the sellers can list their products and manage them directly through the platform.

1.5 Identified Risks

This section describes the possible challenges or risks in project implementation, deployment, and operational performance.

1.5.1 Low User Engagement

There might be a risk of users not being active on the platform, due to unfamiliarity and competition with other established grocery stores, affecting the overall success of the application.

1.5.2 Administrative Overload

Administrators might face a potentially high workload due to the manual maintenance of all the work, including user issues, data management, inventory updates, and terms of conduct.

1.5.3 Communication Gaps

There might be communication gaps among the administrators and the customers while handling feedback and complaints. There is a risk that the stakeholders can also misinterpret the SRS and fail to understand the actual goal and functionalities of the platform. This lack of communication between the developers, vendors, and stakeholders can delay the implementation.

1.5.4 Lack of User Feedback

Limited reference or feedback from early users can hinder the improvement and refinement of the system, which can create hindrance among the users about their expectations and the delivery/shopping system.

1.5.5 Scope Drift

Changes or expansions beyond the initial project scope, addition of more and more new features, may hinder the development of the project and its completion. It can also misguide the developers, project managers, and testers.

1.5.6 Evolving Stakeholder Demands

There can be risks that the application might be disrupted while adjusting as per the requirements of the business as it evolves. It can also affect the timeline or the structure of the project while updating.

1.5.7 Data Security Threats

Vulnerabilities in system or user data protection may arise, threatening the overall integrity of the system. There might be risks of unauthorized access, data leakage, financial transactions, and cyber threats compromising sensitive user information.

Chapter 2

Description

2.1 User Classes and Characteristics

Our grocery shopping system consists of two major classes: the buyer class and the admin class.

2.1.1 User Class: Buyers

Characteristics:

- Buyers can create an account, log in, and manage personal information from their end.
- Can view product details, including images, descriptions, prices, and reviews.
- Can browse, search, and filter across various categories.
- Add products to their carts and modify quantities before checkout.
- Can place orders using secure payment gateways and receive order confirmation messages and invoices via mail.
- Will be able to view order history, provide product reviews, and reorder from that list.

2.1.2 User Class: Administrator

Characteristics:

- Will be able to log in, but not be able to create an account.
- Can add, update, and delete any product.
- Edit product images, description, and pricing.

- Manage inventory after every successful delivery.
- Receive and review customer orders.
- Assign delivery agents and track delivery status.
- Update order status (e.g., Pending, Shipped, Delivered, Canceled, etc.) for the customer to know the current progress.
- Manage reviews and feedback from buyers.
- Respond to buyers' queries and complaints.
- Have full access to all classes for monitoring and maintenance.

2.2 User Needs

2.2.1 Administrator

- **Product Listing and Management:** The sellers need a user-friendly interface to easily list their products, update the product information, and delete them. Features for uploading product images and details will also be available.
- **Order Management:** Sellers will be able to manage customer orders — canceling orders and updating inventory according to orders.
- **Communication:** There will be a messaging option between sellers and buyers to ensure clarity about the products.
- **Track Ratings:** Sellers can track the ratings from buyers and evaluate the product according to that.
- **Manage Buyers:** As the administrator and seller are the same, they will also be able to manage buyers' accounts. There will be a different dashboard for checking sales and banning fraudulent customers.
- **Responsiveness:** The web app should be responsive to any kind of device that users may use.

2.2.2 Buyers

- **Explore and Search:** Buyers should have a simple yet effective user interface to browse for products. They will be able to see products of different categories, search for a specific product by name, filter the product list for various characteristics, and view the product details.

- **Order Placement:** Buyers can choose their desired products from the list and add them to a cart. They can check out the cart items immediately or anytime later. Buyers should be able to order multiple products of multiple quantities in the same shipment, select a preferred delivery time, and a preferred payment method. Within a certain time period of ordering, a buyer can update or cancel their order.

- **Communication:** Buyers can communicate with the sellers in order to clarify any doubts about the product or to learn more details.

- **Provide Feedback:** After delivery, a buyer can give ratings and feedback on the product and delivery system.

- **Responsiveness:** The web app should be responsive to any kind of device that users may use.

2.2.3 General User Needs

- **User Authentication:** A secure authentication process is crucial to ensure all users can access and use their accounts securely.

- **Accessibility:** The system should be accessible to all users using different devices and operating systems. The system user interface should be designed in such a way that every user can easily navigate through the web application.

- **Notification:** Users should receive timely notifications about order status, messages, and other relevant information to stay updated.

2.3 Operating Environment

2.3.1 Hardware Platform

- Desktops: Intel Core i5 processor or equivalent, 16GB RAM, 512GB SSD or higher

- Laptops: Intel Core i5 processor or equivalent, 8GB RAM, 256GB SSD or higher

- Mobile Devices:
  - iOS devices (iPhone X and above)
  - Android devices (Android 12.0 and above)

- Tablets: iPad (iOS 14+) and Android tablets (12.0+)

2.3.2 Operating Systems and Versions

- Windows: 10 (64-bit), 11 (64-bit)
- macOS: Monterey (12.0) and above
- Linux: Ubuntu 20.04 LTS and above, Arch Linux, Fedora 35 and above
- Mobile: iOS 14+, Android 12.0+

2.3.3 Software Components and Applications

- Web browsers:
  - Google Chrome (latest stable version)
  - Brave (latest stable version)
  - Mozilla Firefox (latest stable version)
  - Microsoft Edge (latest stable version)
  - Safari (latest stable version)
- Mobile applications for iOS and Android
- Payment gateway SDKs
- Inventory and order management software
- Push notification services

2.3.4 Database Compatibility

- MySQL 8.4 or higher
- PostgreSQL 15 or higher
- MongoDB 6.0 or higher
- Redis for caching and session management

2.3.5 Integration and Interoperability

- RESTful APIs for internal and external services
- JSON data interchange format
- Support for third-party systems

2.3.6 Network Requirements

- Minimum 5 Mbps internet speed
- Support for Ethernet and Wi-Fi connections
- SSL/TLS for secure data transmission

2.3.7 Security Considerations

- TLS/SSL encryption for all data transmissions
- Compatibility with industry-standard firewalls
- Secure authentication and authorization mechanisms
- Regular security updates and patches

2.3.8 Cloud and Server Environment

- Cloud-hosted servers (AWS, Google Cloud, or Azure) for scalability
- Minimum server specifications:
    - 4-core CPU, 16GB RAM, 500GB SSD
    - Load balancing enabled for high traffic periods
- Support for containerization (Docker/Kubernetes)
- Auto-scaling capabilities based on traffic
- Secure server environment with firewalls
- Logging and monitoring for server performance and uptime
- Maintenance windows scheduled during off-peak hours

2.4 Constraints

2.4.1 Technical Constraints

- Selection of frameworks and technologies for secure payment processing and user authentication
- Ensuring scalability to support increasing numbers of users, products, and orders
- Integration limitations with third-party delivery or inventory systems
- Compatibility with a wide range of devices and operating systems

2.4.2 Time Constraints

- Development phases with deadlines for features such as catalog management, checkout, and delivery tracking
- Testing periods for performance, security, and cross-platform compatibility
- Launch planning based on seasonal demand or marketing campaigns

2.4.3 Budget Constraints

- Allocation of funds for cloud hosting, database services, and server infrastructure
- Costs associated with payment gateway fees and third-party API subscriptions
- Budgeting for mobile app development and maintenance
- Optional marketing and promotional expenses

2.4.4 Regulatory and Compliance Constraints

- Compliance with local e-commerce and data protection laws
- Compliance with consumer protection laws, including return/refund policies
- Age restrictions for certain products or restricted items

2.4.5 Resource Constraints

- Availability of skilled developers with expertise in web/mobile development, database management, and cloud infrastructure
- Limited QA/testing personnel for thorough multi-platform testing
- Human resources for customer support and order fulfillment

2.4.6 Operational Constraints

- Dependence on reliable internet connectivity for real-time order updates
- Delivery logistics limitations in certain geographic areas
- Stock and inventory restrictions based on supplier availability

2.4.7 Security Constraints

- System must maintain secure handling of sensitive user data, including payment and personal information
- Role-based access control to prevent unauthorized actions
- Encryption standards must meet industry best practices

2.5 Assumptions

The assumptions for our system are as follows:

- The system will be used over a stable internet connection for seamless interaction.
- Users, both buyers and sellers/admins, will have valid accounts to log in.
- All users have basic literacy in computers and smartphones.
- Users are assumed to have a basic understanding of how to use an online platform.
- The administrators have the skills and knowledge to manage the entire system.
- Data entered by the user is valid and correctly formatted, including name, address, prices, etc.
- The payment gateway (cash on delivery) is functional and secure.
- Buyers have valid phone numbers and mailing addresses for notifications and invoices.
- The system assumes no simultaneous conflicting edits.
- The backend database maintains data integrity.

Chapter 3

Requirements

3.1 Functional Requirements

3.1.1 Buyer Authentication and Account Management

As a buyer, I want to be able to create a new account if I do not have any existing accounts. While creating an account, I want the system to take my name, phone number, and email address. The system must validate the email address format and enforce password rules. The system shall allow login with email and password. Also, I want to update my personal details such as name, phone number, address, etc.

Success

- The buyer enters all the required fields correctly.
- The system validates the format and password strength successfully.
- A new buyer account is created and stored in the database.
- The system shows a success message: "Account created successfully!".
- For login, after entering the email address and password, the system verifies the credentials successfully.
- Upon the buyer updating the profile, the system saves changes to the database and shows a message such as "Profile updated successfully!".

Failure

- *While creating the account:*
    - ○ One or more required fields are left blank.
    - ○ Invalid email format → "Enter a valid email address."

  - Password does not meet requirements → “Password must contain at least 8 characters and a special symbol.”
  - Email already exists in the database → “Email already registered. Please log in.”

- *While logging in:*
  - Invalid email or password → “Incorrect email or password.”
  - Empty fields → “Email and password are required.”

3.1.2 System Administrator Login

As a seller/admin, I want to be able to log in to the system using credentials provided by the authority, which are stored in the database, so that I can securely access my admin dashboard and manage store operations.

Success

- Seller enters a valid email/username and password provided by the system authority.
- System verifies credentials from the database successfully and displays “Login successful. Welcome back!”.

Failure

- Seller enters incorrect credentials → “Invalid username or password.”
- If the entered credentials do not exist in the database → “Unauthorized access. Contact the system administrator.”

3.1.3 Product Listing

As a seller/admin, I want to add new product details such as name, category, description, price, quantity, and images, and upload them to the system so that the products appear in the store and can be searched and viewed by buyers.

Success

- The seller/admin fills in all the required fields to post a product for sale.
- Product data is successfully uploaded and stored in the database.
- The uploaded image is processed and linked to the corresponding product.
- The system displays a confirmation message such as “Product added successfully.”

Failure

- If required fields are missing, the message “Please fill in all mandatory fields.” is displayed.

3.1.4 Update and Delete Product Details

As a seller, I want to update the information of my published products at any time. I also want to remove an existing product from the system.

Success

- The update feature allows sellers to view and edit existing information about a product. The changes are saved to the database and reflected in the system.
- The deletion feature allows sellers to remove their listed products at any time. This deletes the product information and associated order information from the database.

Failure

- If the database connection fails, the system notifies the user about the error.
- If sellers try to update data with invalid information (e.g., a non-numeric value for price), the changes do not occur, and the system alerts the users with the reason for failure.

3.1.5 Search and Filter Products

As a buyer, I want to view all products available in the system, search for a specific product by name or parts of the name, and also filter the items by category, price range, etc., to easily find what I need. I want to access all these features without creating a user account to know if my desired product is available.

Success

- On the homepage, all products are shown in a small card format with search and filtering options.
- Products matching the user's search and filter criteria are displayed.

Failure

- If no products match the user's search or filter criteria, the system informs the user.
- If fetching product information from the database fails, the system displays an error message.

3.1.6 Add Product to Shopping Cart and Checkout (Buyer)

As a buyer, I want to add my desired products to the cart, recheck, filter, and then proceed for checkout. I want to complete purchasing all the products in one transaction.

Success

- User can easily add products to cart directly from the product listing page or the details page of the product.
- User can add any available products with a single click.
- User can update quantities of the products before checking out.
- User can also remove any product he is not sure about or doesn't want later before checking out.
- While proceeding for the checkout, there will be a summary showing the order list, shipping charges, discounts, and payment options.
- After successful checkout, the order will be confirmed via email/SMS.
- The order information will be stored in the database and be viewable in the order history.

Failure

- If there is any issue in the server, it will show an Error message.
- If any item is out of stock, then there will be a notification saying "Stock Out" or "Item currently unavailable".
- If the user proceeds for checkout without adding anything to the cart, there will a message saying "Your cart is Empty".
- If during checkout, the payment gateway fails or times out, then it will show an appropriate error message by halting the process.
- In case of network interruption, unsaved cart items will be lost.

3.1.7 Order Placement and Cancellation

Users can make an order of the desired products that have already been added to the shopping cart as a buyer. Moreover, the buyers can cancel their orders within a reasonable period, that is, before the order is shipped to be delivered.

Success

- The system will ask the buyer to provide a delivery point using a form for manually inputting the delivery address. After confirming the given location, the system goes to the order confirmation page where the user can get an overview of his or her order, total cost, and time of delivery.
- In case the buyer requests to cancel an order prior to its delivery, a confirmation message is sent to the buyer as well as the administrator indicating the cancellation of the order. The order status is changed to "Cancelled" and payment or inventory records are also updated accordingly.

Failure

- In case there are no items in the cart, the system will show the following alert message: "No products selected! You may add any items to your cart and continue."
- In case of invalid or incomplete location provided, the system displays: "Please type in the valid location of delivery."
- In case the order has been placed and shipped, the cancellation request will be rejected with an alert message: "Your order once already dispatched cannot be cancelled."

3.1.8 Manage Product Inventory (Seller)

As a Seller, I want to manage my product inventory efficiently so that I can keep track of stock levels, update product details, and ensure accurate availability for the buyers.

Success

- The seller can add new products according to their categories.
- Each product will have its name, description, price, image, and quantity.
- The seller can update the existing product details if necessary.
- When an item is sold, it will be deducted from the stock automatically.
- Sellers can view a summary dashboard showing total items in stock, low stock, and recent sales.
- The system will notify the seller once the inventory level falls below a predefined threshold.
- As the seller makes updates and edits, all changes will be seen by the buyers while browsing.

Failure

- If all the required information regarding the product is not filled in, the system will alert the seller to fill in the mandatory fields before saving.
- If the uploaded image does not meet the requirements such as file size, format, or naming convention, the system will notify the seller.
- If the update request fails due to a technical error, the system stores the data locally until synchronization.
- If a seller tries updating or deleting a product while there are ongoing active orders, it will show a message saying "Cannot delete/update product".

3.1.9 Notification System

The notification system ensures that each user type receives relevant alerts and messages about their responsibilities, actions, and system events. Notifications can be delivered via in-app messages, email, or push notifications.

Success

- **As a buyer, I want to receive notifications** when:
  - My order is placed successfully: "Your order #123 has been confirmed."
  - My order is delivered: "Your order #123 has been delivered."
  - My order is canceled or returned: "Your order #123 has been canceled/returned."
  - A product in my wishlist is back in stock or on discount: "Your wishlist item 'Fresh Apples' is now available."
- **As a system administrator, I want to receive notifications** when:
  - Stock levels are low: "The stock for 'Bananas' is below threshold."
  - Products are successfully added, updated, or deleted.
  - A new order is placed: "New order #456 has been received."
  - Orders are delayed or returned.
  - A customer submits feedback or a complaint.
  - Reports are generated or analytics require attention.

Failure

- If the network is unstable, notifications may be delayed or not sent on time.

- If a user's notification settings are disabled, they will not receive updates.
- If the server is under maintenance, real-time notifications might not be delivered.
- If the system cannot connect to the email or SMS gateway, external alerts will be postponed until service is restored.

3.1.10 Feedback/Rating System (Buyers and Sellers)

The system allows buyers to submit responses for purchased items by providing ratings and feedback, and enables sellers (administrators) to view, interpret, and adjust product quality based on the feedback received. This ensures improved customer satisfaction, transparency, and platform reliability.

Success

- Customers can score goods on a 1–5 star scale and leave a text comment about their experience.
- Feedback is automatically attached to the verified purchase history of the buyer.
- All feedback is available to the sellers (admins) on a separate dashboard for analyzing ratings, comments, and trends to measure product quality and seller performance.
- After submission, feedback and ratings are displayed under the product details section for all users to make informed purchasing decisions.
- The system confirms that the buyer is a verified customer.
- Feedback is stored and displayed under the relevant product.
- The mean rating of products is updated automatically.
- Sellers can access summarized reports and insights through their admin dashboard.

Failure

- If the user is not logged in: "Please log in to give feedback."
- If the user has not purchased the product: "Only verified buyers can rate this product."
- If the rating submission fails due to network issues: "Couple of seconds failed to submit feedback. Please try again later."
- If the system fails to load seller analytics: "Feedback data is not available. Re-fresh, or come back to it later."

Inputs:

- Order ID (verified orders)
- Rating value (1–5 stars)
- Comment text (optional)

Outputs:

- Product ratings and reviews displayed below the product.
- Seller analytics summarizing product averages and user sentiment trends.

Preconditions

- Customer must be signed in to an authenticated account.
- Customer must have purchased the product prior to giving feedback.

Postconditions

- Ratings and feedback are stored in the database and displayed under the respective product.
- Seller feedback can be reviewed and analyzed via the dashboard to improve product quality and business knowledge.

Acceptance Criteria

Scenario 1: Successful Feedback Submission

- Given that the consumer has purchased the product and is logged in, when the buyer submits a rating and comment on the product page, the system stores the feedback and displays it under the product information, updating the average product rating.

Scenario 2: Feedback and Ratings View

- Given any user visits a product page, when they scroll to the comments section, they can view all available ratings, comments, and the average score of the product.

3.1.11 Live Chat

Buyer Perspective

As a Buyer, I want to engage in live chat with the customer support so that I can get quick assistance for my orders, payments, or product inquiries.

- **Scenario 1: Initiating Live Chat**
  - **Given:** I am logged in as a buyer.
  - **When:** I open the support section.
  - **Then:** I see an option to start a live chat with customer support.
- **Scenario 2: Real-Time Messaging**
  - **Given:** The chat session is active.
  - **When:** I send or receive a message.
  - **Then:** Messages appear instantly for both sides, and I get notified of new replies.
- **Scenario 3: Ending Chat Session**
  - **Given:** My issue is resolved.
  - **When:** I choose to end the chat.
  - **Then:** I can close the chat window, and get back to the main page.

System Administrator Perspective

As a System Administrator, I want to manage live chat requests from buyers so that I can assist them efficiently and maintain service quality.

- **Scenario 1: Receiving Chat Requests**
  - **Given:** A buyer initiates a chat.
  - **When:** The request appears in my dashboard.
  - **Then:** I can view the buyer's name, issue details, and accept the chat session.
- **Scenario 2: Responding to Messages**
  - **Given:** The chat is active.
  - **When:** I reply to the buyer.
  - **Then:** Messages are delivered instantly, and the conversation updates in real time.
- **Scenario 3: Closing the Conversation**
  - **Given:** The conversation is complete.
  - **When:** I end the chat session.
  - **Then:** The system logs the conversation and marks the issue as resolved.

Failure

- If the network connection is unstable, messages may be delayed or fail to send.
- If notifications are disabled for either party, new messages may go unnoticed.
- If the server is under maintenance, live chat sessions may be temporarily unavailable.
- If the buyer's account is invalid or deleted, the chat cannot be initiated.

3.2 Non-Functional Requirements

3.2.1 Performance Requirements

- **Response Time:** Within standard operating conditions, the system will support user interactions (e.g., login, product search, cart updates) within 2 seconds to provide a smooth user experience.
- **Scalability:** The system should remain stable with up to 25% of concurrent users during peak times (e.g., promotional events).
- **Database Performance:** Product search, filtering, and order queries should execute in under 1 second with datasets of up to 100 products.
- **API Efficiency:** Every API developed using FastAPI in the backend will be tuned to achieve a response time of 1.5 seconds under typical load.

3.2.2 Safety Requirements

- **User Data Protection:** The system must safeguard sensitive user data (names, contact details, delivery address, and payment details) against unauthorized access using secure database processes and hashed credentials.
- **Session Safety:** Sessions will automatically expire after 10 minutes of inactivity to prevent unauthorized access.

3.2.3 Security Requirements

- **User Authentication:** All users (buyers and administrators) are required to undergo secure authentication using verified email and password, with optional two-factor authentication for administrators.
- **Authorization and Role Control:** The system should implement role-based access control (RBAC), allowing administrators to manage products, orders, and feedback, while buyers can only manage their personal accounts.

3.2.4 Software Quality Attributes

- **Usability:** The user interface should be user-friendly and uniform, providing easy navigation for both customers and administrators. The goal is to achieve a 90% satisfaction rate in usability testing.
- **Reliability:** The system should operate 99.9% of the time, excluding planned maintenance. Critical functions such as login, product search, and checkout must always remain operational.
- **Maintainability:** The application code should be modular and well-documented, enabling developers to identify and correct issues or implement new features without disrupting other modules.

3.2.5 Business Rules

- **Order Eligibility:** Only verified users with valid delivery addresses can place grocery orders.
- **Order Modification Window:** Buyers can revise or cancel orders before shipment to ensure smooth delivery coordination.
- **Feedback Submission:** Feedback and ratings can only be submitted by verified buyers who have purchased a product. This feedback is available through seller (admin) dashboards for analytics.
- **Product Listing Control:** Only administrators can add, modify, or delete product listings to ensure inventory accuracy and product authenticity.

3.2.6 Maintainability Requirements

- **Architecture:** The system architecture must be modular, employing microservices that can be maintained independently (e.g., authentication, product management, and order modules).
- **Deployment:** Code updates, bug fixes, and patches should be deployable without rebooting the entire system to minimize downtime.

3.2.7 Availability

- **System Availability:** The system will operate 24/7, with downtime occurring only during scheduled maintenance.