Faculty of Engineering and Technology

Computer Science Department

SOFTWARE ENGINEERING - COMP433

Final Project Report

Agricultural Shop

Group No: G8

Instructor: Dr.Adel Taweel

Student Name: Osama Quttenh	ID: 1222825
Student Name: Hamza Najar	ID: 1192605
Student Name: Yazan Yousef	ID: 1191706
Student Name: Amro hammad	ID:1201010

Section: 4

Date: 17– 1 – 2025

Table of Contents

Chapter 1: Project Planning and Management

1.1. Names of Editor/writer of the Report:

Amro Hammad

1.2. Business Title:

Agricultural Shop

1.3. Group:

Group Name: Unity Group

Group Number: G8

1.4. Name of students:

Student Name: Amro Hammad	ID:1201010
Student Name: Osama Quttenh	ID: 1222825
Student Name: Hamza Najar	ID: 1192605
Student Name: Yazan Yousef	ID: 1191706

1.5. Role of each member:

Manager: Osama Quttenh

Secretary: Hamza Najar

Technical Architect: Amro Hammad

Programmer: Yazan Yousef

1.6. Project management strategy:

The team meets twice a week for status updates, problem-solving, and planning. Decisions are made collaboratively, with the project manager having the final say if needed. Following an Agile software process model, the team uses sprint planning, daily stand-ups, and retrospectives to ensure flexibility and continuous improvement

1.7. Project manager report:

Initially, we divided the group's work equally among its members. Each member focused on individual tasks while collaborating on group assignments.

Osama Quttenh ,as the manager, assumed the roles of task organizer and integrated task coordinator. He led the creation of the Actor analysis and Detailed class model, and actively reviewed and discussed progress across various group tasks. Osama also contributed to defining User Requirements and corresponding System Requirements. Additionally, he developed a scenario demonstrating Feedback on the website, including activity diagram, sequence diagram, and Use-Case Specification.

Hamza Najar, serving as the secretary, took charge of coordinating group meetings and managing time estimation calculations. She played a pivotal role in developing system class diagrams and defining user and system requirements for signup and searching functionalities.
Hamza also designed the deployment diagram and created a signup scenario on the website. His contributions extended to crafting activity and sequence diagrams, along with a comprehensive Use-Case Specification tailored to her specific scenario.

Amro Hammad, serving as the technical architect, led the development of the purchasing scenario and purchase use case description in phase 3. He meticulously defined user and system requirements for shopping cart functionality and company information in phase 2. Amro also formulated the component diagram and developed a detailed purchasing scenario on the website. His deliverables included Use-Case Specification, activity diagram, and sequence diagrams specific to his scenario.

Yazan Yousef, serving as the programmer, spearheaded the creation of Activity diagrams, System Design Goals, and System and Deployment Modeling. He actively reviewed and discussed various aspects of all group tasks and contributed System Requirements aligned with User Requirements. Ali designed an activity diagram and scenario focused on searching for products, and he developed activity and sequence diagrams, alongside Use-Case Specification, for his specific scenario.

1.8. Group members report:

Amro Hammad :

I believe the project was executed as intended, with tasks distributed effectively among team members. Everyone contributed their effort and fulfilled their responsibilities. Clear and consistent communication played a key role in ensuring all tasks were completed accurately and within the expected timeframe, I led the formulation of design goals with a focus on system architecture and performance requirements. I developed the deployment diagram to illustrate the system's infrastructure and component distribution. I ensured that the technical requirements were aligned with the implementation strategies.

Osama Quttenh :

In my opinion, the project proceeded as planned, with an equitable distribution of workload among the team. Each member diligently performed their duties, and effective communication facilitated the timely and accurate completion of all tasks , As Product Manager, I led the actors' analysis and the development of the component diagram. I managed equal task distribution among team members and fostered a collaborative review process, ensuring feedback was effectively implemented across deliverables. I contributed to the system requirements documentation and oversaw the integration of component functionalities. I ensured a balanced workload and maintained project momentum through effective team coordination.

Hamza Najar:

The project tasks distribution was fair, everyone did their job as required, the project was completed successfully without delay and with high efficiency. My main contribution in the project , I directed the creation of the analysis class diagram, focusing on establishing relationships between classes and their attributes. I also led the development of the general use case diagram to effectively capture system functionality and actor interactions. I ensured the proper integration of use cases while maintaining architectural consistency throughout the process.

Yazan Yousef:

I think the project was done as it's supposed to be, the load have been distributed well, all group members gave their effort and done their duty and communication was effective which led in the end to complete all the tasks in the correct form and on time , I spearheaded the development of the detailed class diagrams and activity diagrams. I coordinated with team members, each contributing two classes to the detailed class diagram, and then integrated and refined the complete diagram structure. I also ensured consistency across the diagrams and maintained proper relationships between components.

Chapter 2: Requirement Elicitation, Analysis and Modelling

2.1. Requirement statement/Business Description:

Business: Agriculture Shop Online

Business Description/Outline

Business Overview: The online agriculture shop is a digital marketplace focused on providing agricultural products directly to customers across Palestine. It aims to support both professional farmers and local gardeners by making it easier to purchase essential farming supplies without needing to travel long distances. This platform provides a convenient solution, ensuring essential products are available and delivered to customers' doors.

Services Offered:

1. **Product Sales:**
   - A wide selection of agricultural products such as seeds, fertilizers, and small tools.
   - Options for traditional and organic farming, including gardening supplies, soil enhancers, and pest control solutions.
2. **Home Delivery:**
   - Ensures accessibility by delivering products across urban and rural areas within Palestine.
3. **Customer Support and Advisory:**
   - Assists users with product use and ordering.
   - Offers agricultural guidance to help customers choose the best products for their needs.

Business Capacity:

1. **Customer Base:**
   - Initially aims to serve around 2,500 customers monthly, with plans to expand accessibility to remote areas.
   - Anticipates growth in monthly active users over time.
2. **Concurrent Users:**
   - Plans to support around 500 users concurrently during peak times.
   - Scalability ensures smooth operation as user numbers grow.
3. **Product Range and Inventory:**
   - Items tailored for agriculture, including seeds, fertilizers, tools, machinery, crop protection products, and irrigation equipment.
   - Monthly sales volumes expected to range from 4,000 to 6,000 items, influenced by seasonal demand.
4. **Product Categories:**
   - Includes seeds, soil enhancers, and basic irrigation supplies for large-scale farming and small-scale gardening.

Business Model and Processes: Operating as a direct-to-consumer (D2C) e-commerce platform, the business generates revenue through product sales, small delivery fees, and limited promotional partnerships. Key processes include:

1. **Browsing and Selection:**
   - Customers log into the platform via mobile or desktop to browse categorized products with detailed descriptions, pricing, and reviews.
2. **Order and Payment Process:**

    - Customers add products to their cart and proceed to checkout, selecting delivery options and payment methods like cash on delivery (COD) or digital payments.
3. **Inventory and Order Fulfillment:**
    - Real-time inventory updates occur upon order placement, notifying the team for processing.
4. **Delivery and Tracking:**
    - Orders are dispatched through local delivery services, with customers updated on their order status.
5. **After-Sales Support:**
    - Customer service assists with inquiries related to orders, product use, returns, or replacements.

Employee Roles and Responsibilities:

1. **Product Management:**
    - Manages the product catalog, sources new items, and maintains inventory levels.
2. **Customer Service:**
    - Resolves customer inquiries, order issues, and post-purchase support.
3. **Marketing and Outreach:**
    - Builds awareness, launches promotional campaigns, and engages customers on social media platforms.
4. **Logistics and Delivery:**
    - Coordinates with delivery partners and manages order processing to ensure timely delivery.
5. **IT and Development:**
    - Maintains the platform, ensures regular updates, and supports data security.

Stakeholders and Users:

1. **Primary Users:**
    - Farmers and gardeners in urban and rural Palestinian areas needing reliable access to agricultural supplies.
2. **Secondary Stakeholders:**
    - Local delivery services, suppliers, and external service providers supporting logistics and supply chain management.
3. **Internal Stakeholders:**
    - Employees and management team members handling product sourcing, customer service, and platform maintenance.

Deployment Environment:

1. The platform operates as a mobile-friendly web application accessible via desktop, mobile, or tablet.
2. Optimized for areas with limited connectivity, ensuring efficient, low-bandwidth functionality for reliable access in rural or underserved areas.

2.2. USER and SYSTEM Requirements:

Osama Quttenh 1222825 (MODIFIED)

Search and Filter Functionality

UR1.0: Users shall be able to search for agricultural products.

- SR1.1: The system shall accept search queries in both Arabic and English characters.
- SR1.2: The system shall match search queries against product names, descriptions, and metadata.
- SR1.3: The system shall return search results within 2 seconds of query submission.
- SR1.4: The system shall support exact phrase matching when text is enclosed in quotation marks.
- SR1.5: The system shall handle common spelling variations and mistakes in search queries.

UR2.0: Users shall be able to filter agricultural products.

- SR2.1: The system shall provide filtering by product categories (Seeds, Fertilizers, Tools, Equipment).
- SR2.2: The system shall enable filtering by price range with specific minimum and maximum values in local currency.
- SR2.3: The system shall allow filtering by stock availability status (In Stock, Out of Stock, Pre-order).
- SR2.4: The system shall provide filtering by brand name from the system's approved brand list.
- SR2.5: The system shall enable filtering by product rating on a scale of 1-5 stars.
- SR2.6: The system shall allow selection of multiple values within each filter category.
- SR2.7: The system shall process filter combinations using AND/OR logic.
- SR2.8: The system shall update filtered results within 1 second of filter application.
- SR2.9: The system shall maintain selected filters until explicitly cleared by the user.

UR3.0: Users shall be able to view and navigate product listings.

- SR3.1: The system shall display product cards containing:
  - Product thumbnail image
  - Product name (maximum 100 characters)
  - Current price in local currency
  - Availability status (In Stock/Low Stock/Out of Stock)
  - Average rating (if available)

  - Discount percentage (if applicable)

- SR3.2: The system shall provide sort options by price, rating, and relevance.
- SR3.3: The system shall display the total number of matching products
- SR3.4: The system should update availability status in real-time with maximum 5-minute delay.
- SR3.5: The system shall display 20 products per page by default with pagination controls.
- SR3.6: The system shall display a “No results found” message and suggest related terms when no products match the search criteria.

Product Details Display

UR1.0: Users shall be able to view complete product specifications on dedicated product pages.

- SR1.1: The system shall display the product name with a maximum of 100 characters.
- SR1.2: The system shall show the current price in local currency with 2 decimal places.
- SR1.3: The system shall indicate stock levels as “In Stock”, “Low Stock” (<10 units), or “Out of Stock”.
- SR1.4: The system shall display the manufacturer name and logo.
- SR1.5: The system shall show the full category path from root to current product.
- SR1.6: The system shall display the unique product reference number.

UR2.0: Users shall be able to view and interact with product imagery.

- SR2.1: The system shall display a minimum of 3 product images per product.
- SR2.2: The system shall render the main product image at minimum 800x800 pixels resolution.
- SR2.3: The system shall display thumbnail images at 150x150 pixels resolution.

UR3.0: Users shall be able to access detailed technical specifications.

- SR3.1: The system shall display product dimensions in metric units (length, width, height).
- SR3.2: The system shall show product weight in kilograms with 2 decimal precision.
- SR3.3: The system shall display storage temperature range in Celsius.
- SR3.4: The system shall show product shelf life in months.
- SR3.5: The system shall indicate seasonal applicability by month range.

UR4.0: Users shall be able to view product usage instructions.

- SR4.1: The system shall display step-by-step numbered application methods.
- SR4.2: The system shall highlight safety hazards with red warning labels.
- SR4.3: The system shall show recommended usage quantities per area unit.

UR5.0: Users shall be able to view comprehensive pricing information.

UR6.0: Users shall be able to access and navigate product reviews.

- SR6.1: The system shall show reviewer's star rating.
- SR6.2: The system should display review submission date in "DD/MM/YYYY" format.
- SR6.3: The system should display helpful vote count for each review.
- SR6.4: The system shall provide a "Helpful" button for each review.
- SR6.5: The system shall limit each user to one helpful vote per review.
- SR6.6: The system should update helpful vote count within 1 second of submission.
- SR5.1: The system shall display the base price in local currency.
- SR5.2: The system shall show the discount amount when applicable.
- SR5.3: The system shall display the final price after all discounts.
- SR5.4: The system shall show bulk pricing tiers when applicable.
- SR5.5: The system shall display price per unit measure.

Purchase History Management

UR1.0: Users shall be able to view their purchase transaction details.

- SR1.1: The system shall show product name with link to current product page.
- SR1.2: The system should display purchase price in local currency with 2 decimal places.
- SR1.3: The system shall indicate order status ("Processing", "Shipped", "Delivered", "Cancelled").
- SR1.4: The system should show delivery status with percentage completion bar.

UR2.0: Users shall be able to navigate through their purchase records.

- SR2.1: The system should load 20 transactions per page, sorted by date descending.
- SR2.2: The system shall implement pagination with "Previous" and "Next" controls.
- SR2.3: The system shall display total number of available transaction records.
- SR2.4: The system should load additional transaction pages within 1 second of request.
- SR2.5: The system should enable date range filtering within the past 24 months.
- SR2.6: The system should give suggest to the user to redirect to browsing page if no previous purchases have been made.

Product Managment

UR1.0: The Product Manager shall be able to add, update and remove products.

- SR1.1: The system shall enable product creation with the following required fields:

  - Product name (3-100 characters)
  - Product description (10-5000 characters)
  - Base price (in local currency)
  - SKU (unique identifier)
  - Stock quantity
- SR1.2: The system shall enable price modifications with:
  - New price validation (non-negative values)
  - Price change history logging
- SR1.3: The system shall enable stock level adjustments.
- SR1.4: The system shall enable category reassignment.
- SR1.5: The system shall enable product information updates with:
  - Name and description edits
  - SKU updates (with uniqueness validation)
  - Specification updates
- SR1.6: The system shall support product image management with:
  - Minimum 3 images requirement
  - Maximum 10 images limit
  - Supported formats (JPG, PNG)
  - Maximum file size of 5MB per image
- SR1.7: The system shall provide an archive function that:
  - Maintains product data
  - Removes product from active listings
- SR1.8: The system shall provide a remove function that permanently deletes product data and removes all associated images.
- SR1.9: The system shall display a confirmation dialog for archive/remove actions containing:
  - Product name and SKU
  - Selected action (archive/remove)
  - Consequences of the action
  - Require explicit confirmation
  - Option to cancel ### UR2.0: The Product Manager shall be able to view and analyze product sales data.
- SR2.1: The system shall display a dashboard containing:
  - Daily sales total
  - Weekly sales total
  - Monthly sales total
  - Top 5 selling products

  - Bottom 5 selling products
- SR2.2: The system shall generate sales reports with:
  - Sales by category
  - Sales by product
  - Revenue trends
  - Profit margins
  - Stock levels
- SR2.3: The system shall provide date range selection with:
  - Today
  - Last 7 days
  - Last 30 days
  - Custom date range
- SR2.4: The system shall enable report export in:
  - PDF format
  - Excel format
  - CSV format
- SR2.5: The system shall update dashboard data:
  - Sales data every hour
  - Stock levels every 2 hours
  - Revenue calculations every hour

.

Amro Hammad - 1201010

UR1.0: Users shall be able to create, modify, and cancel delivery schedules via a user-friendly interface.

- **SR1.1:**
  - The system shall enable creating, modifying, and canceling delivery schedules via a user-friendly interface.
  - Store delivery schedule details (e.g., time, customer ID) in a relational database.
  - Scheduling actions must complete within 2 seconds.
- **SR1.2:**
  - The system shall integrate GPS tracking to monitor deliveries and calculate real-time Estimated Time of Arrival (ETA).
  - Store GPS coordinates and delivery status in real-time.
  - ETA updates must be calculated and displayed within 1 second.
  - Secure transmission of location data to prevent unauthorized access.
- **SR1.3:**

    - The system shall send real-time notifications for delivery updates (e.g., ETA, delays).
    - Log notification status in the system (sent, failed, etc.).
    - Notifications must be sent within 2 seconds of a delivery status update.
    - Minimize sensitive information in notifications.

UR2.0: Users expect delivery routes to be optimized for efficiency and timeliness.

- **SR2.1:**
    - Use algorithms to calculate the most cost-effective and time-efficient routes.
    - Store route details and optimization factors (e.g., fuel cost, travel time).
    - Route calculations must complete within 10 seconds for up to 50 stops.
- **SR2.2:**
    - The system shall adjust delivery routes based on traffic, road closures, etc.
    - Update delivery schedules and route information in real-time.
    - Adjustments must propagate to drivers within 3 seconds of receiving updates.
    - Secure data transmissions for real-time adjustments.

UR3.0: Delivery Address Management

- **SR3.1:**
    - Users shall be able to add delivery addresses to customer profiles.
    - Store address data in customer profiles.
    - Address addition must complete within 2 seconds.
    - Encrypt address data during storage and transmission.
- **SR3.2:**
    - Users shall be able to edit existing delivery addresses in customer profiles.
    - Real-time updates to stored address data in customer profiles.
    - Provide API endpoints for editing addresses, ensuring data consistency.
    - Address edits must complete within 1-2 seconds across all connected systems.
    - Ensure encrypted transmission of address edits and secure data storage.
- **SR3.3:**
    - Users shall be able to delete delivery addresses from customer profiles, ensuring no active orders are linked to deleted addresses.
    - Handle address deletion while maintaining data integrity.
    - Provide API endpoints to securely delete addresses and verify active order linkage.
    - Address deletion should occur within 1 second.
    - Ensure data consistency, audit logs for deletion actions, and secure deletion processes.
- **SR3.4:**
    - The system shall validate delivery addresses through external services like Google Maps API to ensure accuracy and format compliance.
    - Store validated address information, with validation results logged.

  - Address validation must complete within 3 seconds.
  - Ensure secure transmission of address data during validation.
- **SR3.5:**
  - The system shall display success or error messages immediately after address modifications (add/edit/delete).
  - Log user feedback status for tracking and audit purposes.
  - Feedback messages must appear within 1 second of the operation completion.
  - Secure handling of sensitive feedback data.
- **SR3.6:**
  - The system shall log all changes to delivery addresses with timestamps, user IDs, and action types for audit and troubleshooting.
  - Logging operations should not impact address modification performance and must be performed within 1 second.
  - Encrypt sensitive log data and implement role-based access for log viewing.

UR4.0: Users shall be notified if emergencies affect delivery schedules.

- **SR4.1:**
  - The system shall monitor real-time traffic and road conditions for closures or blockades.
- **SR4.2:**
  - The system shall reroute deliveries automatically during emergencies and provide updated ETAs.
  - Detect emergency situations such as road closures, accidents, or extreme weather conditions using real-time data sources (e.g., traffic reports, weather APIs, GPS data).
- **SR4.3:**
  - The system shall send notifications to users about emergencies and allow them to reschedule deliveries quickly after identifying the case.

UR5.0: Users shall receive real-time updates on the status of their deliveries.

- **SR5.1:**
  - The system shall update delivery statuses at every stage, including:
    - "Order Placed": Status set when the customer successfully places an order.
    - "In Transit": Status updated when the order is picked up and on its way to the customer.
    - "Delivered": Status updated once the order reaches the customer, linked to proof of delivery (e.g., signature, photo).
- **SR5.2:**
  - The system shall provide proof of delivery through digital signatures or photos.

1. Customer Account Management

- **UR1.0:** The system shall allow users (customers) to create accounts and manage them. These accounts are used for accessing personal order information and order history.
    - **SR1.1:** The system shall store customer information in a securely encrypted and hashed database that only the user can access.
    - **SR1.2:** The system shall have a section that shows the history of orders completed by the customer, including:
        - Order date
        - Order status
        - Products
    - **SR1.3:** The system shall enable customers to edit their personal information, including:
        - Name
        - Address
        - Payment methods
        - Contact options (phone/email)
    - **SR1.4:** The system shall provide a password reset feature via a time-limited link sent to the selected contact option.
    - **SR1.5:** The system should provide multi-factor authentication for extra security.

---

2. Customer Support System

- **UR2.0:** The system shall include a support section that allows users to submit inquiries about accounts, services, and item issues.
    - **SR2.1:** The system shall have a ticketing feature with options to divide requests into categories such as:
        - Product issues
        - Delivery issues
        - Account issues
    - **SR2.2:** The system shall store the following data for tickets:
        - Customer ID and name
        - Category
        - Time
        - Ticket status
    - **SR2.3:** The system shall send an automated email when a ticket is submitted and update the user via email whenever the ticket status changes, including states:
        - Sent
        - In progress
        - Solved
        - Closed
    - **SR2.4:** The support interface shall update ticket statuses periodically (every few minutes) to allow real-time follow-up.

---

3. Product Usage Guidance & After-Sale Support

- **UR3.0:** The system shall provide product usage guidance and post-purchase support.
  - **SR3.1:** The system shall display the following properties for verified purchasers:
    - Application methods
    - Safety instructions
    - Best practices
    - Return and replace policies
  - **SR3.2:** Verified purchasers shall be able to submit inquiries for additional usage guidance after purchase.
  - **SR3.3:** Post-purchase inquiries shall be stored and categorized within the ticketing system for tracking and resolution.

---

4. Feedback and Complaint Handling

- **UR4.0:** The system shall allow customers to submit feedback and complaints about products, delivery, and overall service quality.
  - **SR4.1:** Feedback data shall be stored in a database, including:
    - Customer ID
    - Product ID
    - Rating (1-5)
    - Comments (optional)
  - **SR4.2:** A complaint submission form shall categorize complaints into:
    - Delivery issues
    - Product issues
    - Service issues
  - **SR4.3:** Each complaint shall be assigned a unique reference number, and the system shall send an automated acknowledgment email to the user along with updates.
- **Performance Requirement:**
  - The system shall handle up to 2,500 feedback/complaint submissions monthly.

Hamza Najar | 1192605: Order & Payment Processing Focus

Shopping Cart Functionality

- **UR1.0:** The system shall enable users to add, remove, and modify products in a shopping cart.
  - **SR1.1:** The system shall allow users to add products to the cart directly from product listing pages or product detail pages.
  - **SR1.2:** The system shall enable users to view the cart with the following details:

        - Product name
        - Quantity
        - Price per unit
        - Subtotal per product
        - Total cart value
    - **SR1.3:** The system shall allow users to update product quantities directly in the cart.
    - **SR1.4:** The system shall notify users if their selected quantity exceeds available stock.
    - **SR1.5:** The system shall display promotional discounts, if applicable, for cart items.
    - **SR1.6:** The system shall allow users to save a shopping cart for later use.
    - **SR1.7:** The system shall automatically empty the cart after 30 days of inactivity.

Checkout Process

- **UR2.0:** The system shall provide a seamless multi-step checkout process for users.
    - **SR2.1:** The system shall include the following steps in the checkout process:
        1. Review Cart
        2. Enter Shipping Details
        3. Select Payment Method
        4. Confirm Order
    - **SR2.2:** The system shall validate shipping details, ensuring required fields are completed.
    - **SR2.3:** The system shall calculate shipping fees based on user location and total weight of products.
    - **SR2.4:** The system shall enable users to apply discount codes or vouchers during checkout.
    - **SR2.5:** The system shall support guest checkout for users who do not want to create an account.
    - **SR2.6:** The system shall provide a summary of the order, including:
        - Itemized product list
        - Total cost (inclusive of shipping and discounts)
    - **SR2.7:** The system shall prompt users for confirmation before placing an order.

Payment Methods

- **UR3.0:** The system shall support multiple payment methods for order processing.
    - **SR3.1:** The system shall allow users to pay using the following methods:
        - Cash on Delivery (COD)
        - Credit/Debit cards
        - Digital payment platforms (e.g., PayPal, Google Pay, etc.)
    - **SR3.2:** The system shall securely store payment details for returning users who opt-in.

    - **SR3.3:** The system shall validate all payment transactions and display a success or failure message.
    - **SR3.4:** The system shall offer users an option to split payments across multiple methods.

Order Confirmation System

- **UR4.0:** The system shall notify users about their order status after placement.
    - **SR4.1:** The system shall generate a unique order ID for each completed transaction.
    - **SR4.2:** The system shall send an order confirmation email or SMS immediately upon order placement.
    - **SR4.3:** The system shall provide the following details in the order confirmation:
        - Order ID
        - Itemized list of purchased products
        - Total amount charged
        - Estimated delivery date
    - **SR4.4:** The system shall allow users to track their order status (e.g., Pending, Processed, Shipped, Delivered).
    - **SR4.5:** The system shall update the order status in real-time based on inventory and logistics updates.

Inventory Management Updates

- **UR5.0:** The system shall integrate inventory updates with the order processing system.
    - **SR5.1:** The system shall reduce product stock levels automatically when orders are confirmed.
    - **SR5.2:** The system shall send low-stock alerts to administrators when stock levels fall below a predefined threshold.
    - **SR5.3:** The system shall display "Out of Stock" or "Limited Stock" labels for products nearing zero availability.
    - **SR5.4:** The system shall update inventory levels in real-time for cancellations or returns.
    - **SR5.5:** The system shall generate monthly inventory reports, tracking:
        - Total items sold
        - Current stock levels
        - Restocking requirements

2.3. SCENARIOs

2.3.1Scenario: Viewing Purchase History Osama Quttenh 1222825 (MODIFIED)

Initial assumption: The user is authenticated and logged into their account in the e-commerce system.

Normal: The user clicks on “View Purchase History” in their account dashboard. The system displays their 20 most recent transactions, showing product names (as clickable links), purchase prices, and order statuses. Each order includes a visual progress bar indicating delivery status. The user navigates through older transactions and applies a date filter to view orders from the last 6 months.

Alternative: The user has no previous purchases. The system displays a “No Records Found” message and provides a direct link to the product browsing page.

Error: The user’s session expires while viewing their history. The system redirects them to the login page with an “Authentication Required” message.

Other activities: The system updates the user’s last access timestamp and maintains logs of history access for security purposes.

System state on completion: The user has viewed their formatted purchase history or been redirected appropriately based on authentication status or purchase record availability.

Scenario 2: Modifying a Delivery Schedule

2.3.2 Amro Hammad (Technical Architect): 1201010 (modified)

- **Normal Flow:** A user logs in to the delivery management system, navigates to "Delivery Schedules," and selects a specific schedule to modify. They edit the delivery date and add a note, such as "Leave at the back door." The system validates the inputs, ensures no conflicts, updates the schedule, and sends a confirmation notification.
- **Alternative Flow:** The user enters invalid data (e.g., incorrect date or incomplete fields). The system highlights the errors and displays a message: "Invalid input. Please review the highlighted fields and try again." After correcting the issues, the user successfully submits the changes.
- **Error Flow:** A database failure prevents saving the changes. The system logs the error and notifies the user with: "System Error. Please try again later." The user can retry or contact support.

2.3.4Scenario 3: Managing a Shopping Cart

Hamza Najar (Secretary): 1192605

- **Initial Assumption:** The user is logged in and browsing products to add items to their cart and proceed to checkout.
- **Normal Flow:** The user adds an item to the cart, reviews it, updates the quantity, and applies a promotional code during checkout. The system validates the code, adjusts the total, and processes the payment. The system generates a unique order ID and sends a confirmation email with the order details.
- **Alternative Flow:** The product is out of stock, and the system prevents adding it to the cart, displaying an "Out of Stock" message.
- **Error Flow:** An invalid promotional code is entered, or payment fails due to insufficient funds. The system notifies the user and prompts them to correct the issue or try a different payment method.
- **System State on Completion:** The order is confirmed, inventory is updated, and a confirmation notification is sent.

2.3.5 Scenario 4: Submitting a Support Ticket

Yazan Yousef (Programmer): 1191706

- **Initial Assumption:** The user is logged into their account and encounters an issue with an order, product, or account.
- **Normal Flow:** The user navigates to the support tab, submits a ticket by selecting a category, and provides a description of the issue. The system assigns a ticket ID, displays the ticket status, and sends a confirmation email. Updates are sent as the ticket is processed.
- **Alternative Flow:** If the issue doesn't fit a predefined category, the user selects "Other" and writes a custom description. If the user cannot log in, they contact support via phone or email, and the team creates a ticket on their behalf.
- **Error Flow:** Mandatory fields are left empty, prompting the system to display an error message. In case of a system outage, the user is advised to retry later.
- **System State on Completion:** The ticket status is updated to "Closed," and a solution summary is emailed to the user. The ticket remains accessible in the user account.

2.4. Effort/Time Estimation Calculation: (MODIFIED)

- (Lead: Hamza, reviewing: Osama, Amro, discussion: Yazan)

UR	Estimated Effort	Estimated No of Developers	Total Effort
UR.1	2PW	3DEV	6PW
UR.2	2PW	3DEV	6PW
UR.3	2PW	3DEV	6PW
UR.4	1PW	2DEV	2PW
UR.5	3PW	3DEV	9PW
UR.6	1PW	2DEV	2PW
UR.7	2PW	2DEV	4PW
Total Effort/Avg	13PW	18/7 = 2.5 DEV. on avg needed	35PW
Schedule Time 30%	13 * 1.3 = 16.9 (min time to complete)		35*1.3 = 45.5 (max time to complete)
Cost		Avg salary=250$	200*45.5=9100$
Profit Margin **Min=10%** **Max=30%**		Min Cost ➔ Max Cost ➔	9100*1.10=10,010$ 9100*1.30=11,830$

The table offers an overview of the time and personnel required to complete various development tasks, based on the involvement of 4 developers. It presents estimates for each task, the total project duration, and the related costs, considering both minimum and maximum profit margins. This summary is crucial for comprehending the project's resource allocation and financial implications.

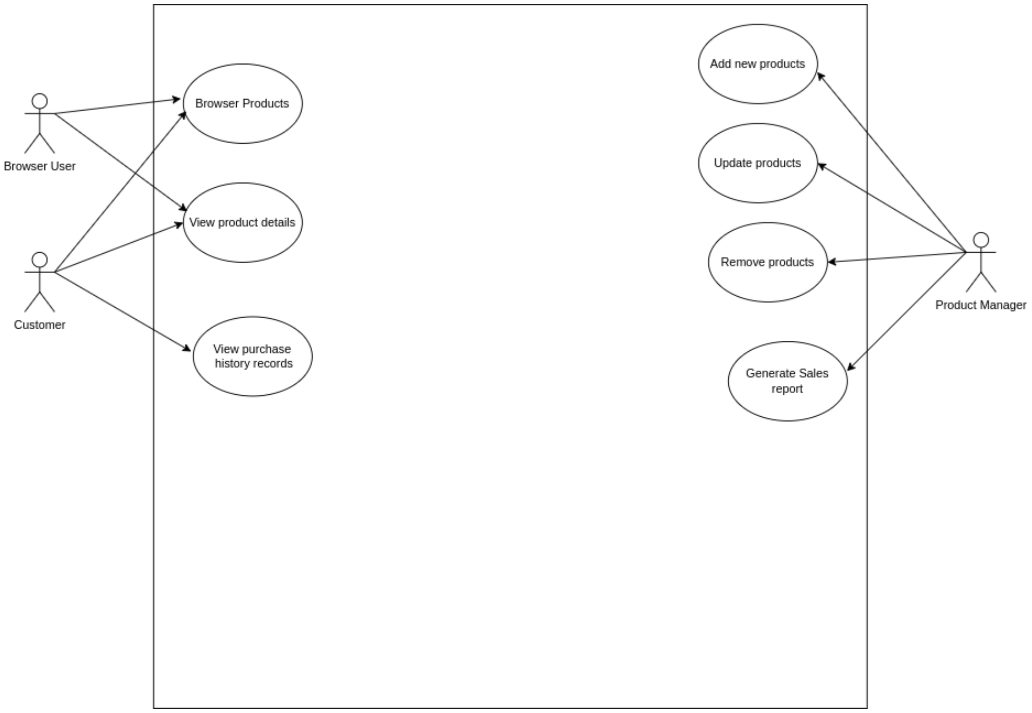
2.5. Actor analysis:

(Lead: Osama, reviewing: Yazan, Amro, discussion: Hamza)

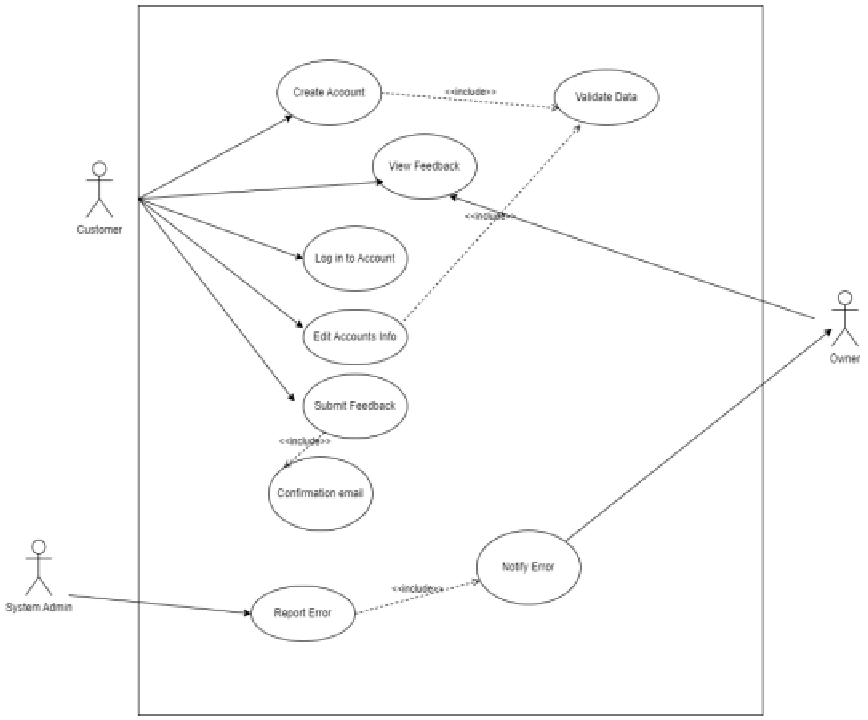
Actor	Description
User	Interacts with the system to browse products, search and filter items, view product details, and create a new account. Limited to basic viewing and searching functionalities before registration.
Customer	Has all Browser User capabilities plus the ability to manage their account, add products to cart, process payments, track delivery status, rate products, report issues, view purchase history, and manage delivery schedules. Can also write product reviews and report corrupt/damaged products.
Product Manager	Responsible for product catalog management including adding new products, updating existing products, removing products, managing stock availability, viewing sales dashboard, and generating sales reports. Has access to total sales analytics and inventory management functions.
Delivery Manager	Manages delivery operations including creating and modifying delivery schedules, setting schedule availability, managing delivery routes, declaring delivery emergencies, and updating delivery status. Has authority to mark schedules as unavailable and monitor delivery operations.
Payment Service	Handles transaction approvals for customer payments, validates payment information, and ensures secure financial transactions. Limited to payment processing and transaction approval functionalities.
Location Service	External service integrated to calculate optimal delivery routes for efficient delivery management. Provides route optimization capabilities to support the delivery management system.
Authorization Service	Manages user authentication through external identity providers, verifies user credentials, and maintains secure session tokens. Limited to authentication flows and user identity verification functionalities.

2.6. USE-CASE Diagram:

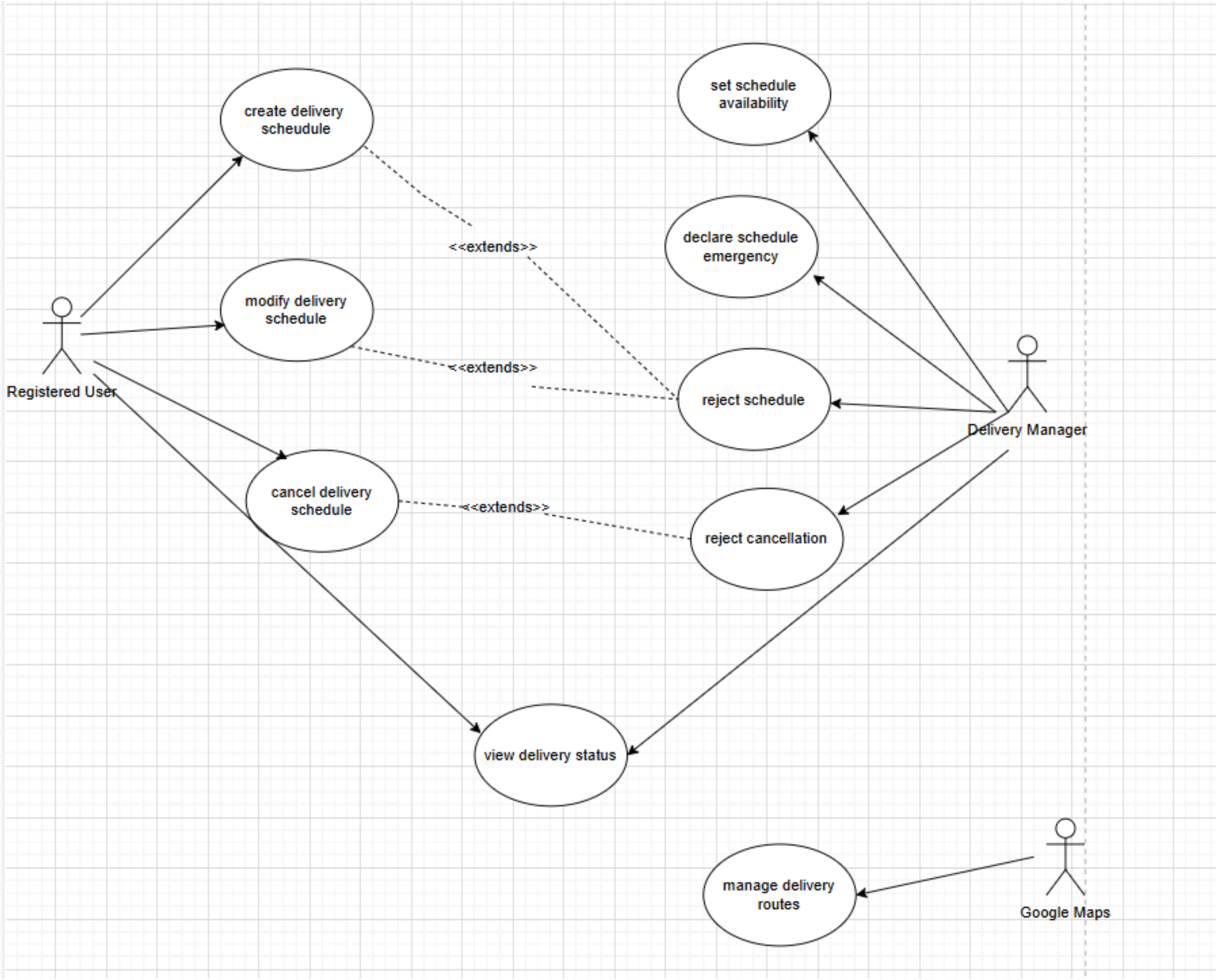
2.6.1 Osama Quttenh (Product Manager): 1222825



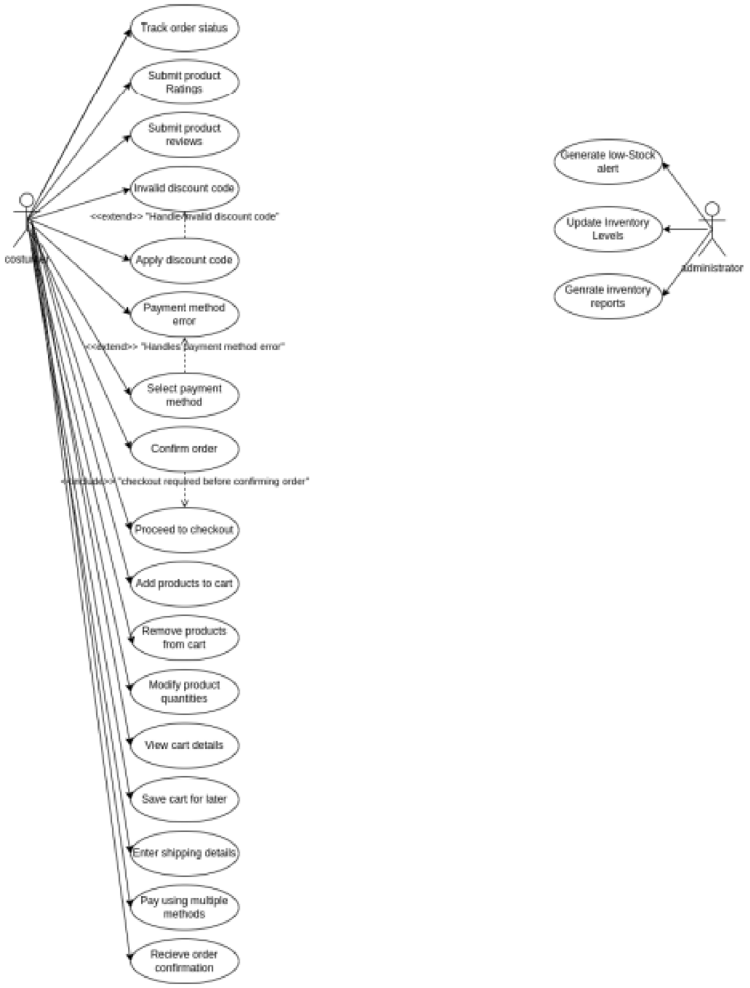
2.6.2 Yazan Yousef (Programmer): 1191706



2.7.2 Amro Hammad (Techincal Architect): 1201010 (MODIFIED)

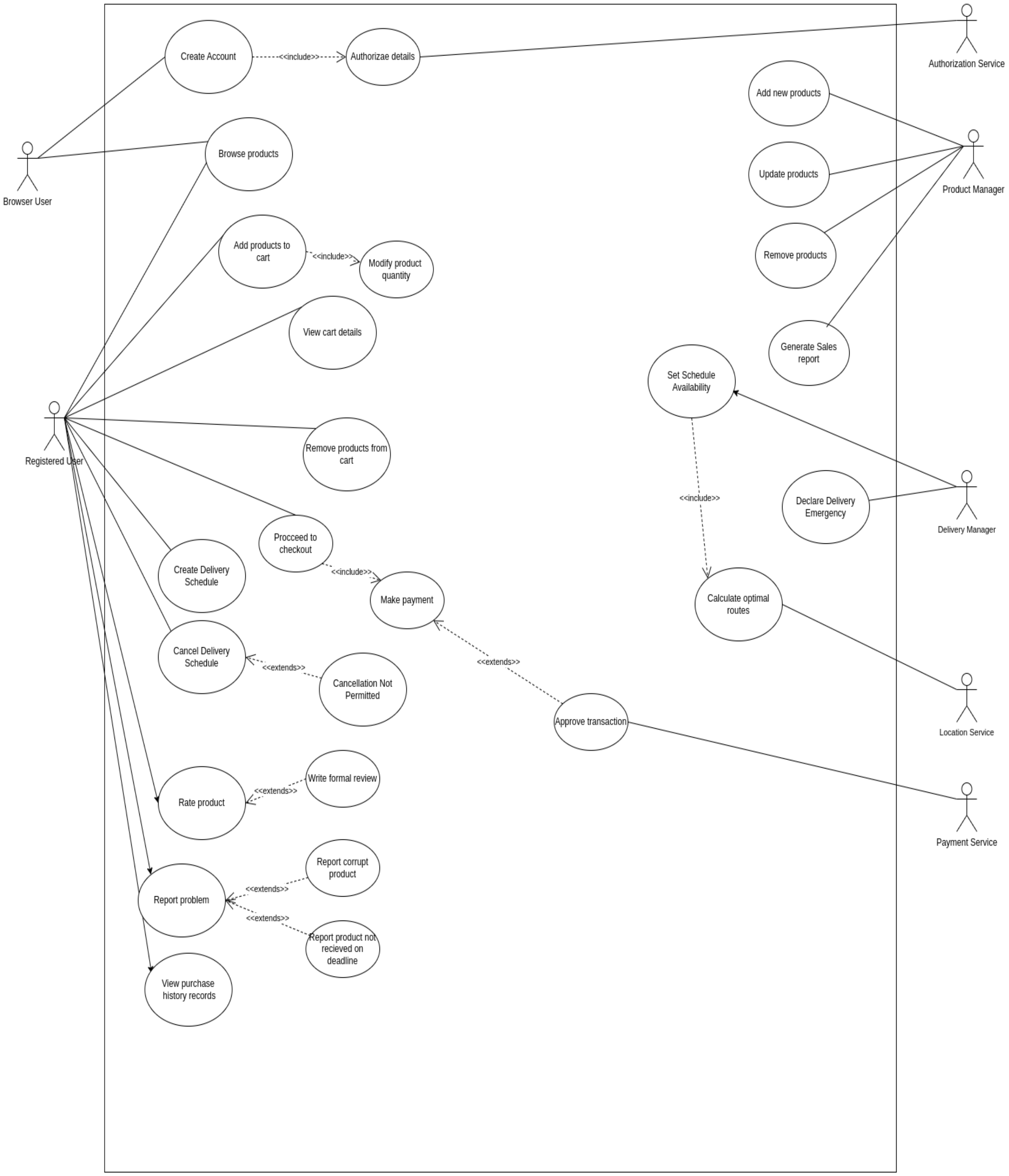


2.7.4 Hamza Najar (Secretary): 1192605



2.7.5 USE CASE DIAGRAM (MODIFIED)

(Lead: Hamza, reviewing: Yazan, Amro, discussion: Osama)



2.7 Description of key USE-CASES:

2.7.1 Osama Quttenh (Product Manager): 1222825

Description table

Section	Content
System Name	E-Commerce System
Use case Title	View Purchase History Records
Description	A Registered User may view their purchase history records from the system's transaction database. The system displays 20 transactions per page in chronological order (newest first), with each record containing the product name (linked to current product page), purchase price (in local currency with 2 decimal places), and order status ("Processing", "Shipped", "Delivered", or "Cancelled"). A visual percentage completion bar indicates delivery progress. Users can navigate through their transaction history using pagination controls, with the ability to filter transactions within a 24-month period. If no previous purchases exist, the system will extend to a redirect workflow, guiding the user to the browsing page.
Actors	Customer, AuthorizationService
Data	- Transaction records (order dates, product details, prices)- Order status information (Processing/Shipped/Delivered/Cancelled)- Product information (names, links to current product pages)- Delivery tracking information (percentage completion)
Stimulus/Trigger	User command issued by Registered User to view purchase history
Pre-conditions	1. The user must be authenticated and logged into the system2. The user must have a registered account in the system
Workflow OR Sequence/Flow of Events	1. The Registered User selects to view purchase history2. The system validates user's authentication status3. If Valid: 3.1 System retrieves last 20 transaction records (sorted by date) 3.2 System formats transaction data (prices, status, delivery progress) to a readable format 3.3 System displays formatted transaction list with navigation controls4. Else, if Invalid: 4.1 System redirects to login page 4.2 Displays authentication required message5. If No Records Found: 5.1 Displays suggestion to browse products

Section	Content
Post-conditions/Response	1. The system has updated the user’s last access timestamp, if successful
Comments	The Registered User must have valid authentication credentials to access their purchase history information. System should maintain appropriate security measures for protecting sensitive transaction data.

2.7.2 Amro Hammad (Techincal Architect): 1201010 (MODIFIED)

Description table

Section	**Content**
Description	Allows users to create a new delivery schedule by providing required details through the system interface. The actor for this use-case is the user.
Sequence/Flow of Events	1. User selects the "Create Delivery Schedule" option from the system interface. 2. The system displays a form for the user to input delivery details, including: *Customer ID* Delivery Date/Time *Location (optional)* 3. *User enters the required details into the form fields.* 4. *The system validates the input fields to ensure:* Delivery Date/Time is in a valid format. *Customer ID exists in the database.* 5. *If all inputs are valid:* The system stores the delivery schedule in the relational database. *Schedule creation is completed within 2 seconds.* 6. *The system displays a success message:* "Delivery schedule successfully created."

Data	**Input Data:** *Customer ID* Delivery Date/Time *Location (optional)* ***Output Data:*** Success Message * Error Messages
Stimulus/Trigger	User selects the "Create Delivery Schedule" option and submits delivery details.
Special Requirements	*The schedule creation process must not exceed 2 seconds.* Success or error messages must be displayed within 1 second. * The system must ensure that no duplicate delivery schedules are created for the same Customer ID and time.
Section	**Content**
Pre-Conditions	*The user must have an active account.* Delivery information must be accessible in the system.
Post-Conditions/Response	**If successful:** *The delivery schedule is stored in the relational database.* The user can view the newly created delivery schedule. **If unsuccessful:** * The system displays appropriate error messages and allows the user to correct their input or try again later.

2.7.3 Yazan Yousef (Programmer): 1191706

Description table

Section	Content
Description	The system allows to create, update, or delete users account information, to ensure accurate details for purchases, orders, and feedback/error reporting

Actors	* User
Trigger	The user start account management by selecting the "Manage Account" option in the system.
Pre-conditions	*The user is registered or ready to be.* If registered, the user is logged in.
Workflow	1. The user selects "Manage Customer Account" from the main menu. 2. The system displays options: Create Account, Update Account, or Delete Account. 3. If the user chooses "Create Account": 3.1. The system prompts for customer name, contact details, purchase preferences, and other optional information 3.2. The system generates a unique Account ID 3.3. The system saves the information and displays a success message 4. If the shop user chooses "Update Account": 4.1. The system retrieves and displays current customer information 4.2. The user modifies the required fields (e.g., contact number, address, purchase preferences) 4.4. The system updates the account with a confirmation message 5. If the shop manager chooses "Delete Account": 5.1. The system prompts to enter the password 5.2. The system confirms if the account is to be deleted 5.3. Upon confirmation, the system removes the account and generates a success message
Alternative Workflow	If mandatory fields (e.g., name, contact details) are missing during account creation, the system prompts: "Please fill in all required fields."
Postconditions	Customer account information is successfully added, updated, or deleted.

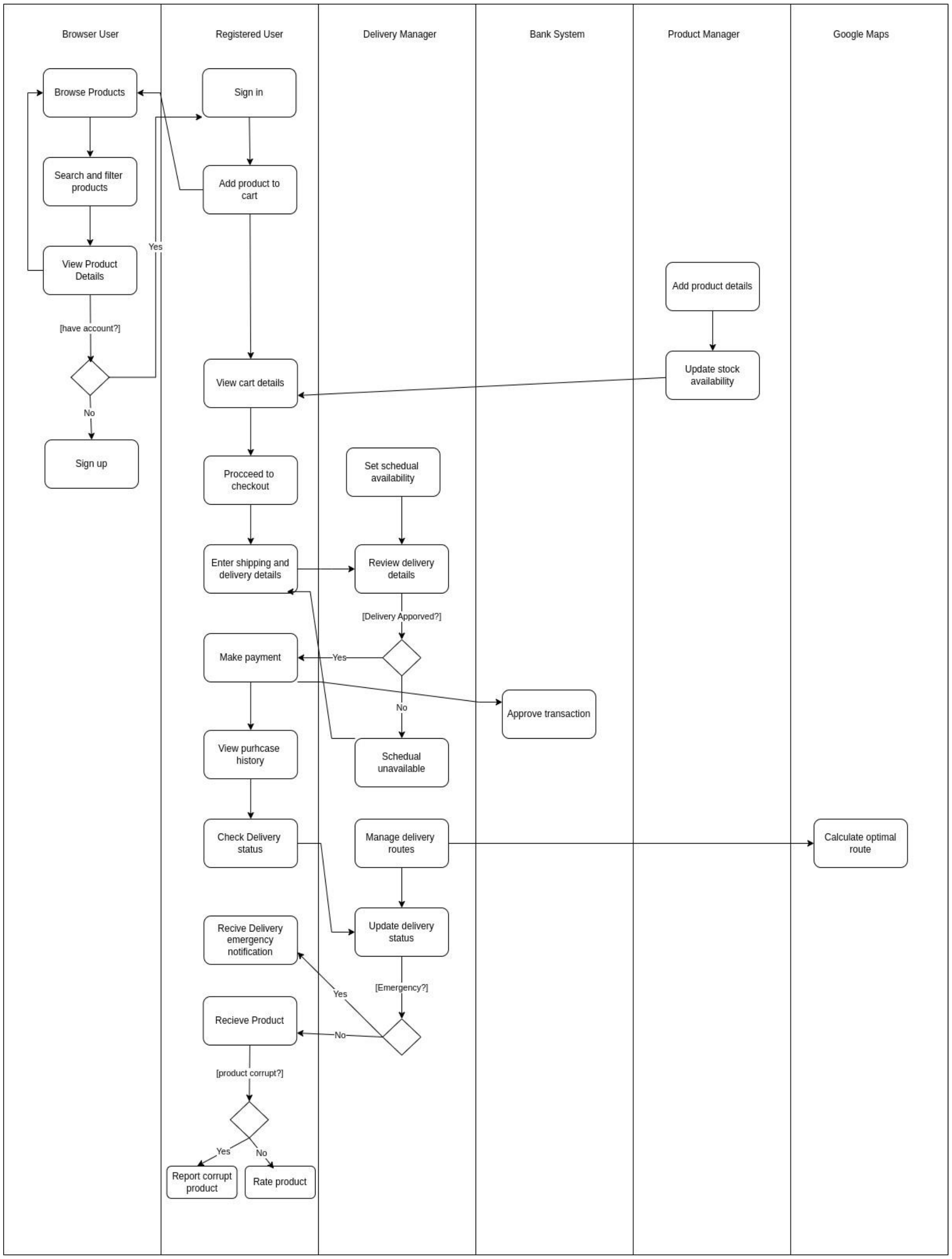
2.7.4 Hamza Najar (Secretary): 1192605

Description table

Section	**Content**
Title	Agriculture Shop Management System
Purpose	To allow users to add, remove, or modify products in their shopping cart
Actors	* User (Primary)
Trigger	User interacts with the shopping cart via the interface
Pre-condition	The user is browsing the website, and products are listed.
Workflow	1. The user selects a product to add to the cart. 2. The system verifies product availability and stock levels. 3. The system adds the product to the cart with the selected quantity.4. The user views the cart to review product details (name, quantity, price, subtotal). 5. The user can modify product quantities in the cart. 6. The system validates the updated quantity against stock levels. 7. The user can remove products from the cart. 8. The system updates the cart accordingly. 9. If any promotional discounts apply, the system calculates and displays them. 10. The user saves the cart for later use or proceeds to checkout.
Section	**Content**
Alternative Workflow	The system notifies the user if stock is insufficient or a selected quantity exceeds availability.
Post-condition	The shopping cart is updated with the latest changes made by the user.

2.8 ACTIVITY Diagram:

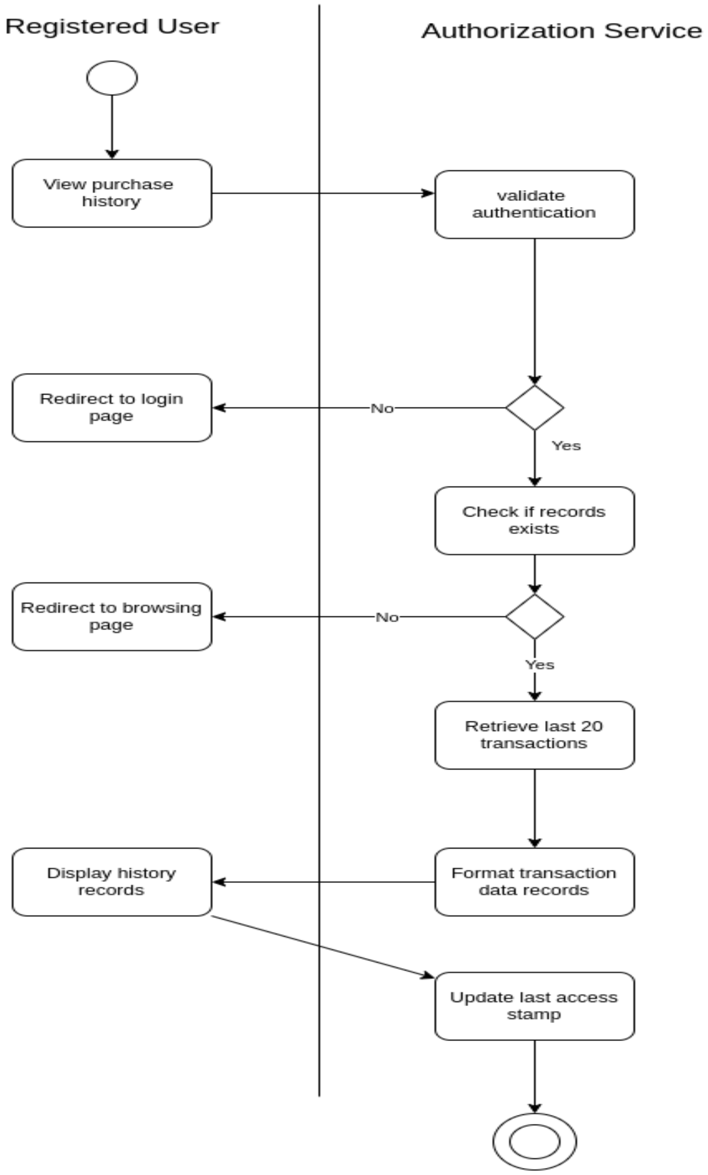
(Lead: Yazan, reviewing: Osama, Amro, discussion: Hamza)



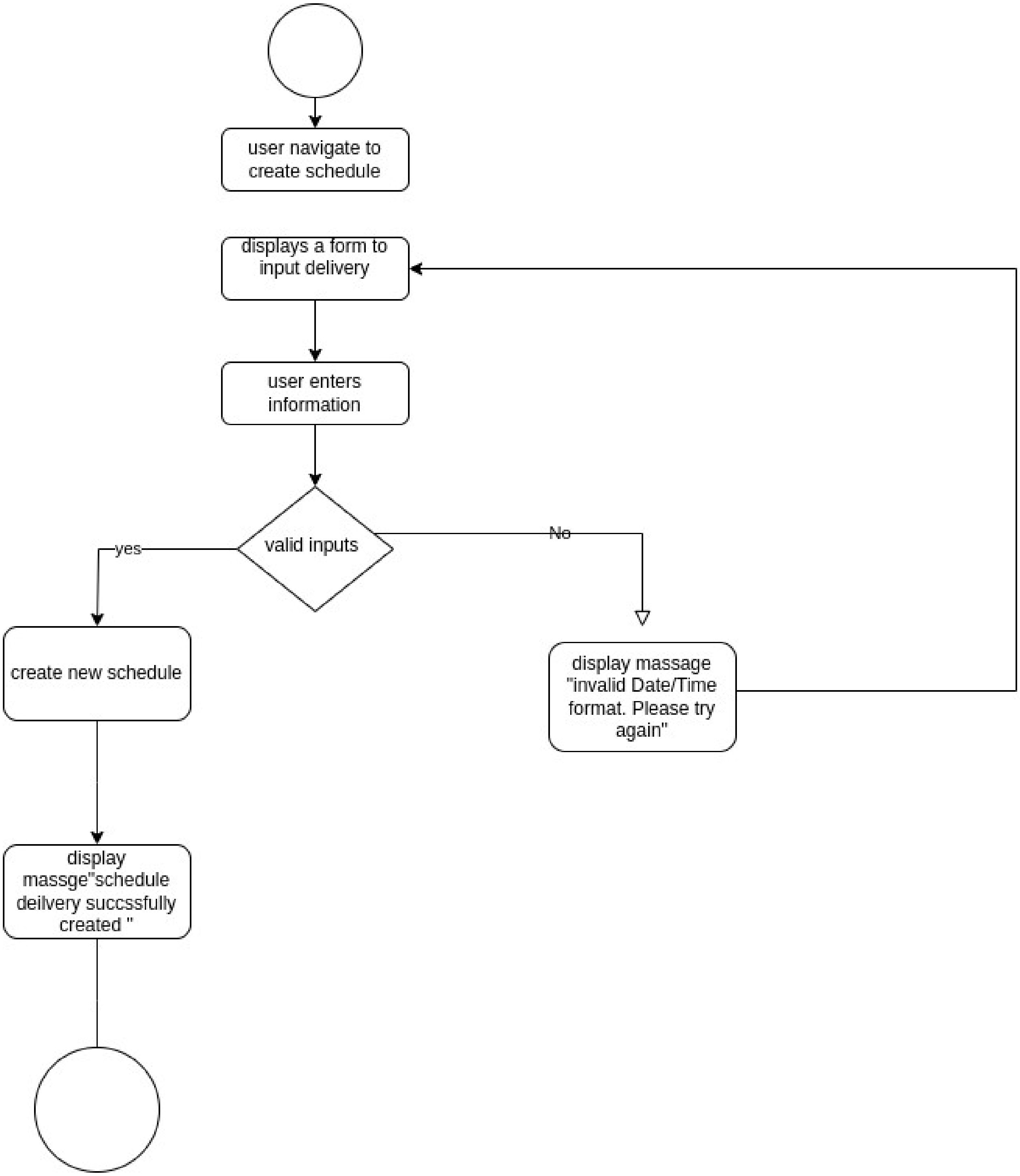
2.9 Instance Activity Diagrams:

: Purchase history

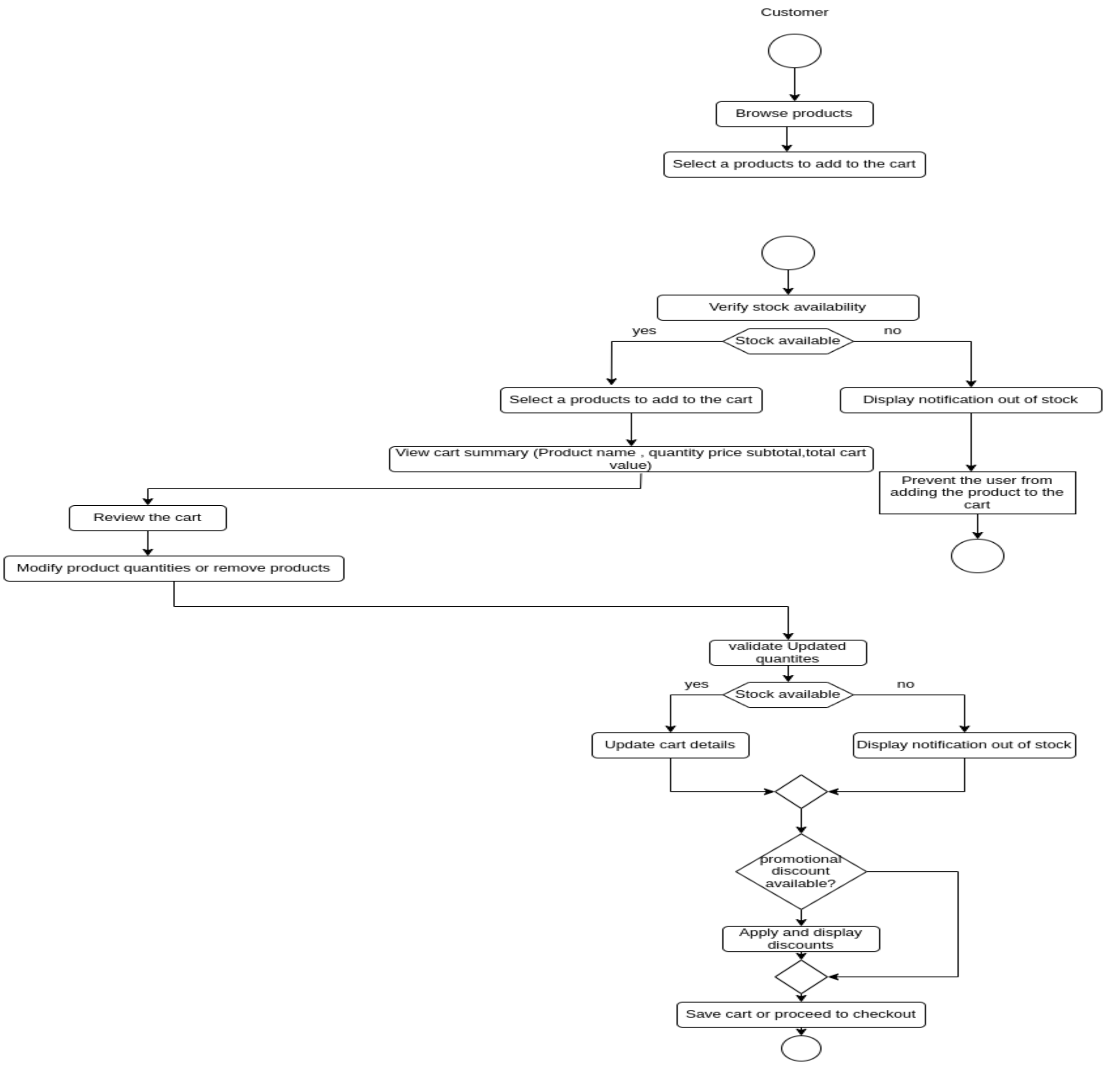
2.9.2 Osama Quttenh :1222825


Registered User
Authorization Service
View purchase history
validate authentication
Redirect to login page
No
Yes
Check if records exists
Redirect to browsing page
No
Yes
Retrieve last 20 transactions
Display history records
Format transaction data records
Update last access stamp

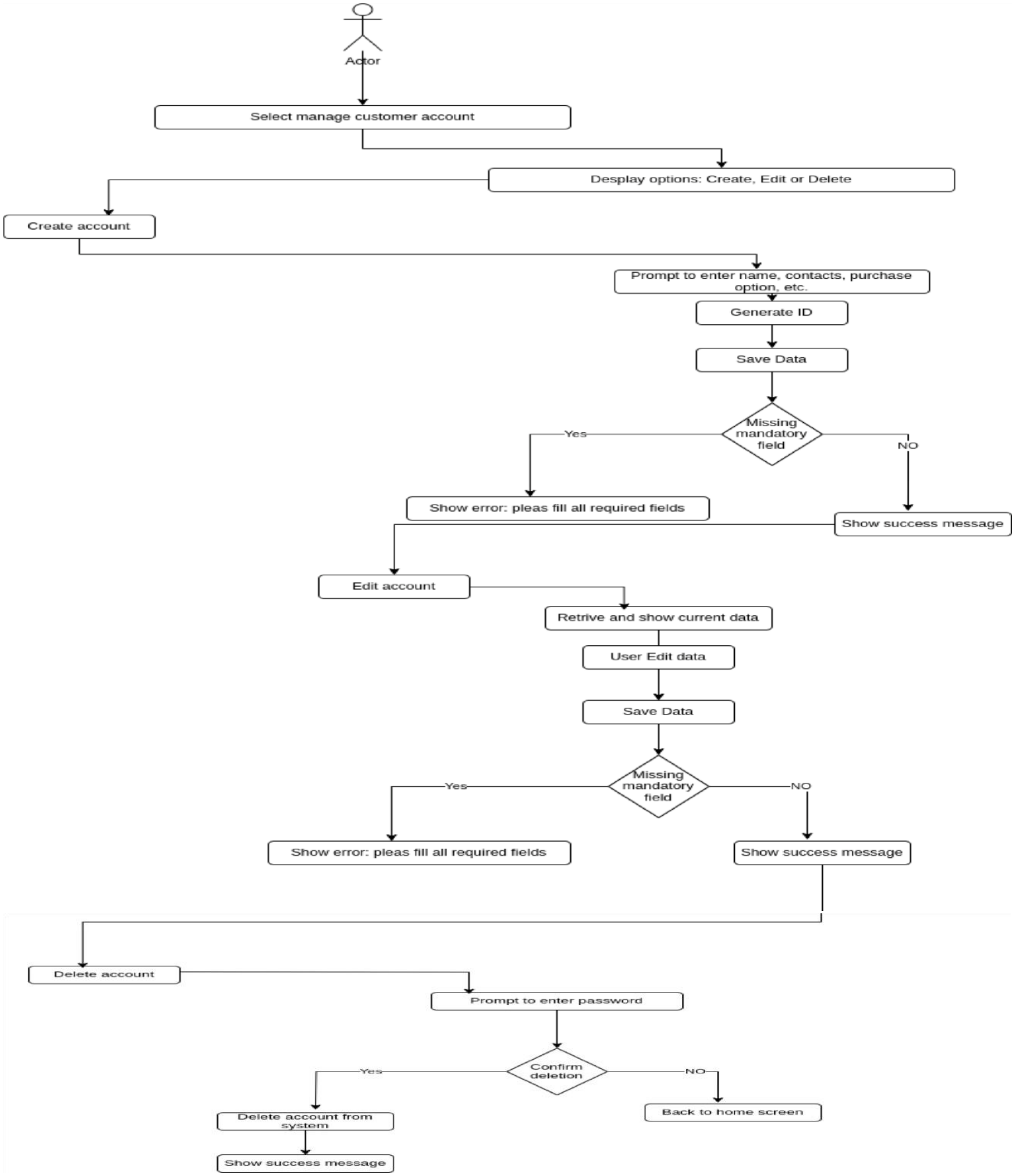
2.9.2 create new schedule (Amro Hammad)


user navigate to create schedule
displays a form to input delivery
user enters information
valid inputs
yes
No
create new schedule
display massage "invalid Date/Time format. Please try again"
display massge"schedule deilvery succssfully created "

2.9.3add, remove, or modify products (Hamza Najar)


Customer
Browse products
Select a products to add to the cart
Verify stock availability
yes
Stock available
no
Select a products to add to the cart
Display notification out of stock
View cart summary (Product name , quantity price subtotal,total cart value)
Prevent the user from adding the product to the cart
Review the cart
Modify product quantities or remove products
validate Updated quantites
yes
Stock available
no
Update cart details
Display notification out of stock
promotional discount available?
Apply and display discounts
Save cart or proceed to checkout

2.9.3 create, update, or delete users account information (Yazan Yousef)

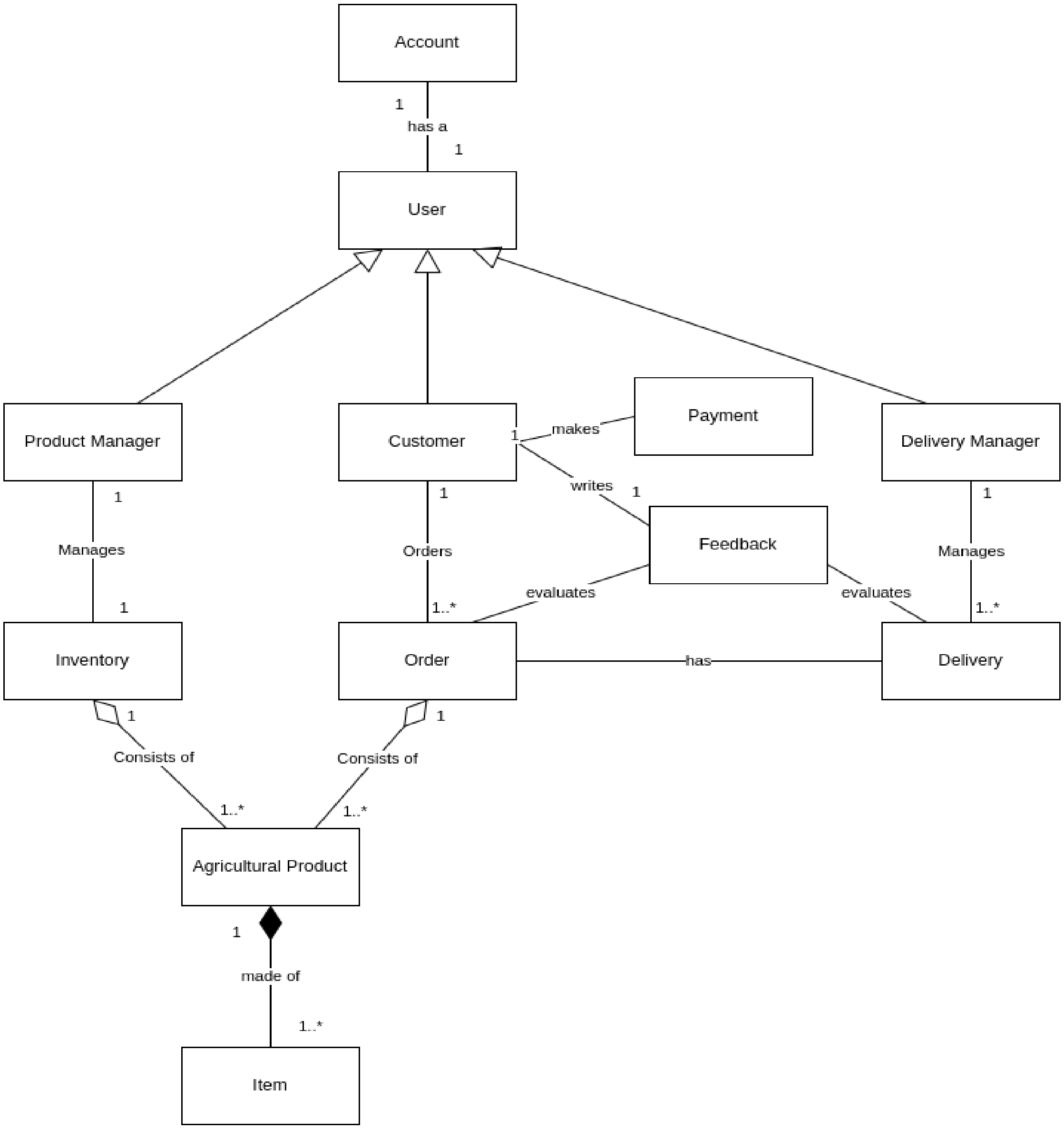


Chapter 3: System Analysis and Modelling

3.1. System CLASS Diagrams:

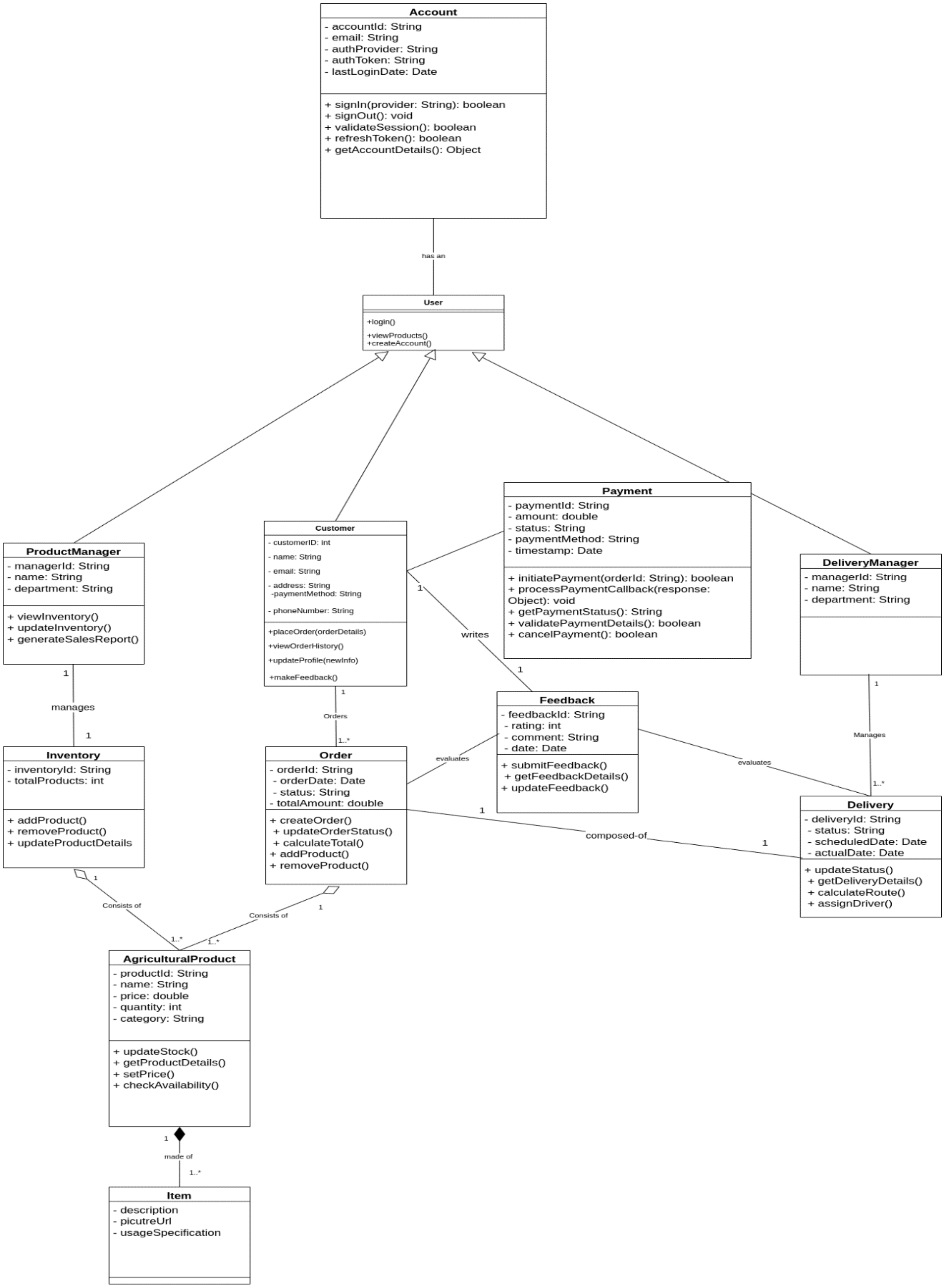
3.1.1. Analysis class model

(Lead: Hamza, reviewing: Osama, Amro, discussion: Osama)



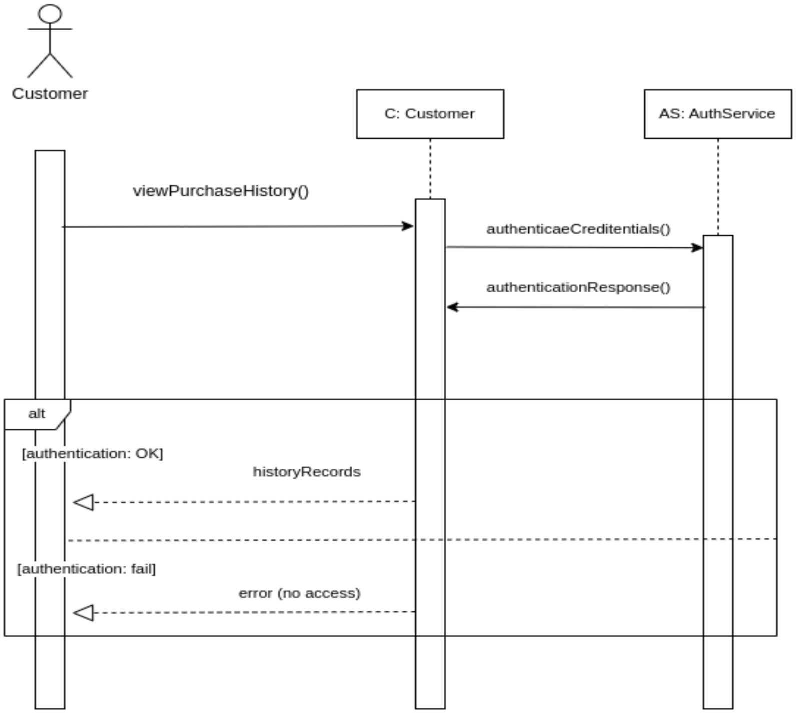
3.1.2. Detailed class model

(Lead: Yazan, reviewing: Hamza, Amro, discussion: Osama)

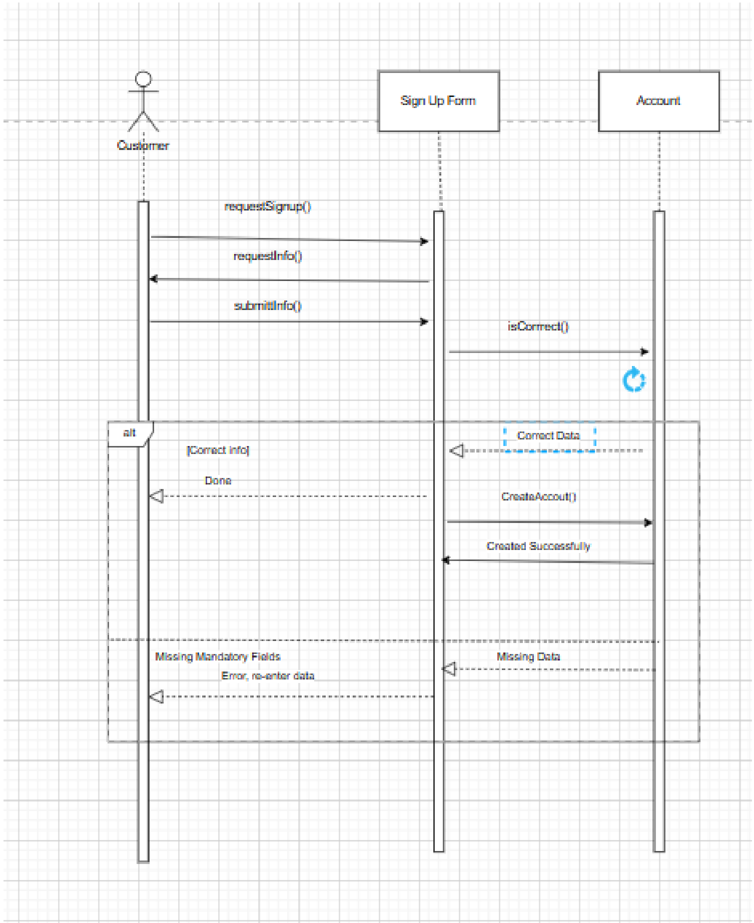


3.2. SEQUENCE Diagrams:

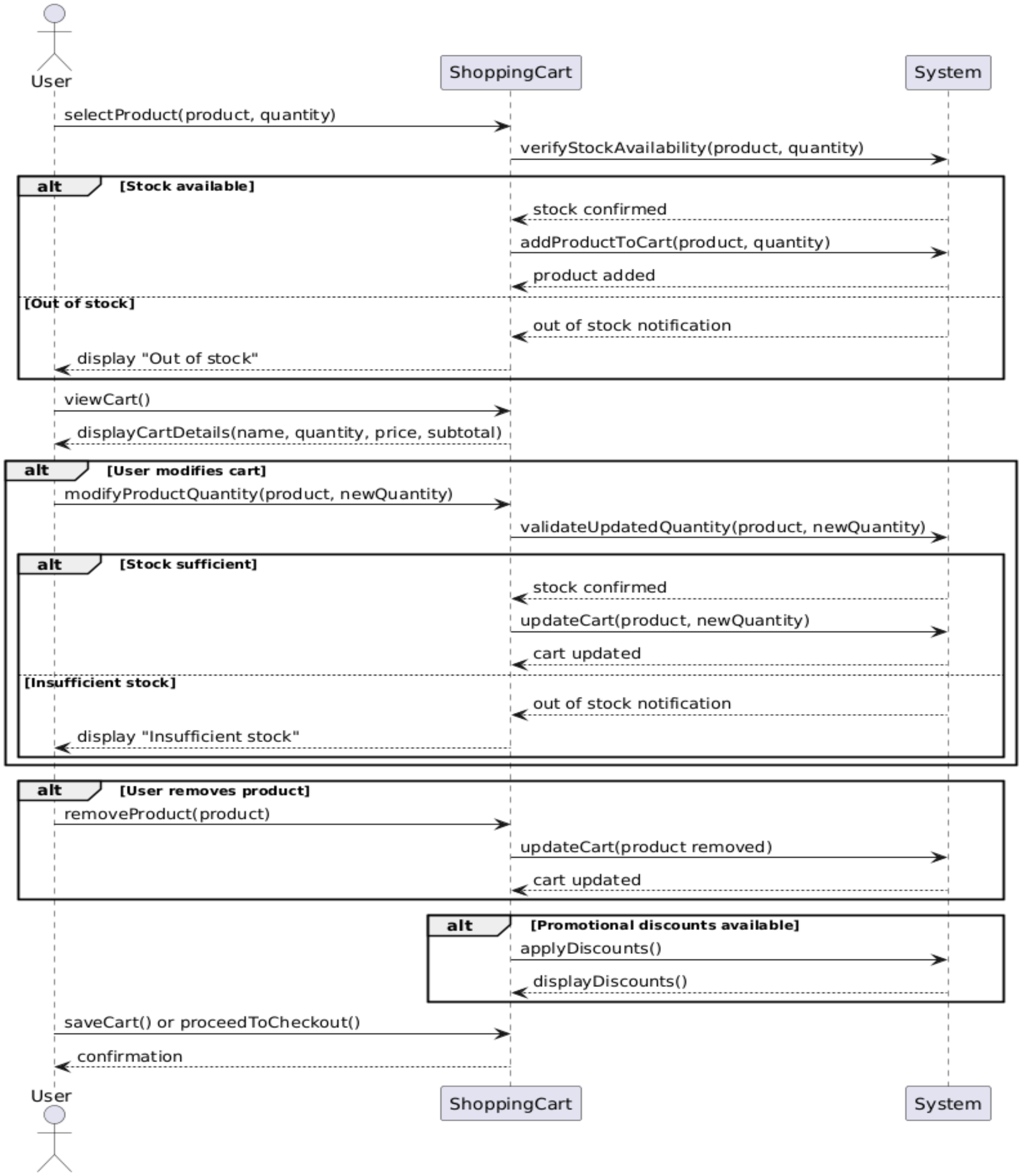
3.2.1 view purchase history (Osama Quttneh)



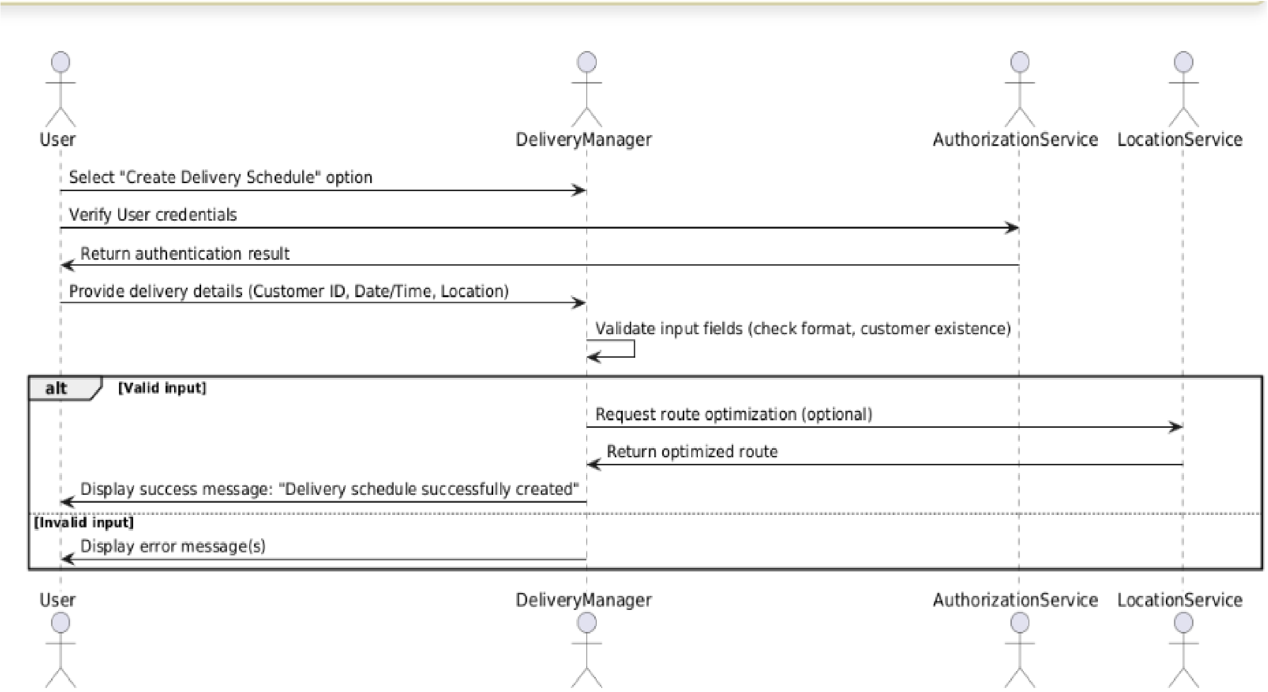
3.2.2 SignUp (Yazan Yousef)



3.2.3 Cart Managment (Hamza Najar)


User
ShoppingCart
System
selectProduct(product, quantity)
verifyStockAvailability(product, quantity)
alt
[Stock available]
stock confirmed
addProductToCart(product, quantity)
product added
[Out of stock]
out of stock notification
display "Out of stock"
viewCart()
displayCartDetails(name, quantity, price, subtotal)
alt
[User modifies cart]
modifyProductQuantity(product, newQuantity)
validateUpdatedQuantity(product, newQuantity)
alt
[Stock sufficient]
stock confirmed
updateCart(product, newQuantity)
cart updated
[Insufficient stock]
out of stock notification
display "Insufficient stock"
alt
[User removes product]
removeProduct(product)
updateCart(product removed)
cart updated
alt
[Promotional discounts available]
applyDiscounts()
displayDiscounts()
saveCart() or proceedToCheckout()
confirmation
User
ShoppingCart
System

3.2.4 create delivery schedule (Amro Hammad)


User
DeliveryManager
AuthorizationService
LocationService
Select "Create Delivery Schedule" option
Verify User credentials
Return authentication result
Provide delivery details (Customer ID, Date/Time, Location)
Validate input fields (check format, customer existence)
alt
[Valid input]
Request route optimization (optional)
Return optimized route
Display success message: "Delivery schedule successfully created"
[Invalid input]
Display error message(s)
User
DeliveryManager
AuthorizationService
LocationService

Chapter 4: System Design and Modelling

4.1. Description of chosen Design Goals : (MODIFIED)

(Lead: Amro, reviewing: Hamza, Osama, discussion: Yazan)

- **General Design Goals:**

  - **Loose Coupling**:
    The system should minimize dependencies between components by ensuring they communicate through well-defined interfaces. Each layer (UI, Business Logic, Infrastructure) should be able to evolve independently without forcing changes in other

layers. For example, changing the UI technology shouldn't require modifications to the business logic. Similarly, changing the Location service provider in the infrastructure layer shouldn't require changes in the middle/Business-logic layer.

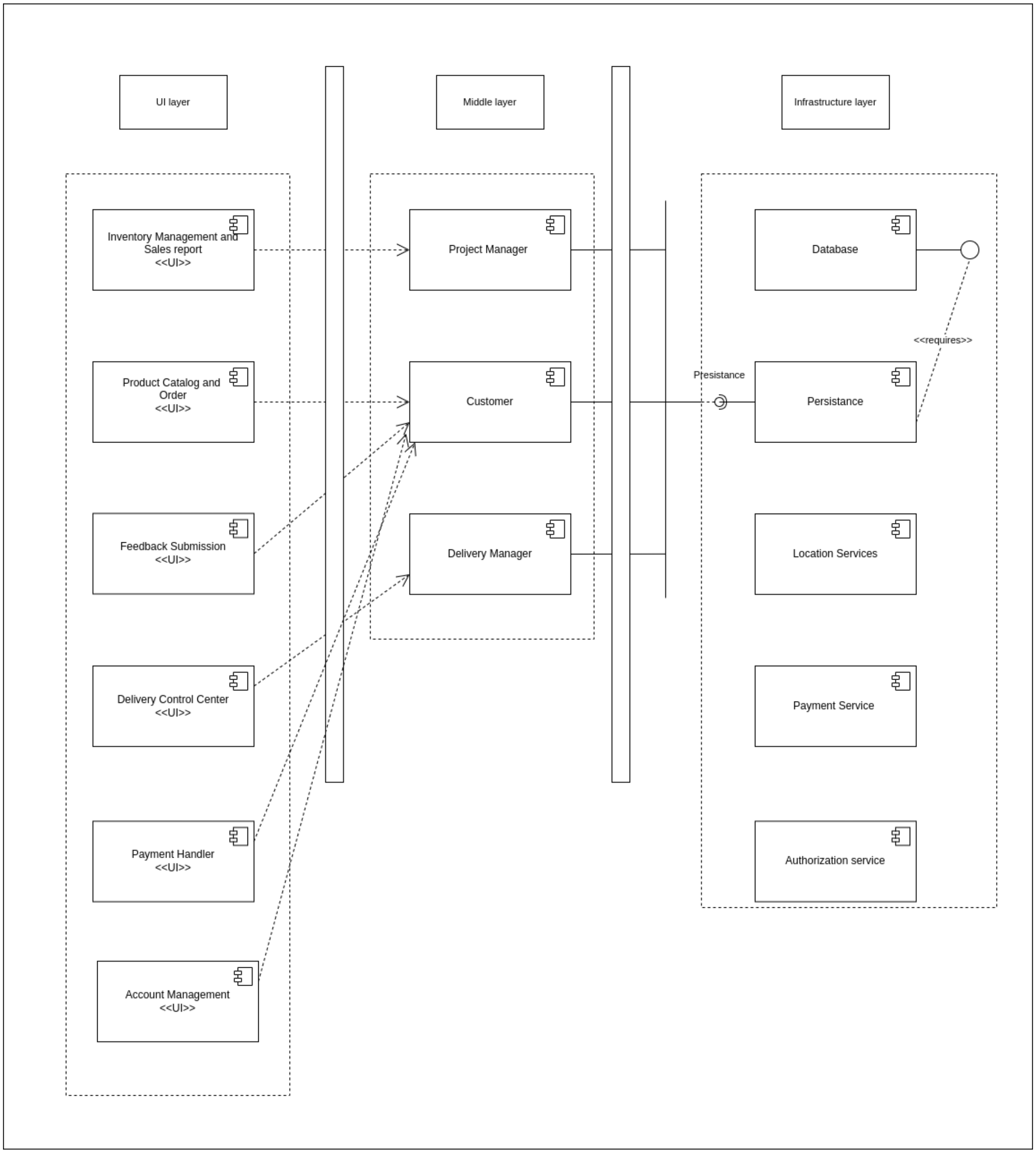
- **High Cohesion**:
  Each component should have a clear, focused responsibility. The UI layer handles only presentation logic and user interaction, the business layer manages core business logic and operations, and the infrastructure layer deals with data persistence and external services. Within each layer, components are grouped by their related functionality (e.g., Product Management, Customer Operations, Delivery Management).

Specific Design Goal:

- **Performance and Reliability**:
  The system must ensure real-time tracking and efficient delivery routing for agricultural products, with response times under 2 seconds for route calculations and delivery status updates, as these are perishable goods that require timely delivery.
- **Usability**:
  The system must be easy to use and quick to learn by any user type (Product Manager, Delivery Manager, Customer) and ensure a fast learning track of the application's core functionalities.

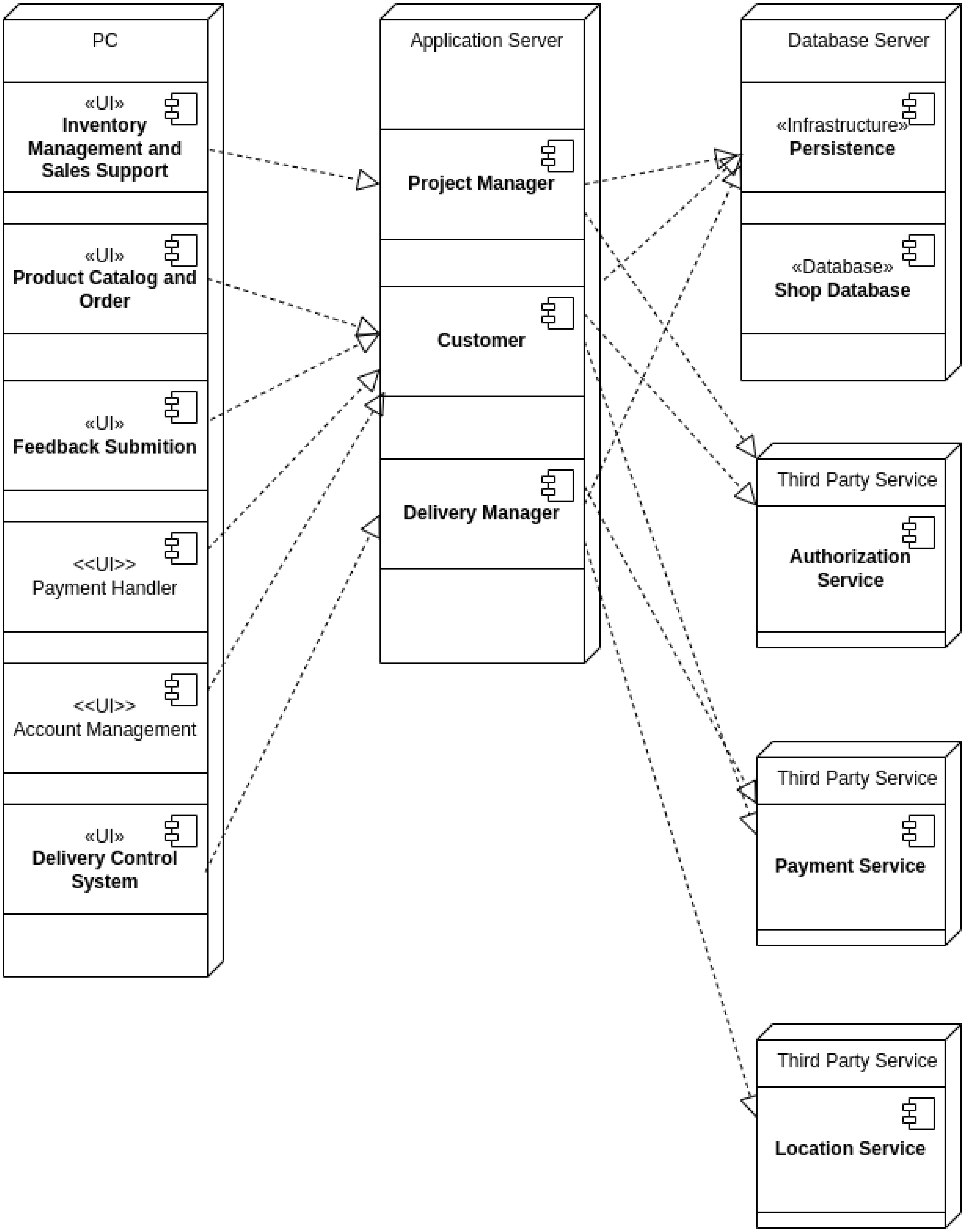
4.3. Overall architecture Diagram: (MODIFIED)

(Lead: Osama, reviewing: Hamza, Amro, discussion: Yazan)


UI layer
Middle layer
Infrastructure layer
Inventory Management and Sales report <<UI>>
Product Catalog and Order <<UI>>
Feedback Submission <<UI>>
Delivery Control Center <<UI>>
Payment Handler <<UI>>
Account Management <<UI>>
Project Manager
Customer
Delivery Manager
Database
Persistance
Location Services
Payment Service
Authorization service
<<requires>>
Presistance

4.4. Deployment Diagram: (MODIFIED)

(Lead: Amro, reviewing: Hamza, Osama, discussion: Yazan)


PC
«UI»
Inventory Management and Sales Support
«UI»
Product Catalog and Order
«UI»
Feedback Submition
<<UI>>
Payment Handler
<<UI>>
Account Management
«UI»
Delivery Control System
Application Server
Project Manager
Customer
Delivery Manager
Database Server
«Infrastructure»
Persistence
«Database»
Shop Database
Third Party Service
Authorization Service
Third Party Service
Payment Service
Third Party Service
Location Service