Software Requirements and Design Document

for

NEW AGE GROCERY STORE

Prepared by Munam Mustafa & Alian Anwar

<CodeFlow Dynamics>

<2-December-2023>

Table of Contents

1. Introduction

1.1 Purpose

The software's objective is to create an object-oriented grocery store management system for a sizable business with operations all over Pakistan. The main objective is to switch from manual to automated management and operations, which will improve customers' in-person and online purchasing experiences. The system's recognized modules include online shopping capabilities, payment processing, inventory control, store and user management, product catalog management, online registration, login functionality, and an administratively-purposed store checkout simulation. The program aims to address every facet of the business operations and encompass the whole gamut of the grocery store management system.

1.2 Product Scope

The software is a grocery store management system designed to automate tasks for a sizable Pakistani corporation. Its objectives are to boost productivity, elevate client satisfaction, maximize inventory control, and foster company expansion. In keeping with the business's strategy, it prioritizes customer satisfaction and technology innovation to ensure operational efficiency and streamline processes.

1.3 Title

New age grocery store management system: Enhancing Efficiency and Customer Experience

1.4 Objectives

1. ***Automation and Efficiency:***
    - *Implement automated processes for streamlined operations.*
    - *Enhance user interface for efficient navigation.*
2. ***Online Shopping Integration:***
    - *Introduce a secure and seamless online shopping experience.*
    - *Cater to evolving customer needs through digital solutions.*
3. ***Inventory Optimization:***
    - *Real-time tracking to prevent overstocking or stockouts.*
    - *Implement efficient inventory management practices.*
4. ***Scalable Architecture:***
    - *Design a flexible system to accommodate new store additions.*
    - *Ensure minimal disruption during branch integration.*

1.5 Problem Statement

The primary goal of this project is to overcome the inherent difficulties and inefficiencies that come with the large grocery store company's current manual operations. User registration, inventory

control, and customer transactions are just a few of the manual procedures that have been shown to be time-consuming, prone to errors, and inadequate to satisfy the changing needs of the market. The business understands that, in order to overcome these obstacles and improve its competitiveness in the grocery retail industry, automation is a critical need.

The manual operations that are in place now contribute to operational inefficiencies, which cause inventory anomalies and delays in customer service. With more customers and retail locations opening up all throughout Pakistan, the current system finds it difficult to keep up ideal inventory levels and deliver a flawless shopping experience. The company's capacity to adjust to market trends and shifting customer preferences is further hampered by the manual nature of duties like inventory tracking and customer administration.

According to feasibility studies, switching to an automated system for managing grocery stores is a wise financial and technological decision. Automation is anticipated to streamline numerous processes and drastically lower operational costs related to human error. Additionally, the potential for higher revenue through the introduction of online shopping and higher customer satisfaction lend support to the project's viability. The project's overall goals are to improve operational effectiveness, lessen the drawbacks of manual workflows, and set up the business for long-term success in the cutthroat grocery retail market.

2. Overall Description

2.1 Product Perspective

The grocery store management system described in this Software Requirements Specification is a new and self-contained product aimed at revolutionizing the large grocery store company's operational processes. It is not a replacement for an existing system, but rather an innovative solution for automating and streamlining manual workflows. This system is envisioned as a critical component that will integrate seamlessly into the company's existing infrastructure, acting as the backbone for managing multiple aspects of the grocery retail business.

The grocery store management system is made to interface with different parts of the larger system, such as online shopping, inventory control, user registration, and administrative simulations. The company's main objective of enhancing customer experiences and switching from manual to automated processes depends on this system. Real-time data exchange and coherent functionality are ensured by the interfaces between various modules, including user registration and inventory management.

To visualize the integration, a simplified diagram showing the main parts of the entire system, their connections, and external interfaces could be useful. By establishing connections with user interfaces, inventory databases, and online shopping platforms, the grocery store management system acts as a central hub. This graphic depiction can help explain how the new system works with the current parts and enhances the overall efficacy and efficiency of the grocery store's operations.

2.2 Product Functions

1. *Add Inventory Items:*
   - *Enable Store Managers to add new items to their store's inventory.*
   - *Include details such as product name, quantity, and pricing.*
2. *Update Inventory Item:*
   - *Allow Store Managers to modify details of existing inventory items.*
   - *Provide options for updating quantities, prices, or other product information.*
3. *View Inventory:*
   - *Enable Store Managers to view the entire inventory of their respective stores.*
   - *Display details such as product names, quantities, and current stock status.*
4. *Manage Stores:*
   - *Provide Admins with the ability to add, remove, and update store information.*
   - *Facilitate assignment of managers to specific stores based on location.*
5. *Manage Product Catalog:*
   - *Allow Admins to add, remove, and update product information in the catalog.*
   - *Ensure changes made in the catalog are reflected in real-time.*
6. *Search and View Inventory Items:*
   - *Enable Store Managers to search for specific products within their store's inventory.*
   - *Allow viewing of detailed information about selected inventory items.*
7. *Add to Cart:*
   - *Enable Customers to add desired items to their shopping cart.*
   - *Update the cart total dynamically with each added item.*
8. *Remove from Cart:*
   - *Allow Customers to remove items from their shopping cart.*
   - *Ensure the cart total is adjusted accordingly.*
9. *View Cart:*
   - *Provide Customers with a summary view of items in their shopping cart.*
   - *Display details such as product names, quantities, and total cost.*
10. *Provide Feedback:*
    - *Allow Customers to submit feedback and ratings for their shopping experience.*
    - *Enable Admins to manage and respond to customer feedback.*
11. *Remove Inventory Items:*
    - *Enable Store Managers to remove items from their store's inventory.*
    - *Implement confirmation mechanisms to prevent accidental removal.*
12. *Process Payment:*
    - *Facilitate payment processing for Customers during the checkout process.*
    - *Integrate secure gateways for various payment options, including COD, cards, and digital wallets.*

2.3 List of Use Cases

1. **Use Case 1: Add New Item**
   - **Actor:** Store Manager
   - **Description:** Store Manager adds a new item to the store's inventory, providing details such as product name, quantity, and pricing.
2. **Use Case 2: Modify Inventory Item**
   - **Actor:** Store Manager

- **Description:** Store Manager updates details of an existing inventory item, such as modifying quantities, prices, or other product information.

3. **Use Case 3: View Store Inventory**
    - **Actor:** Store Manager
    - **Description:** Store Manager views the entire inventory of their store, including product names, quantities, and current stock status.
4. **Use Case 4: Update Store Information**
    - **Actor:** Admin
    - **Description:** Admin updates store information, including adding, removing, or modifying store details.
5. **Use Case 5: Update Product Information**
    - **Actor:** Admin
    - **Description:** Admin updates product information in the catalog, ensuring real-time reflection of changes.
6. **Use Case 6: Search for Product**
    - **Actor:** Store Manager
    - **Description:** Store Manager searches for a specific product within the store's inventory and views detailed information about it.
7. **Use Case 7: Add Item to Cart**
    - **Actor:** Customer
    - **Description:** Customer adds a desired item to the shopping cart, specifying quantity if necessary.
8. **Use Case 8: Remove Item from Cart**
    - **Actor:** Customer
    - **Description:** Customer removes an item from the shopping cart, adjusting the cart total accordingly.
9. **Use Case 9: View Shopping Cart**
    - **Actor:** Customer
    - **Description:** Customer views a summary of items in their shopping cart, including product names, quantities, and total cost.
10. **Use Case 10: Submit Feedback**
    - **Actor:** Customer
    - **Description:** Customer provides feedback and ratings for their shopping experience.
11. **Use Case 11: Remove Item from Inventory**
    - **Actor:** Store Manager
    - **Description:** Store Manager removes an item from the store's inventory, with confirmation mechanisms to prevent accidental removal.
12. **Use Case 12: Complete Checkout**
    - **Actor:** Customer
    - **Description:** Customer completes the checkout process, selecting a payment method and finalizing the transaction.

2.4 Extended Use Cases

- **Use Case 4: Update Inventory Item**

a. Use Case Name: Update Inventory	**f. Pre-conditions:** 1. Store Manager is logged in. 2. Store Manager has the privilege to manage inventory.
b. Scope the system under design: Grocery Store System	**g. Post Conditions:** Inventory is updated with the new quantities.
c. Level: Subsystem	**H. Main success scenario:** 1. Store Manager selects the "Update Inventory" option. 2. Store Manager selects items to update and provides new quantities. 3. The system validates the changes. 4. Inventory is updated with the new quantities.
d. Primary Actor: Store Manager	**i. Extensions:** If the store manager lacks the privilege to manage inventory, the system displays an error message.
e. Stakeholder and interests: **Store Manager:** Wants to update the quantity of items in the store's inventory.	

By Munam Mustafa

- **Use Case 5: View Inventory**

a. **Use Case Name:** View Inventory	**f. Pre-conditions:** 1. Store Manager is logged in.
b. **Scope the system under design:** Grocery Store System	**g. Post Conditions:** Inventory details are displayed.
c. **Level:** Subsystem	**H. Main success scenario:** 1. Store Manager selects the "View Inventory" option. 2. The system displays a list of items in the store's inventory along with their quantities.
d. **Primary Actor:** Store Manager	**i. Extensions:** If the store manager lacks the privilege to view inventory, the system displays an error message.
e. **Stakeholder and interests:** **Store Manager:** Wants to view the current state of the store's inventory.	

By Munam Mustafa

- **Use Case 6: Manage Stores and Users**

a. **Use Case Name:** Manage Stores and Users	**f. Pre-conditions:** 1. Admin is logged in.
b. **Scope the system under design:** Grocery Store System	**g. Post Conditions:** Stores and user accounts are managed.
c. **Level:** Subsystem	**H. Main success scenario:** 1. Admin selects the "Manage Stores and Users" option. 2. Admin can add new stores, remove stores, and manage user accounts.
d. **Primary Actor:** Admin	**i. Extensions:** If the admin lacks the privilege to perform these actions, the system displays an error message.
e. **Stakeholder and interests:** **Admin:** Wants to add/remove stores and manage user accounts.	

By Munam Mustafa

- **Use Case 7: Manage Product Catalog**

a. **Use Case Name:** Manage Product Catalog	**f. Pre-conditions:** 1. Admin is logged in.
b. **Scope the system under design:** Grocery Store System	**g. Post Conditions:** Product catalog is updated.
c. **Level:** Subsystem	**H. Main success scenario:** 1. Admin selects the "Manage Product Catalog" option. 2. Admin can add new products, remove products, and update product details.
d. **Primary Actor:** Admin	**i. Extensions:** If the admin lacks the privilege to manage the product catalog, the system displays an error message.
e. **Stakeholder and interests:** **Admin:** Wants to add/remove/update products in the product catalog.	

By Munam Mustafa

- **Use Case 8: Search and View Inventory Items**

a. **Use Case Name:** Search and View Inventory Items	**f. Pre-conditions:** 1. Store Manager is logged in.
b. **Scope the system under design:** Grocery Store System	**g. Post Conditions:** Inventory items are searched and viewed.
c. **Level:** Subsystem	**H. Main success scenario:** 1. Store Manager selects the "Search and View Inventory Items" option. 2. Store Manager can search for specific items and view inventory details.
d. **Primary Actor:** Store Manager	**i. Extensions:** If the store manager lacks the privilege to view inventory, the system displays an error message.
e. **Stakeholder and interests:** **Store Manager:** Wants to search and view inventory items.	

By Munam Mustafa

- **Use Case 9: Add to Cart**

a. **Use Case Name:** Add to Cart	**f. Pre-conditions:** 1. Customer is logged in.
b. **Scope the system under design:** Grocery Store System	**g. Post Conditions:** Items are added to the shopping cart
c. **Level:** Subsystem	**H. Main success scenario:** 1. Customer browses the product catalog. 2. Customer selects items and adds them to the cart. 3. Cart total is updated.
d. **Primary Actor:** Customer.	**i. Extensions:** If the customer is not logged in, the system prompts them to log in.
e. **Stakeholder and interests:** **Customer:** Wants to add items to the shopping cart.	

By Munam Mustafa

- **Use Case 10: Remove from Cart**

a. **Use Case Name:** Remove from Cart	**f. Pre-conditions:** 1. Customer is logged in.
b. **Scope the system under design:** Grocery Store System	**g. Post Conditions:** Items are removed from the shopping cart.
c. **Level:** Subsystem	**H. Main success scenario:** 1. Customer selects the "View Cart" option. 2. Customer selects items to remove from the cart. 3. The system updates the cart by removing the selected items.
d. **Primary Actor:** Customer.	**i. Extensions:** If the customer is not logged in or the cart is empty, the system provides appropriate feedback.
e. **Stakeholder and interests:** **Customer:** Wants to remove items from the shopping cart.	

By Munam Mustafa

- **Use Case 11: View Cart**

a. **Use Case Name:** View Cart	**f. Pre-conditions:** 1. Customer is logged in. 2. Customer has items in the cart.
b. **Scope the system under design:** Grocery Store System	**g. Post Conditions:** Cart contents are displayed.
c. **Level:** Subsystem	**H. Main success scenario:** 1. Customer selects the "View Cart" option. 2. The system displays a list of items in the cart, along with their quantities and total cost.
d. **Primary Actor:** Customer.	**i. Extensions:** If the customer is not logged in or the cart is empty, the system provides appropriate feedback.
e. **Stakeholder and interests:** **Customer:** Wants to view the items in their shopping cart.	

By Alian Anwar

- **Use Case 13: Provide Feedback**

a. **Use Case Name:** Provide Feedback	**f. Pre-conditions:** 1. Customer is logged in.
b. **Scope the system under design:** Grocery Store System	**g. Post Conditions:** Feedback are submitted.
c. **Level:** Subsystem	**H. Main success scenario:** 1. Customer selects the "Provide Feedback/Rating" option during checkout. 2. Customer provides feedback and a rating for the store. 3. Feedback is submitted successfully.
d. **Primary Actor:** Customer.	**i. Extensions:** If the customer is not logged in, the system prompts them to log in.
e. **Stakeholder and interests:** **Customer:** Wants to provide feedback and ratings for a store.	

By Alian Anwar

- **Use Case 14: Remove Inventory Items**

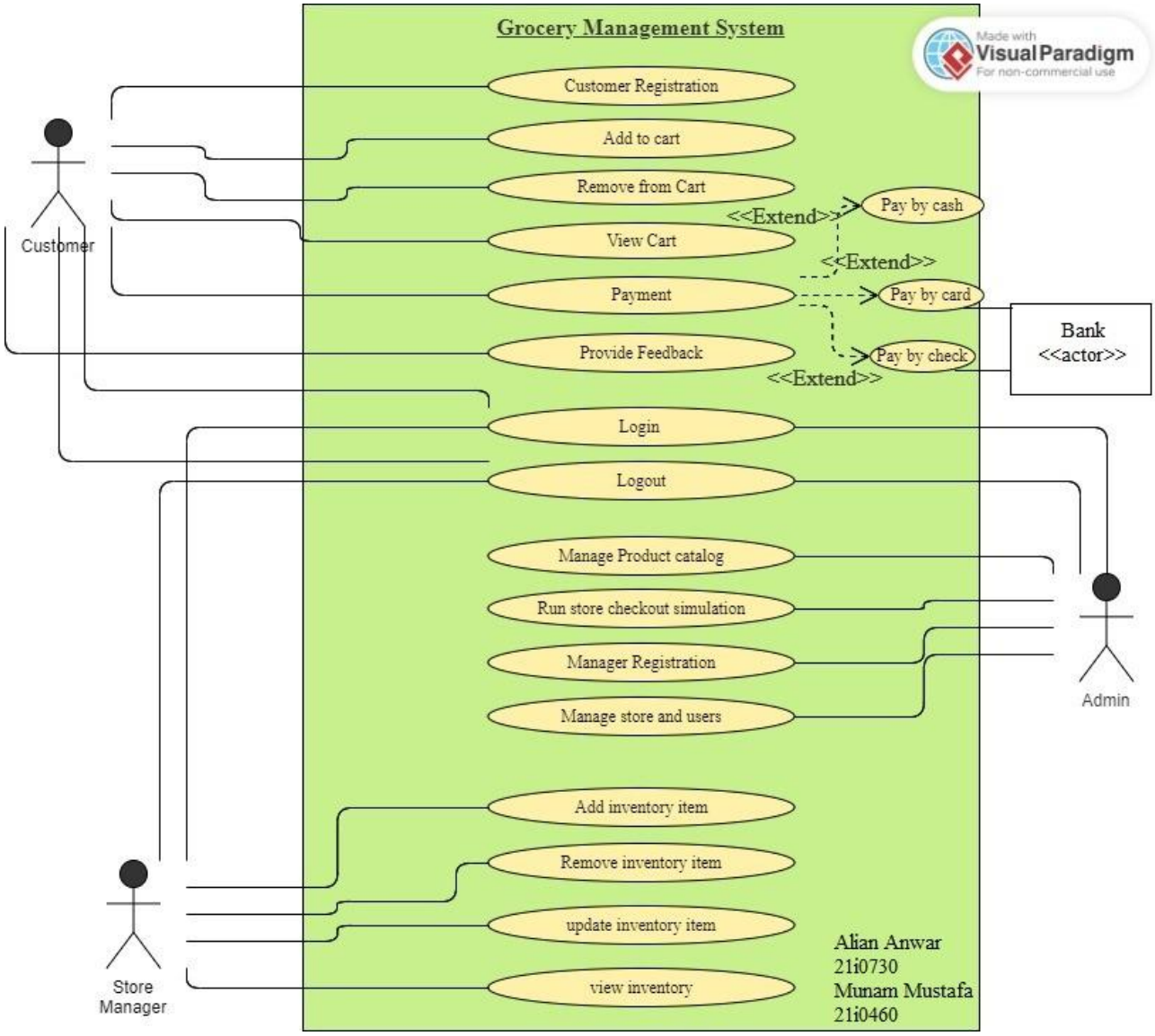
a. Use Case Name: Remove Inventory Items	**f. Pre-conditions:** 1. Store Manager is logged in. 2. Store Manager has the privilege to manage inventory.
b. Scope the system under design: Grocery Store System	**g. Post Conditions:** Items are removed from the store's inventory.
c. Level: Subsystem	**H. Main success scenario:** 1. Store Manager selects the "Remove Inventory Items" option. 2. Store Manager selects items to be removed. 3. The system confirms the removal. 4. Items are removed from the inventory.
d. Primary Actor: Store Manager	**i. Extensions:** If the store manager lacks the privilege to manage inventory, the system displays an error message.
e. Stakeholder and interests: **Store Manager:** Wants to remove items from the store's inventory.	

By Alian Anwar

- **Use Case 15: Process Payment**

a. Use Case Name: Process Payment	**f. Pre-conditions:** 1. Customer is logged in. 2. Customer has items in the cart.
B. Scope the system under design: Grocery Store System	**g. Post Conditions:** Payment is processed, and inventory is updated.
c. Level: Subsystem	**H. Main success scenario:** 1. Customer selects the "Checkout" option from the shopping cart. 2. The system displays the finalized cart with the total amount to be paid. 3. Customer chooses a payment method (e.g., COD, debit/credit card). 4. The system calculates any applicable extra charges based on the payment method and location. 5. Customer confirms the payment. 6. The system processes the payment and updates the inventory to reflect the purchased items. 7. Customer receives a payment confirmation.

2.5 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

1. **Inventory Management:**
   - **Transaction Processing Time:** Add, update, or remove inventory items should be processed in real-time, with changes reflected in the system within 1 second to maintain accurate inventory levels.
   - **Search Response Time:** The system should provide search results for inventory items within 3 seconds to expedite store manager operations.

2. **Online Shopping and Checkout:**
    - **Cart Update Time:** When customers add or remove items from their cart, the system should update the cart total in real-time, ensuring a responsive and dynamic shopping experience.
    - **Checkout Processing Time:** The checkout process, including payment processing, should be completed within 5 seconds to prevent customer frustration and delays.
3. **Feedback and Rating Submission:**
    - **Submission Response Time:** The system should process customer feedback and ratings submissions within 3 seconds to encourage user participation and engagement.

3.2 Safety Requirements

1. **Inventory Management:**
    - **Access Control:** Implement role-based access control to restrict unauthorized users from making changes to the inventory, preventing potential data manipulation or loss.
    - **Data Integrity:** Employ data validation mechanisms to ensure accurate and secure inventory management, minimizing the risk of errors or discrepancies.
2. **Online Shopping and Checkout:**
    - **Secure Payment Processing:** Adhere to industry-standard encryption protocols and secure payment gateways to protect customer financial information during transactions.
    - **Transaction Verification:** Implement validation checks during checkout to prevent fraudulent transactions and protect customers from unauthorized payments.
3. **Feedback and Rating Submission:**
    - **Content Moderation:** Implement content moderation mechanisms to filter inappropriate or harmful feedback, ensuring a safe and positive environment for users.

3.3 Security Requirements

1. **Access Control:**
    - **Role-Based Access Control:** Implement RBAC to restrict access to specific modules and functionalities based on user roles, preventing unauthorized actions and data exposure.
    - **Session Management:** Implement secure session management to protect against session hijacking and unauthorized access to user accounts.
2. **Data Integrity:**
    - **Checksums and Hashing:** Utilize checksums and hashing algorithms to verify the integrity of data, protecting against data tampering and ensuring the reliability of information stored in the system.
3. **Payment Processing Security:**

- **PCI DSS Compliance:** Adhere to Payment Card Industry Data Security Standard requirements to ensure secure handling of payment card information during transactions.
- **Secure Payment Gateways:** Use trusted and secure payment gateways that comply with industry standards, providing a secure environment for financial transactions.

3.4 Software Quality Attributes

1. ***Usability:***
   - *Interface satisfaction score of 4 out of 5.*
2. ***Reliability:***
   - *99.9% system uptime.*
   - *Downtime less than 30 minutes per month.*
3. ***Maintainability:***
   - *Code maintainability score of at least 80%.*
4. ***Performance Efficiency:***
   - *Response times within 1-2 seconds.*
   - *Minimum throughput of 1000 transactions per minute.*
5. ***Scalability:***
   - *Handle a 20% increase in user load without degradation.*
   - *Support addition of 50 new stores without a major overhaul.*

3.5 Business Rules

1. ***User Roles and Permissions:***
   - ***Rule:*** *Only Admins can register and assign managers to specific stores.*
   - ***Implication:*** *Implement role-based access control to enforce user roles and permissions.*
2. ***Customer Registration:***
   - ***Rule:*** *A new customer must provide a valid 13-digit CNIC for registration.*
   - ***Implication:*** *Implement CNIC validation during the customer registration process.*
3. ***Password Requirements:***
   - ***Rule:*** *Customer passwords must be 9 characters long with at least one uppercase letter and one numeric digit.*
   - ***Implication:*** *Implement password validation to ensure compliance with the specified requirements.*
4. ***Unique Customer Accounts:***
   - ***Rule:*** *Customer accounts cannot be repeated; each CNIC should have a unique account.*
   - ***Implication:*** *Implement a mechanism to check for existing customer accounts during registration.*
5. ***Manager Registration:***
   - ***Rule:*** *Only Admins can register new managers and assign stores based on location.*
   - ***Implication:*** *Implement a controlled manager registration process accessible only by Admins.*

3.6 Operating Environment

1. ***Hardware Platform:***
   - ***Requirements:*** *The software is designed to operate on standard computing hardware commonly available in the market.*
   - ***Examples:*** *PCs, laptops, and servers with configurations meeting or exceeding industry standards.*
2. ***Operating System:***
   - ***Requirements:*** *The software is compatible with modern operating systems.*
   - ***Examples:*** *Windows 10 or later, macOS Big Sur or later, and Linux distributions such as Ubuntu*
3. ***Web Browsers:***
   - ***Requirements:*** *The system should be accessible through widely used web browsers.*
   - ***Examples:*** *Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge.*

4. *Database System:*
   - ***Requirements:*** *The software relies on a compatible database system for data storage.*
   - ***Examples:*** *MySQL, PostgreSQL, or any other relational database management system supporting SQL.*

5. *Programming Languages:*
   - *Requirements: The system is developed using standard programming languages.*
   - *Examples: Python, Java, JavaScript.*

3.7 User Interfaces

1. ***Registration Interface:***

- *Components:*
  -> User input fields for CNIC, password, gender, phone number, and address.
  -> Password confirmation and error handling mechanisms.
- *Standards:*
  -> Follow industry best practices for password strength and validation.
  -> Provide clear error messages for user guidance.

2. ***Login Interface:***

- *Components:*
  -> Username and password fields for Admins, Managers, and Customers.
  -> Login button and links for password recovery.
- *Standards:*
  -> Implement secure password hashing for user authentication.
  -> Display helpful error messages for login failures.

3. ***Home Screen:***

- *Components:*
  -> Dynamic home screen based on user type (Admin, Manager, Customer).
  -> Intuitive menus for navigation to different modules.

- *Standards:*
  -> Maintain consistency in menu layout across user roles.
  -> Use clear icons and labels for menu items.

4. ***Store and User Management Interface (Admin):***

- *Components:*
  -> Tables for displaying and managing stores and users.
  -> Buttons for adding, updating, and removing stores and users.
- *Standards:*
  -> Provide confirmation dialogs for critical actions.
  -> Follow Create, Read, Update, Delete conventions.

5. ***Product Catalog Management Interface (Admin):***

- *Components:*
  -> Interactive product catalog with categories and subcategories.
  -> Buttons for adding, updating, and removing products.
- *Standards:*
  -> Use clear visual cues for product categories.
  -> Include tooltips for product details.

6. ***Inventory Management Interface (Manager):***

- *Components:*
  -> Inventory tables for the manager's assigned store.
  -> Add, update, and remove buttons for inventory items.
- *Standards:*
  -> Implement real-time updates in inventory tables.
  -> Include search and filter options for inventory items.

7. ***Online Shopping Interface (Customer):***

- *Components:*
  -> Product catalog for browsing and selecting items.
  -> Shopping cart for managing selected items.
  -> Checkout and payment options.
- *Standards:*
  -> Clearly display product details and prices.
  -> Provide a step-by-step checkout process.

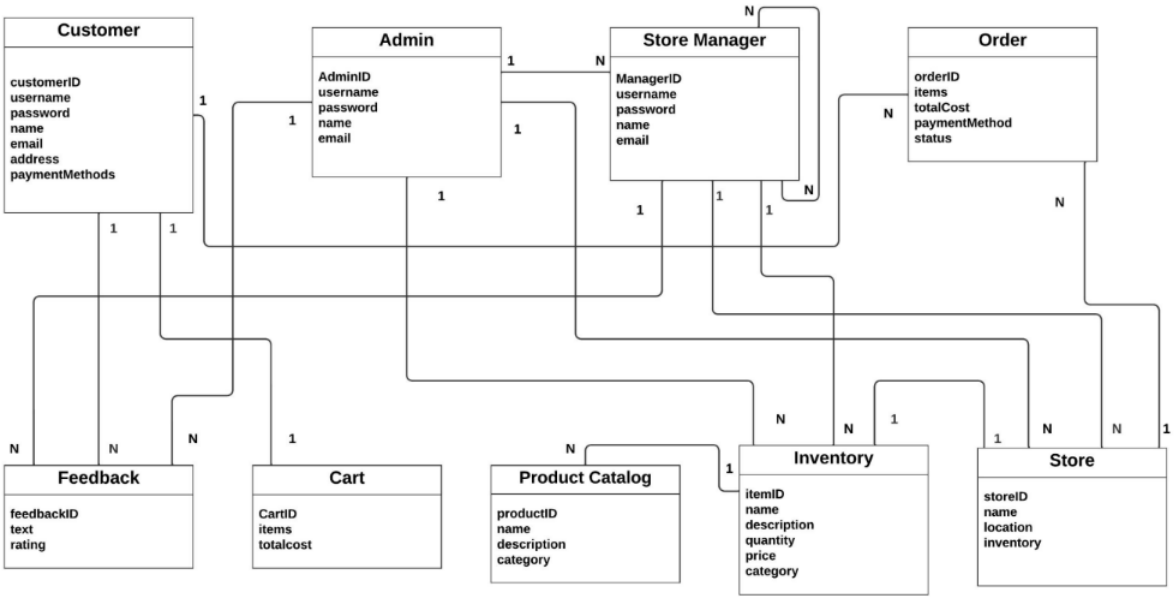
8. ***Feedback and Rating Interface (Customer)****:*

- *Components:*
  -> Form for submitting feedback and ratings against a store.
  -> View and manage past feedback submissions.
- *Standards:*
  -> Set character limits for feedback to maintain brevity.
  -> Allow users to edit or delete their feedback within a specified timeframe.

9. ***Payment Interface:***

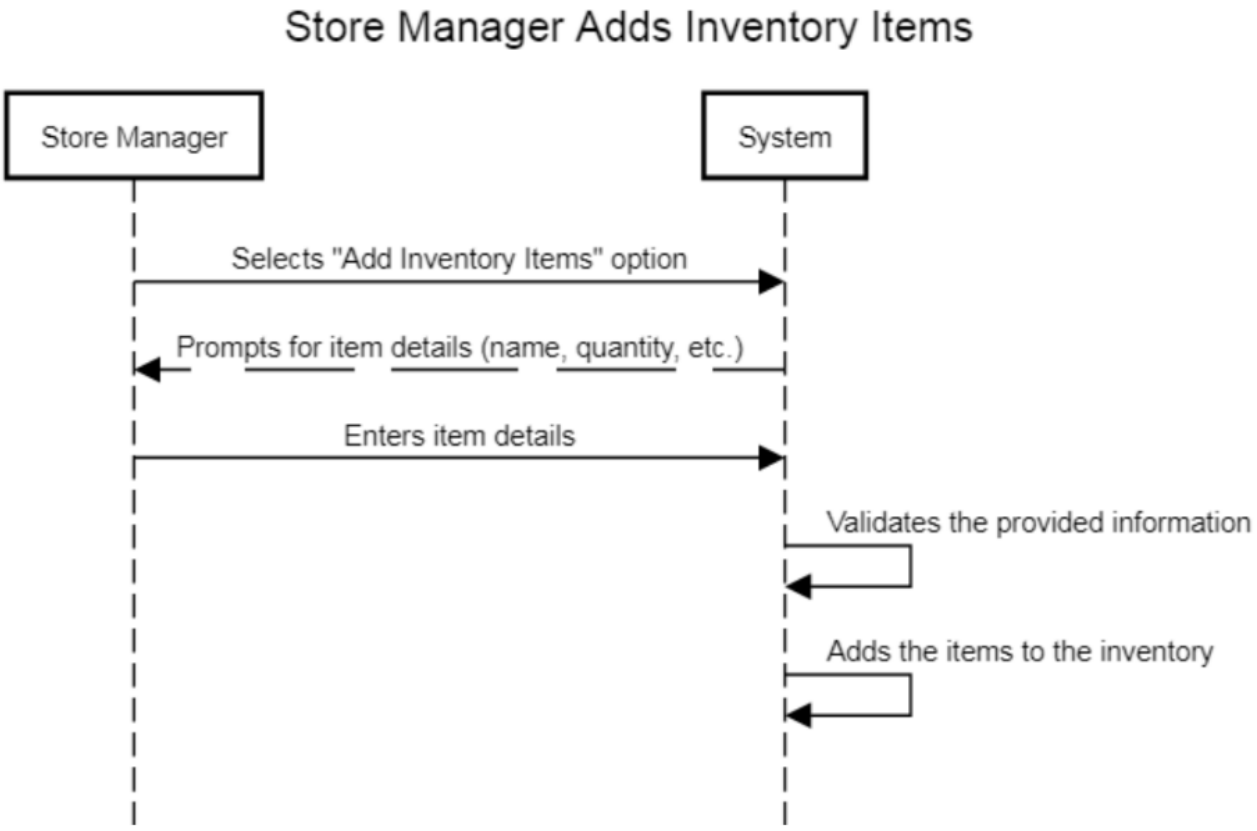
- *Components:*
  -> Secure payment gateway options.
  -> Finalized cart display before payment.
- *Standards:*
  -> Highlight selected payment method.
  -> Display transaction confirmation and receipt.

4. Domain Model



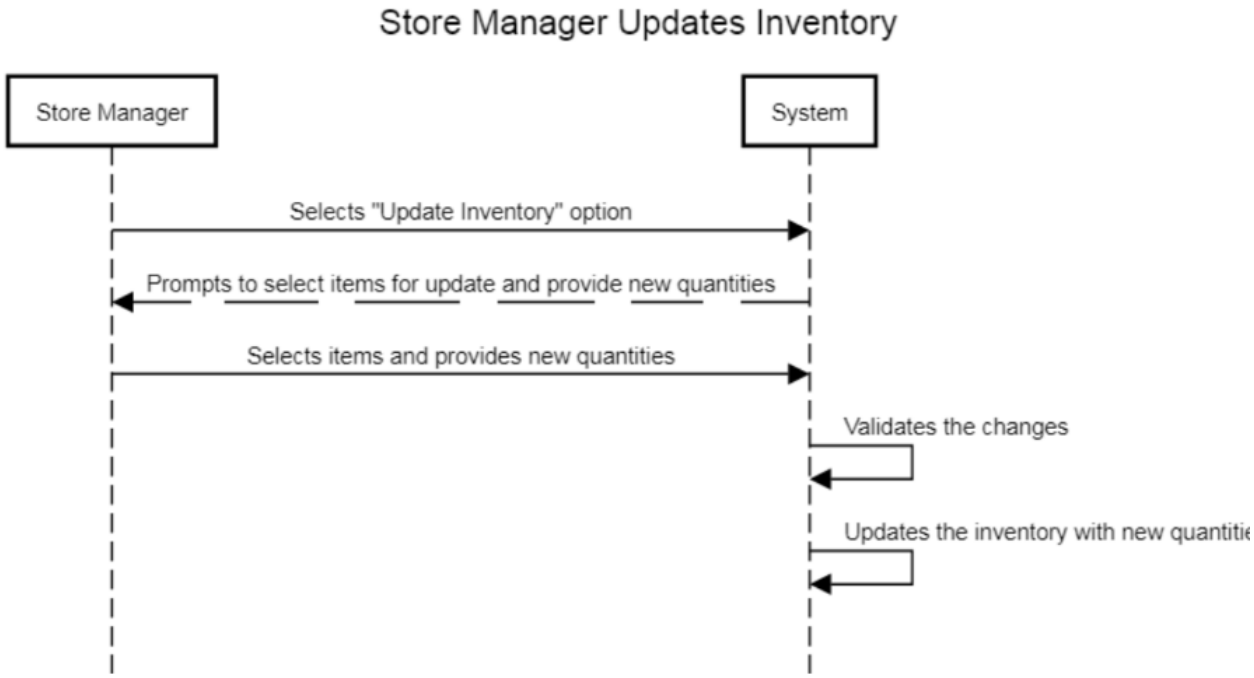
5. System Sequence Diagram

System sequence diagram: 3



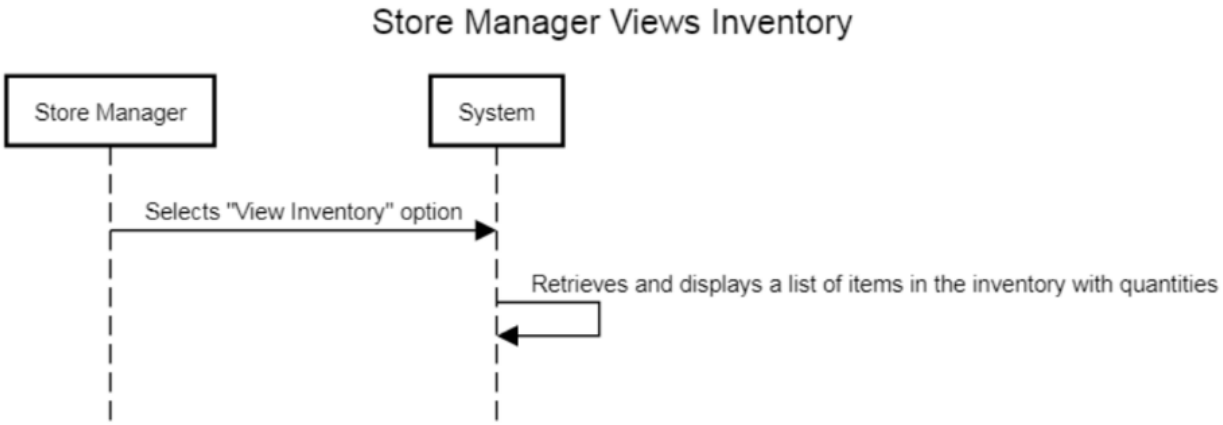
By Munam Mustafa

System sequence diagram: 4



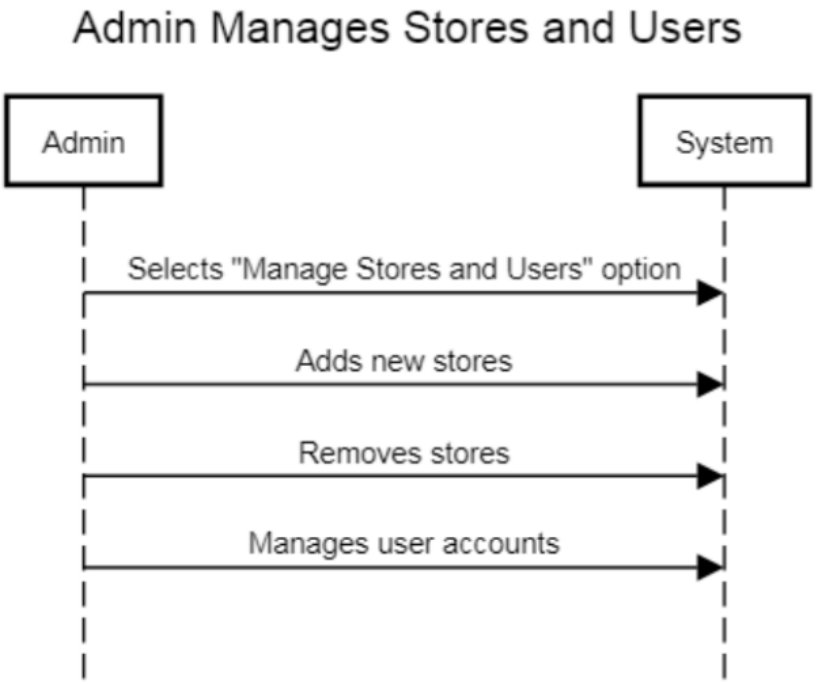
By Munam Mustafa

System sequence diagram: 5



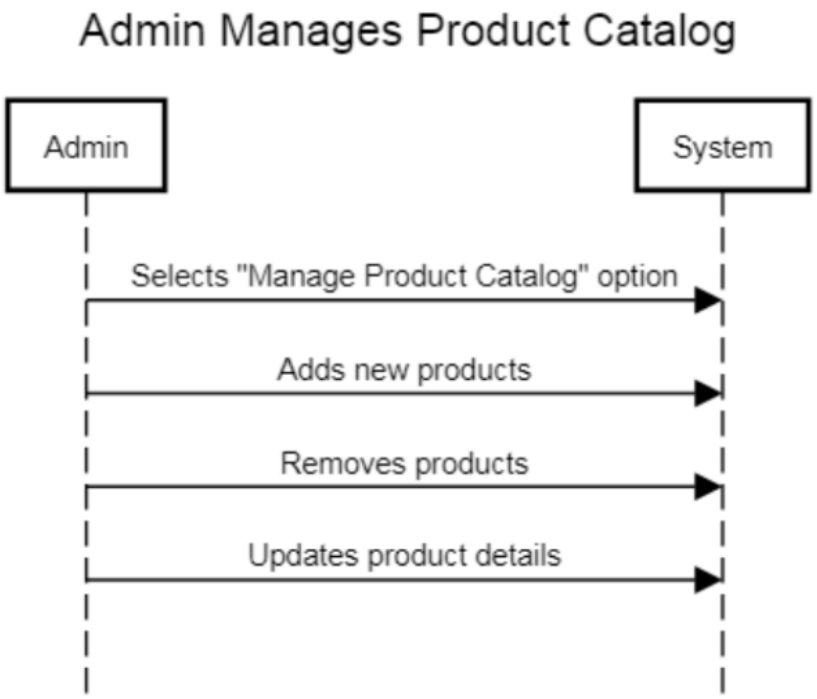
By Munam Mustafa

System sequence diagram: 6



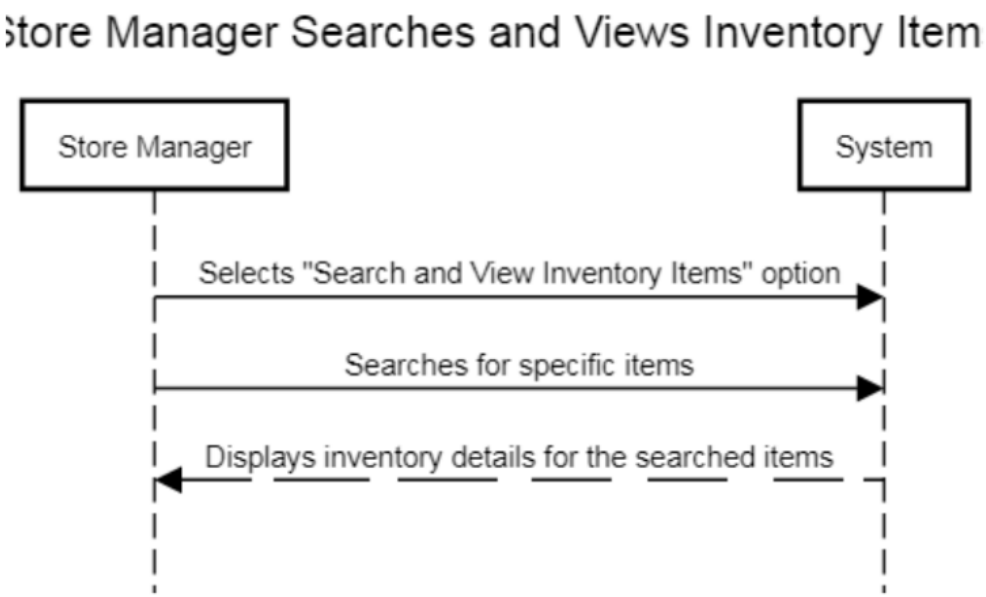
By Munam Mustafa

System sequence diagram: 7



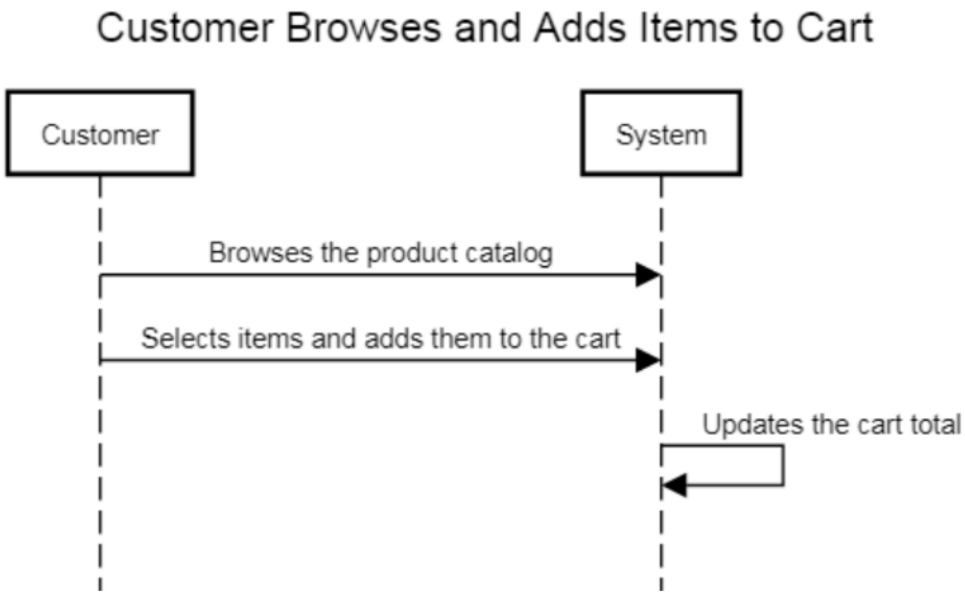
By Munam Mustafa

System sequence diagram: 8



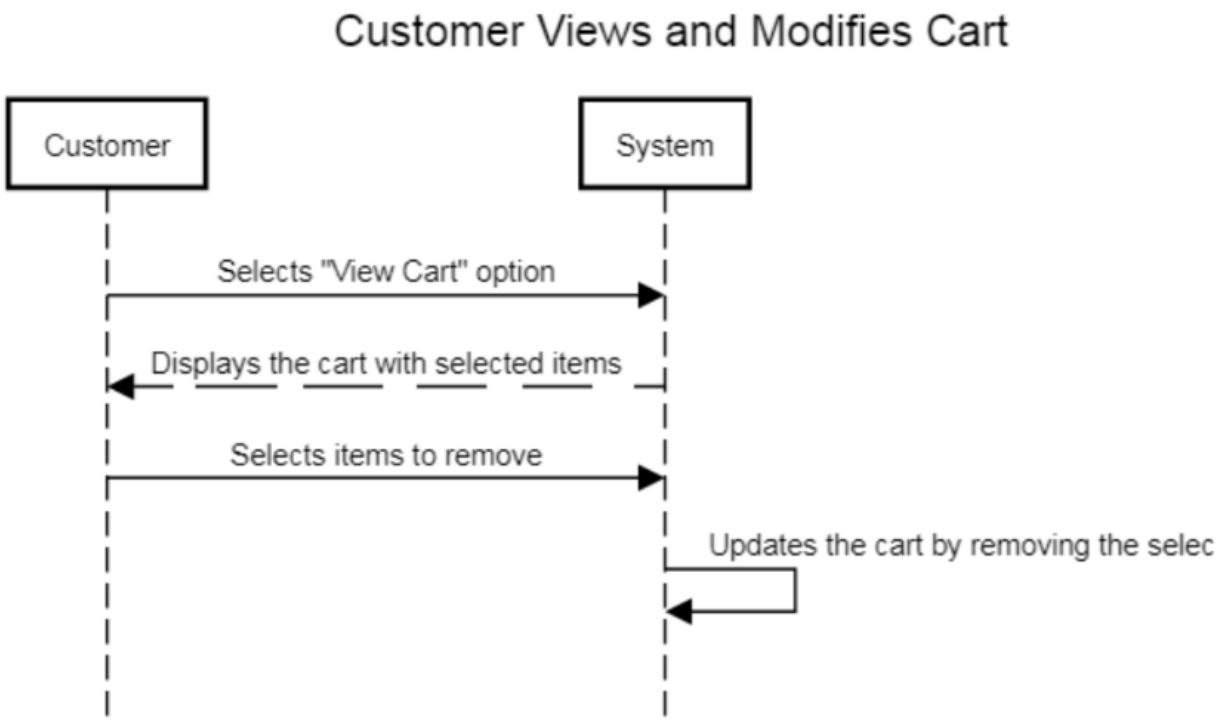
By Munam Mustafa

System sequence diagram: 9

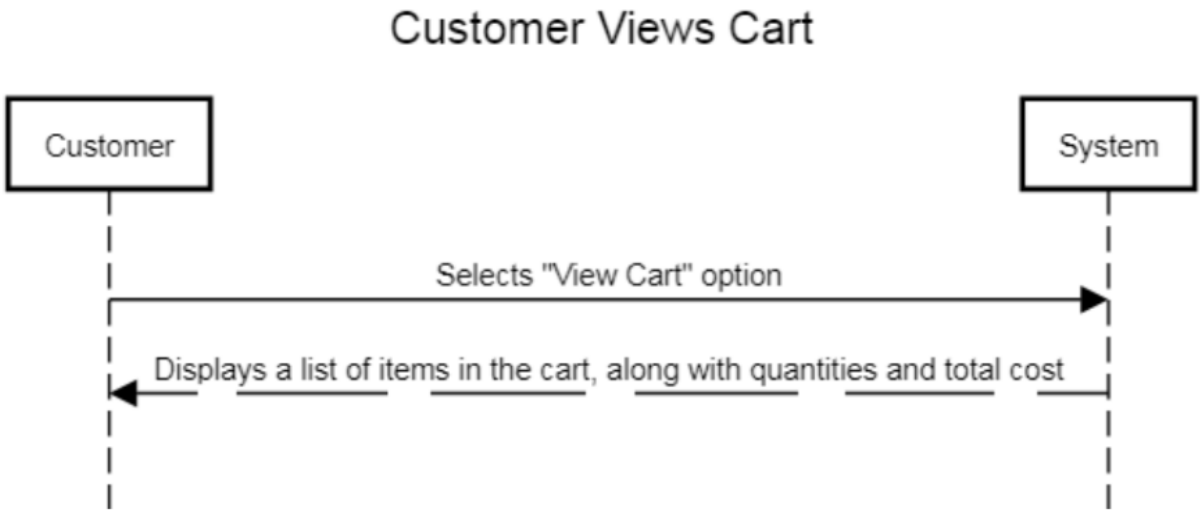


By Munam Mustafa

System sequence diagram: 10

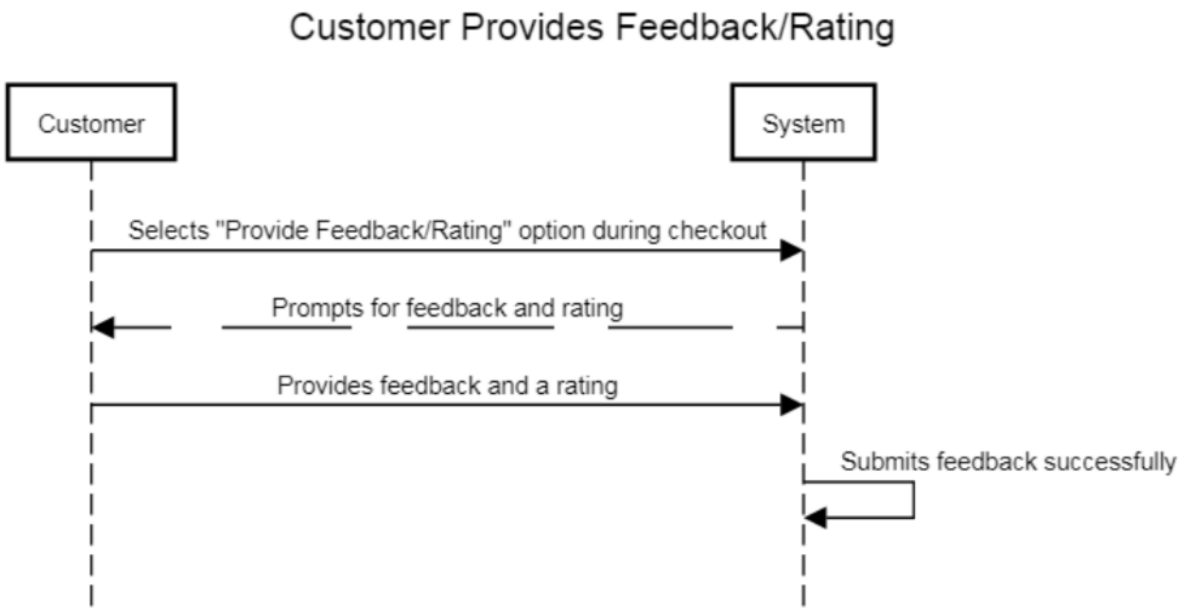


System sequence diagram: 11



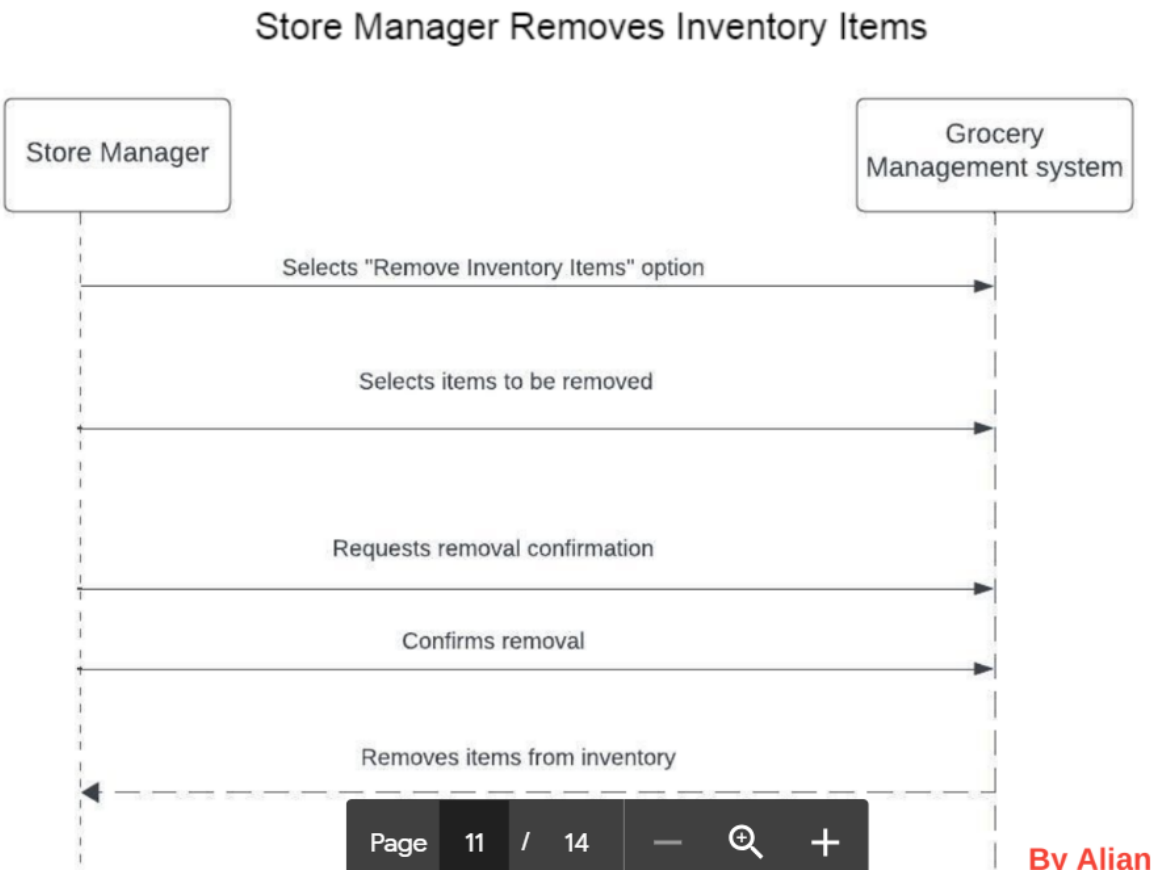
By Alian Anwar

System sequence diagram: 13

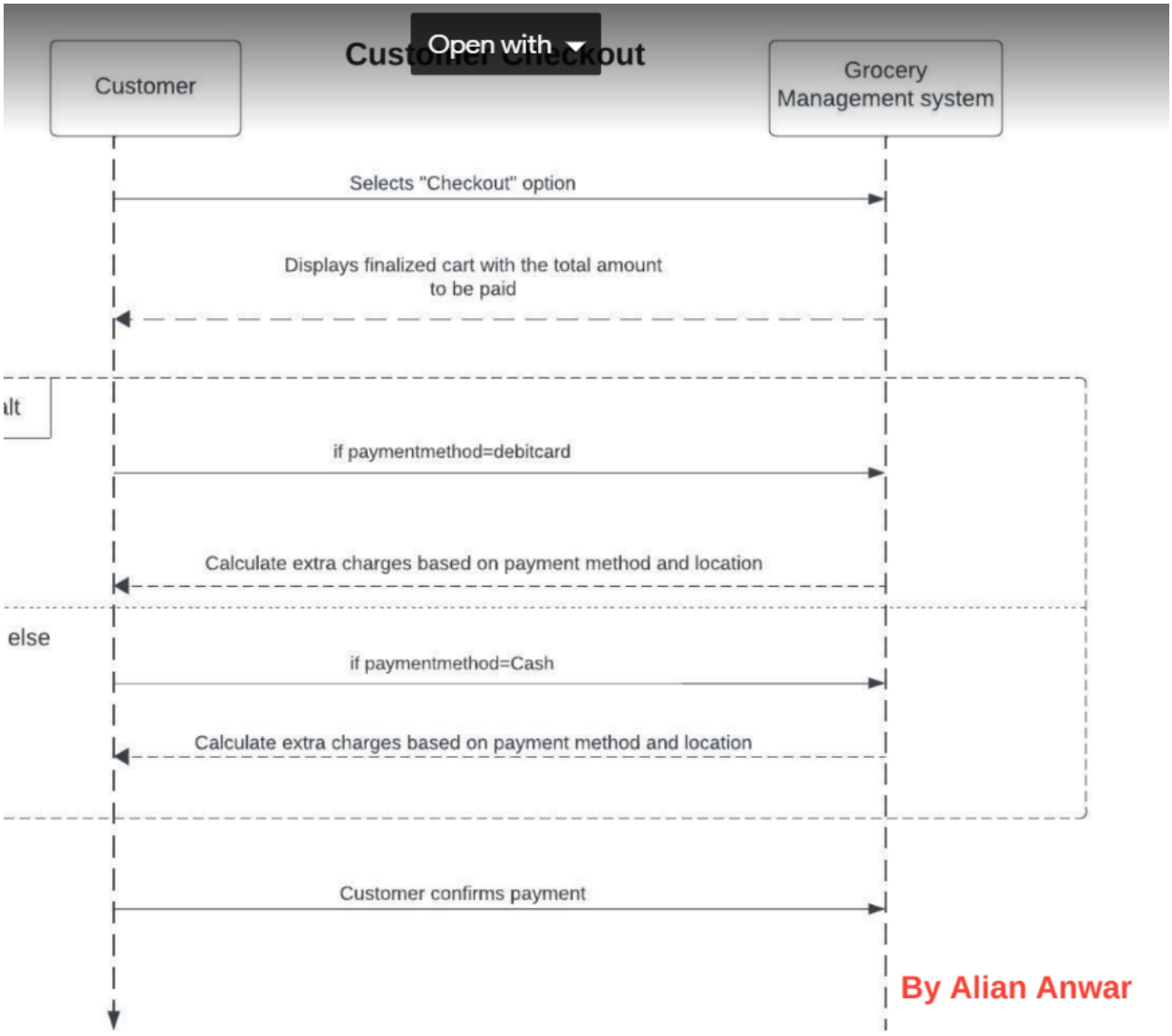


By Alian Anwar

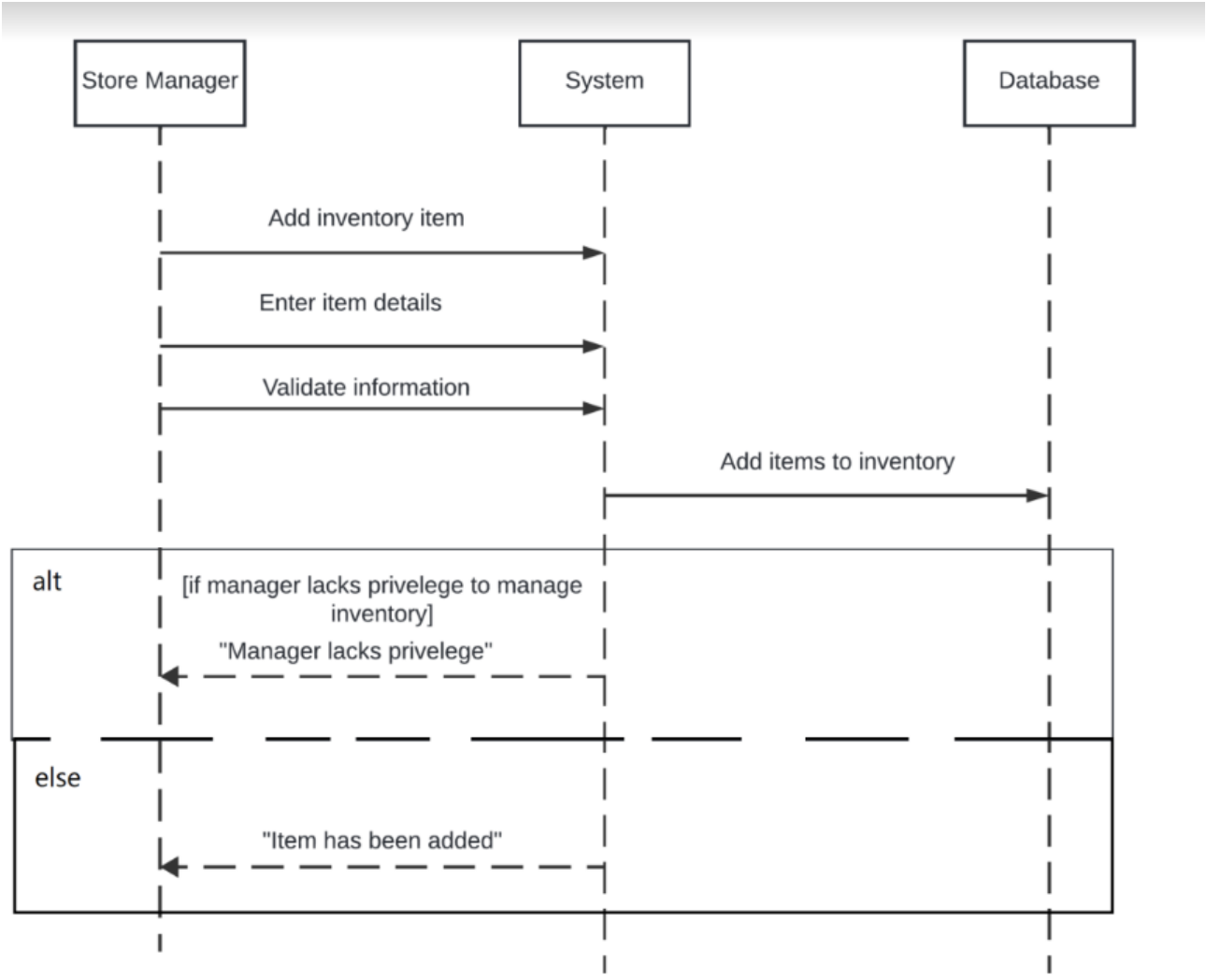
System sequence diagram: 14

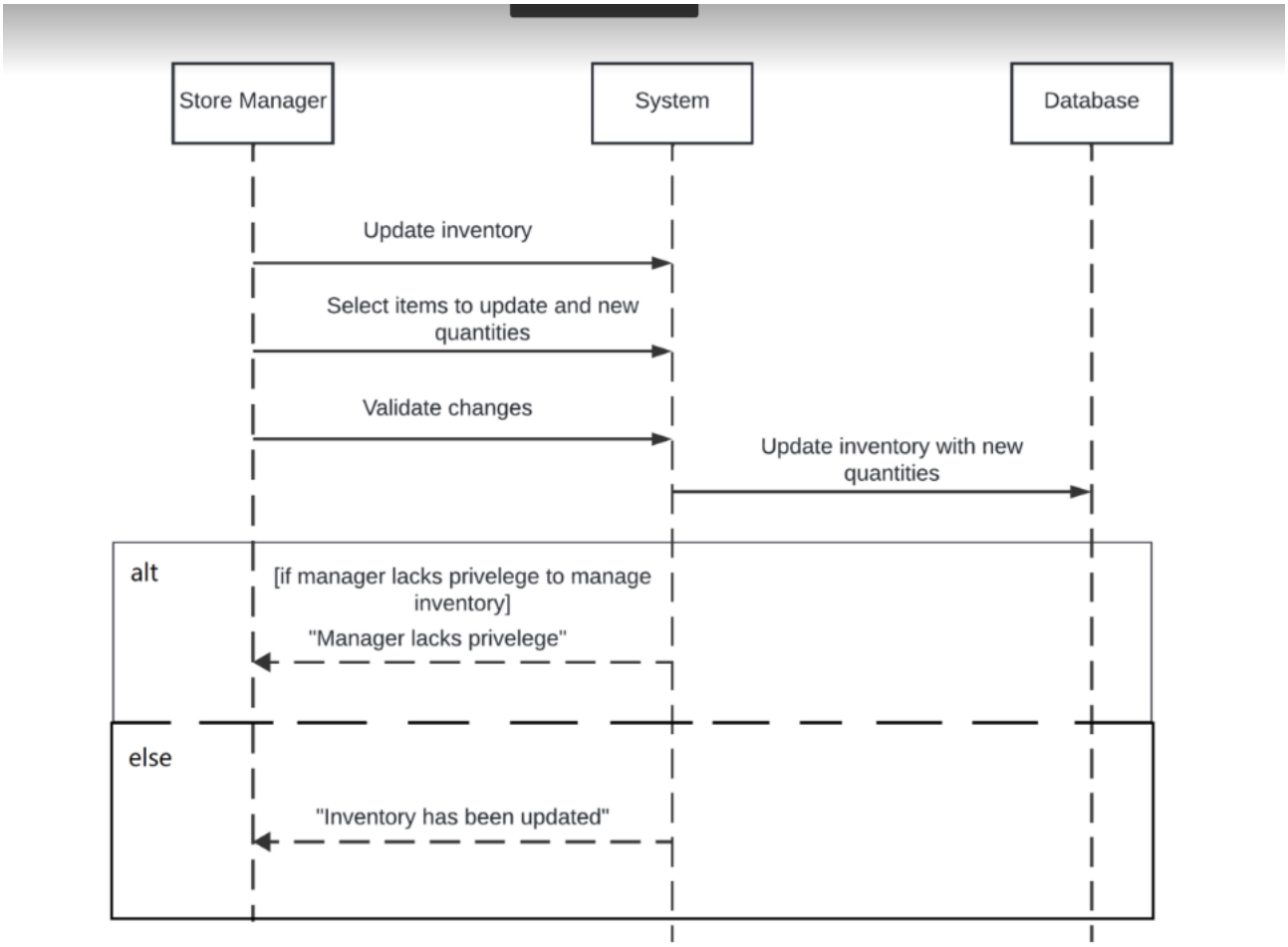


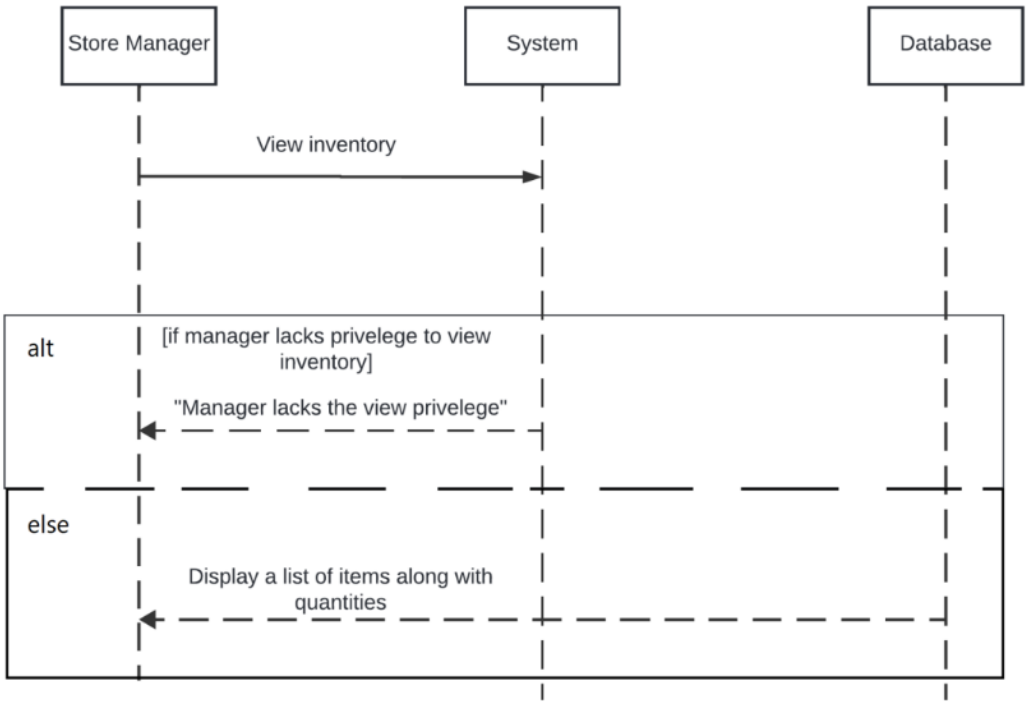
By Alian Anwa

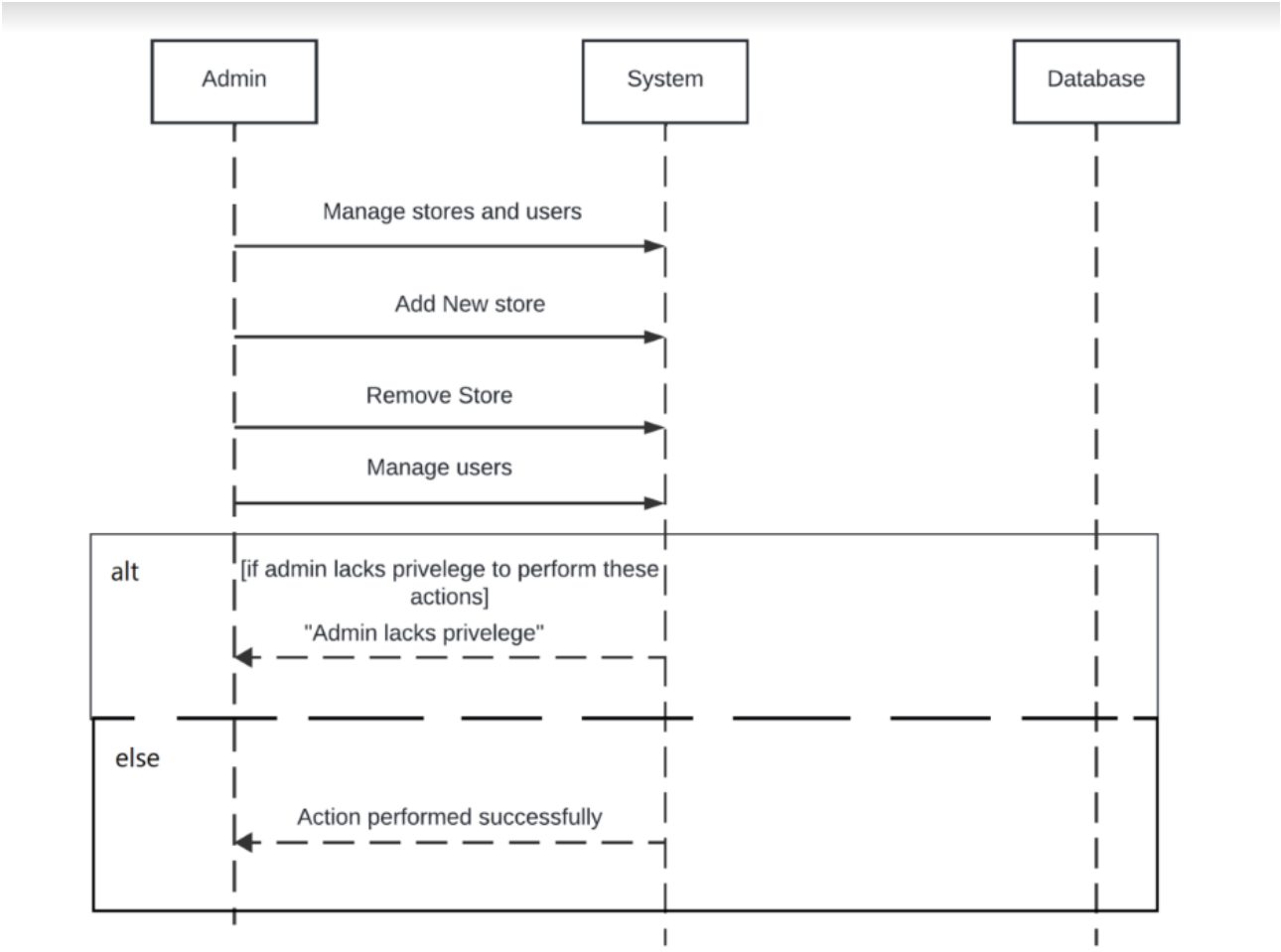

Open with
Customer Checkout
Customer
Grocery Management system
Selects "Checkout" option
Displays finalized cart with the total amount to be paid
alt
if paymentmethod=debitcard
Calculate extra charges based on payment method and location
else
if paymentmethod=Cash
Calculate extra charges based on payment method and location
Customer confirms payment
By Alian Anwar

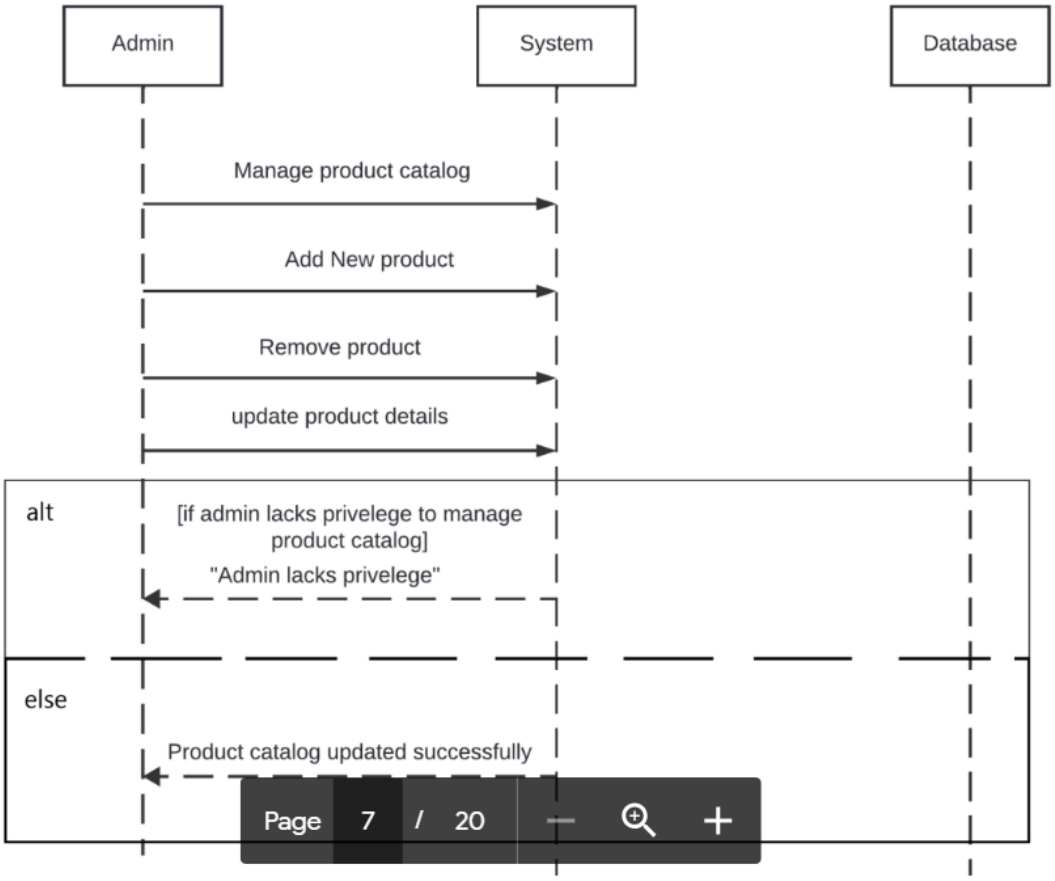
6. Sequence Diagram

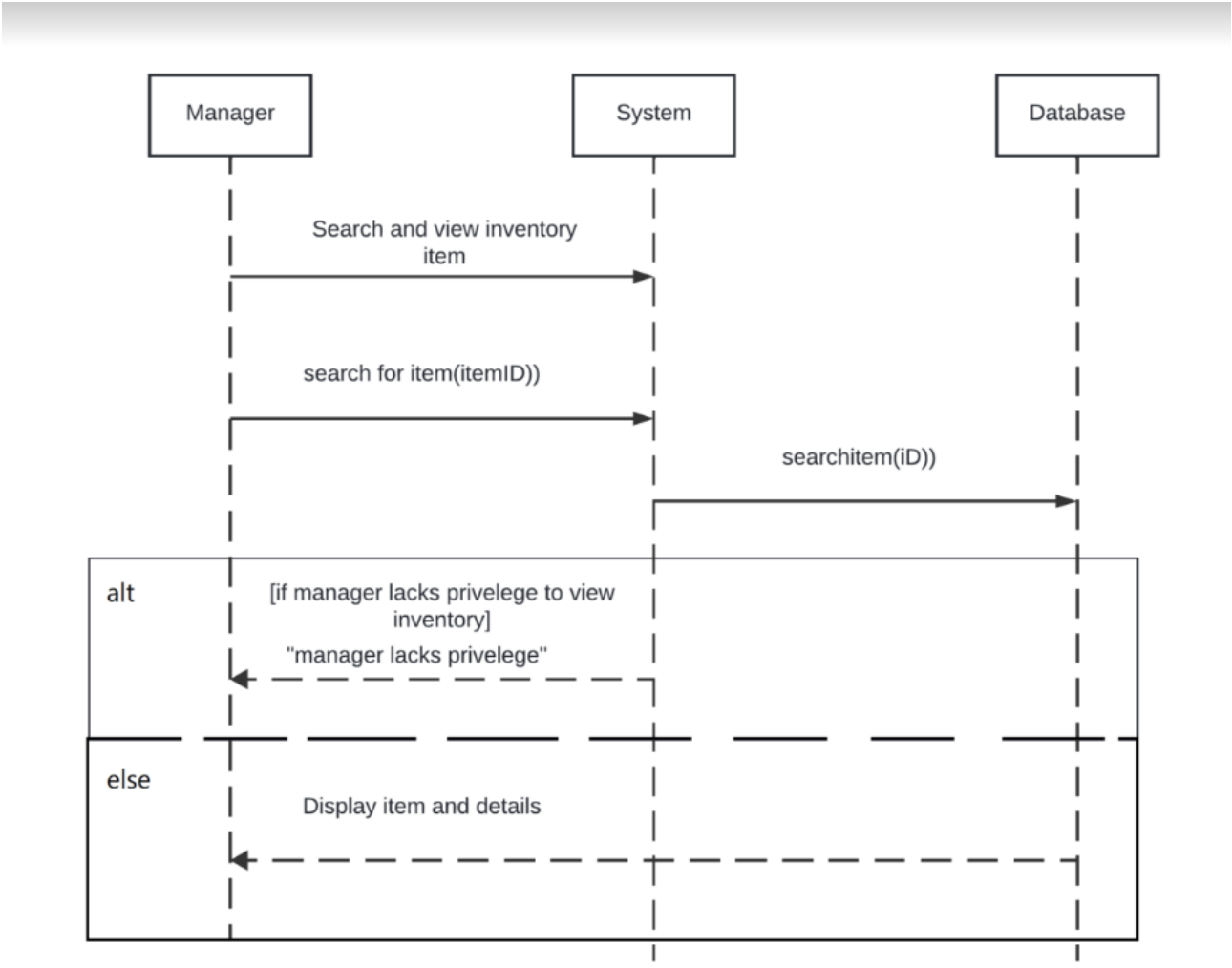

Store Manager
System
Database
Add inventory item
Enter item details
Validate information
Add items to inventory
alt
[if manager lacks privelege to manage inventory]
"Manager lacks privelege"
else
"Item has been added"

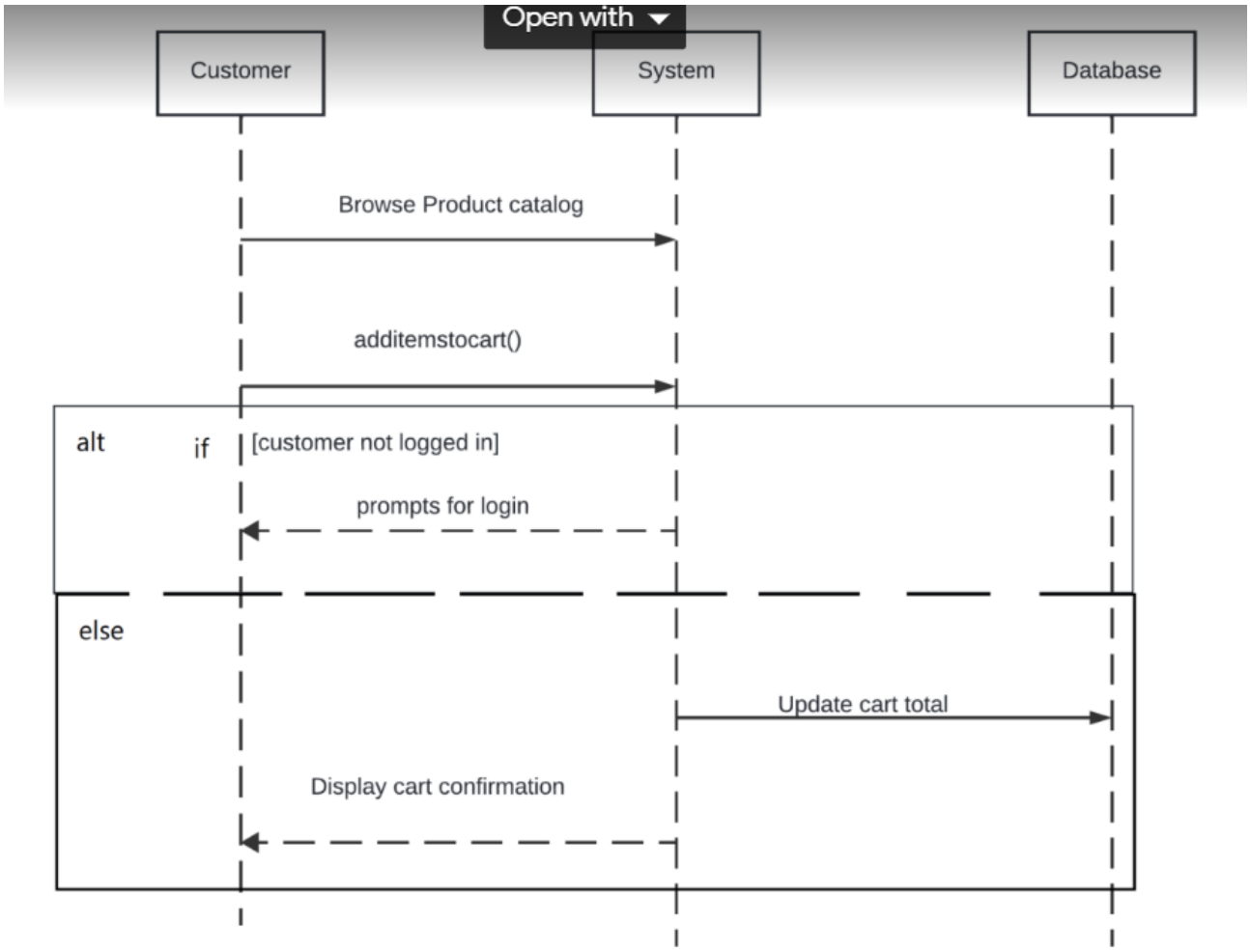

Store Manager
System
Database
Update inventory
Select items to update and new quantities
Validate changes
Update inventory with new quantities
alt
[if manager lacks privelege to manage inventory]
"Manager lacks privelege"
else
"Inventory has been updated"

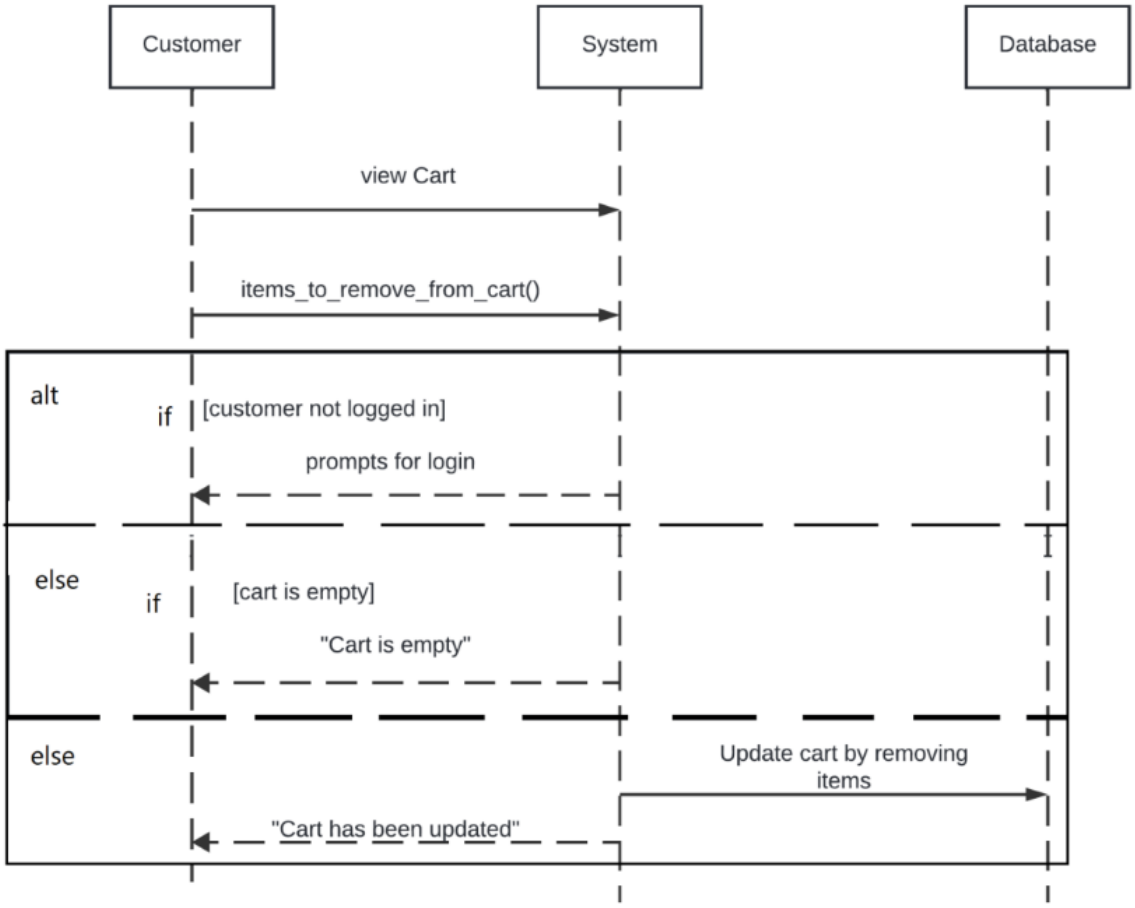

Store Manager
System
Database
View inventory
alt
[if manager lacks privelege to view inventory]
"Manager lacks the view privelege"
else
Display a list of items along with quantities

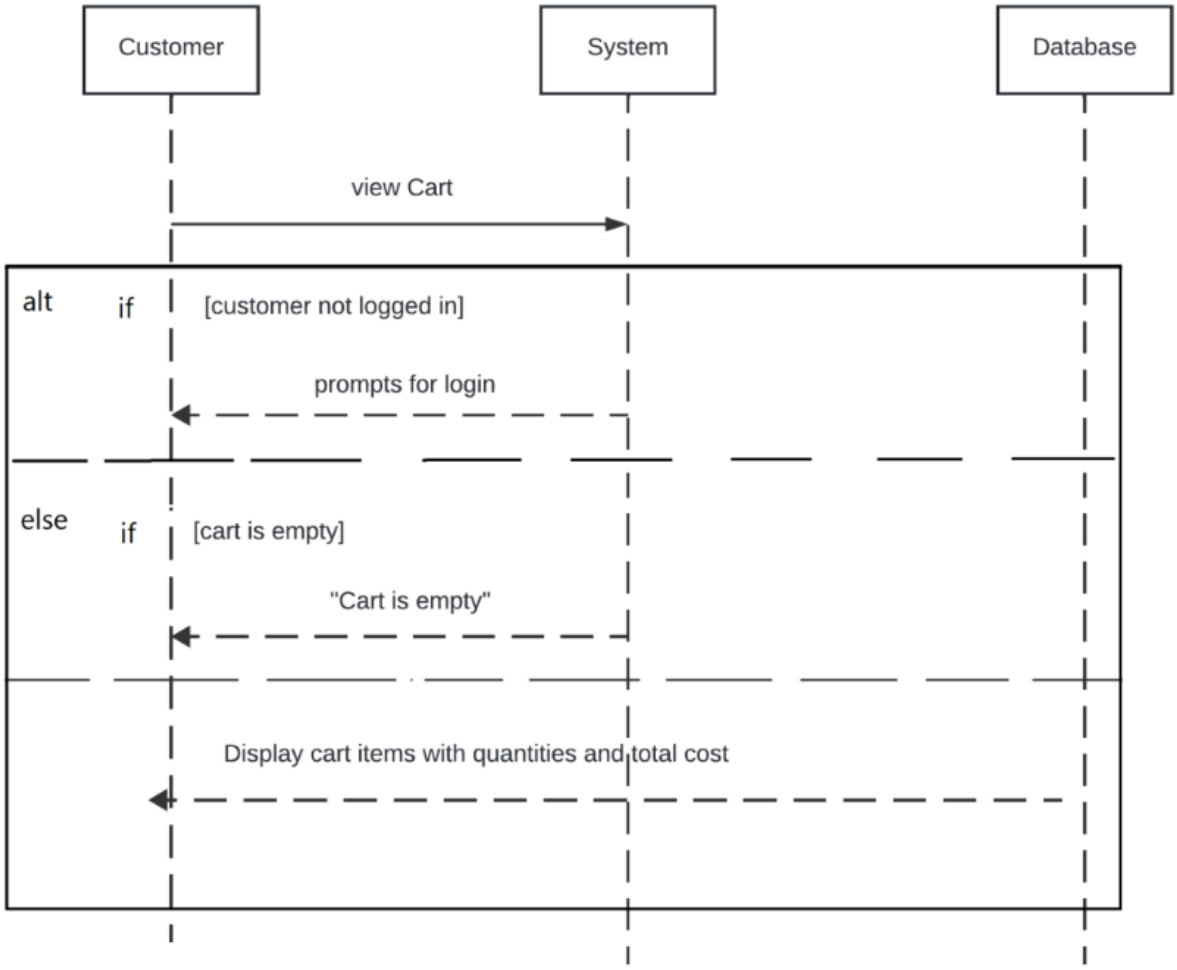

Admin
System
Database
Manage stores and users
Add New store
Remove Store
Manage users
alt
[if admin lacks privelege to perform these actions]
"Admin lacks privelege"
else
Action performed successfully

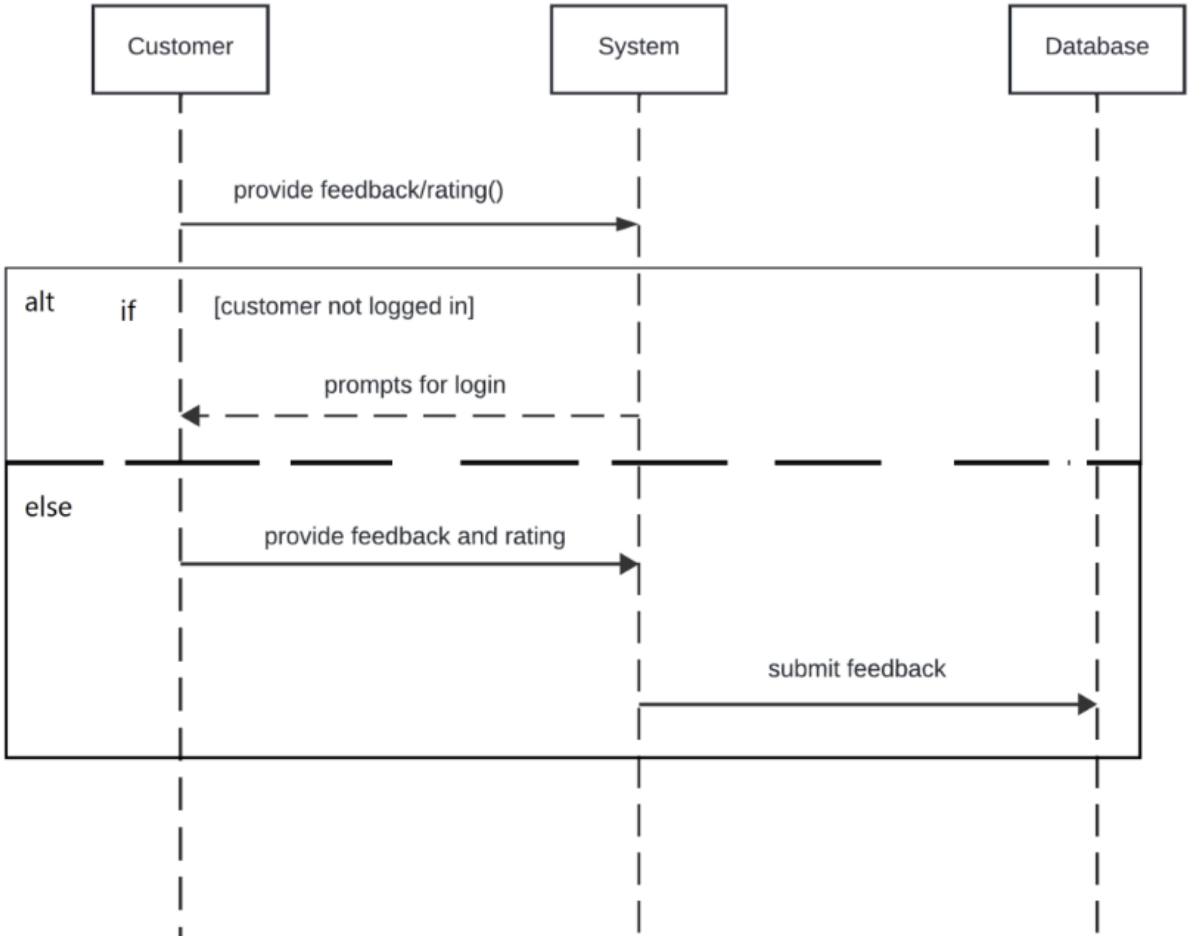

Admin
System
Database
Manage product catalog
Add New product
Remove product
update product details
alt
[if admin lacks privelege to manage product catalog]
"Admin lacks privelege"
else
Product catalog updated successfully
Page 7 / 20

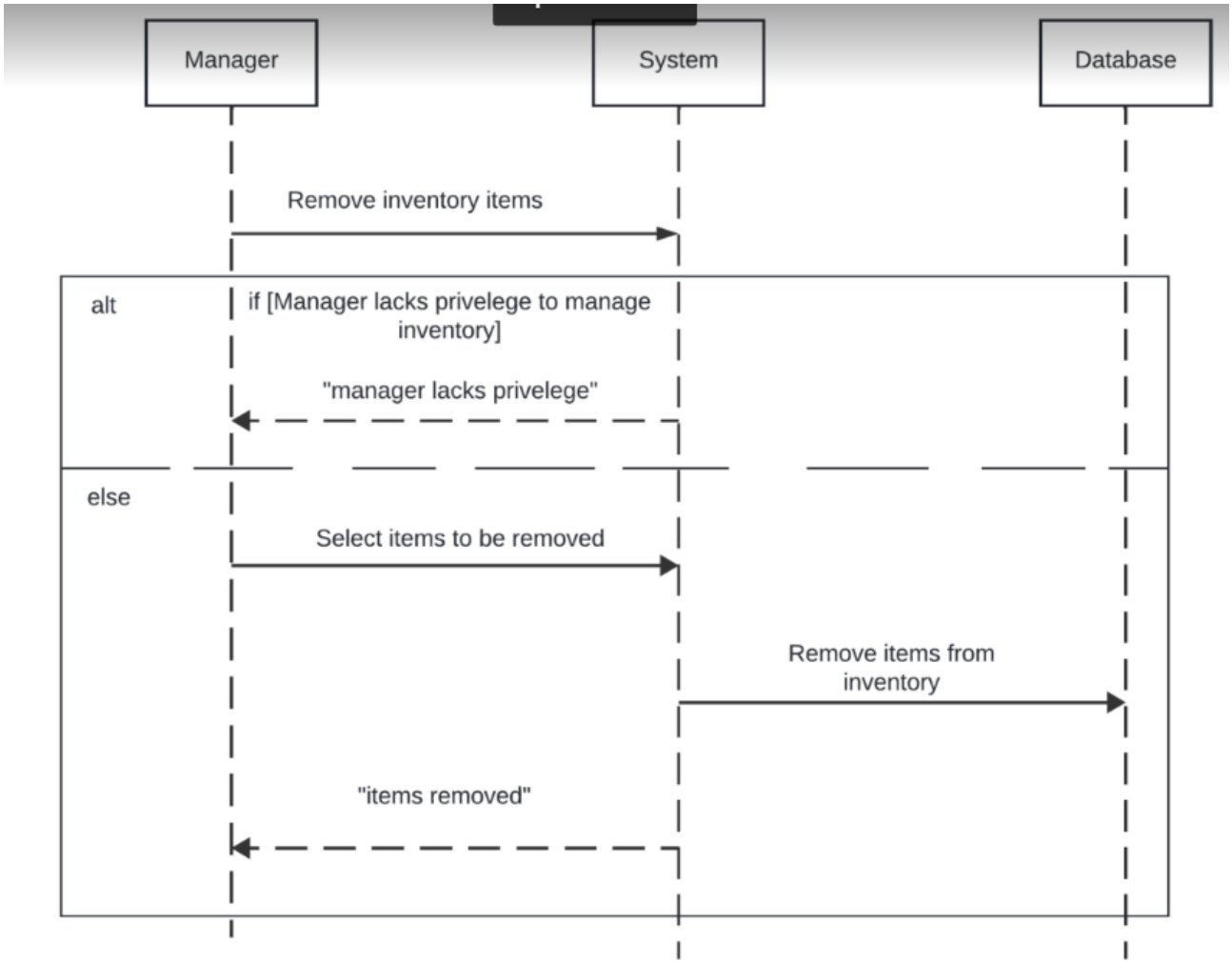

Manager
System
Database
Search and view inventory item
search for item(itemID))
searchitem(iD))
alt
[if manager lacks privelege to view inventory]
"manager lacks privelege"
else
Display item and details

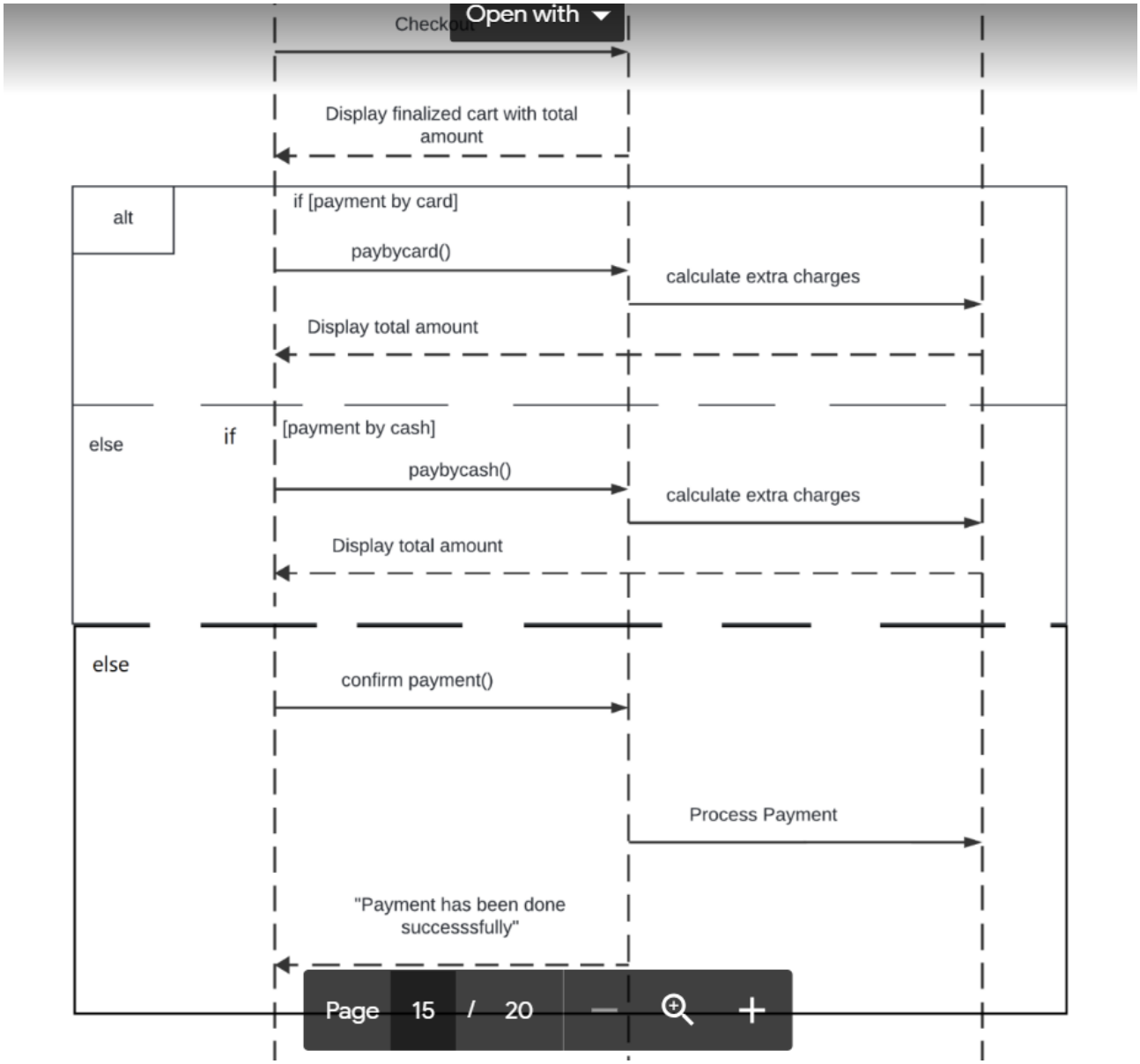

Open with
Customer
System
Database
Browse Product catalog
additemstocart()
alt
if
[customer not logged in]
prompts for login
else
Update cart total
Display cart confirmation


Customer
System
Database
view Cart
items_to_remove_from_cart()
alt
if
[customer not logged in]
prompts for login
else
if
[cart is empty]
"Cart is empty"
else
Update cart by removing items
"Cart has been updated"


Customer
System
Database
view Cart
alt
if
[customer not logged in]
prompts for login
else
if
[cart is empty]
"Cart is empty"
Display cart items with quantities and total cost

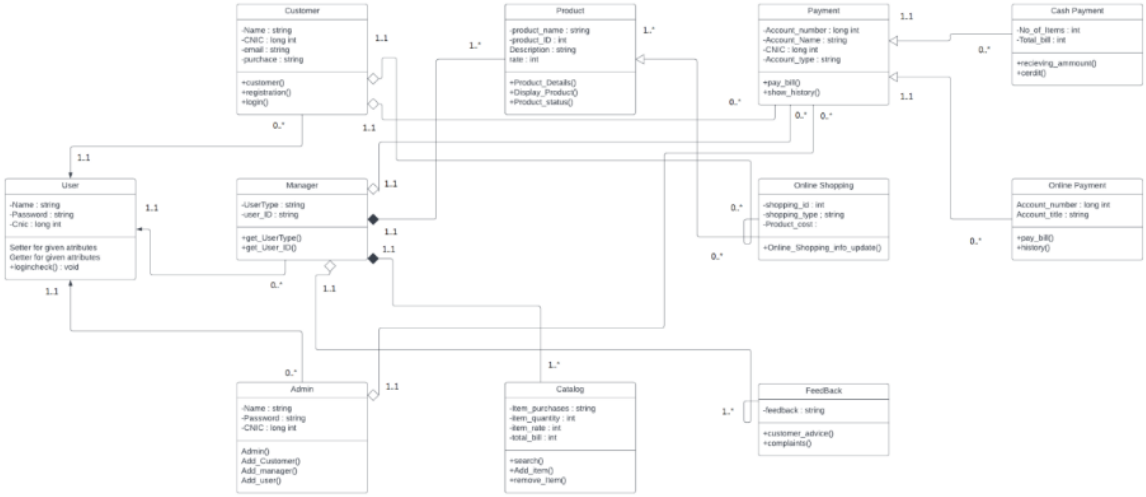

Customer
System
Database
provide feedback/rating()
alt
if
[customer not logged in]
prompts for login
else
provide feedback and rating
submit feedback


Manager
System
Database
Remove inventory items
alt
if [Manager lacks privelege to manage inventory]
"manager lacks privelege"
else
Select items to be removed
Remove items from inventory
"items removed"


Open with
Checkout
Display finalized cart with total amount
alt
if [payment by card]
paybycard()
calculate extra charges
Display total amount
else if
[payment by cash]
paybycash()
calculate extra charges
Display total amount
else
confirm payment()
Process Payment
"Payment has been done successsfully"
Page 15 / 20

7. Class Diagram

Class Diagram:



By Munam Mustafa & Alian Anwar