**Smart Inventory and Order Management System (SIOMS)**

|  |
| --- |
| **Riphah** |

**By:**

**Madiha Abbasi**

**(47383)**

**Hanzla Alvi**

**(47583)**

**Ayesha Butt**

**(48288)**

**Rida Fatima**

**(48403)**

**Faculty of Computing**

**Riphah International University, Islamabad**

**Fall 2022A Dissertation Submitted To**

**Faculty of Computing,**

**Riphah International University, Islamabad**

**As a Partial Fulfillment of the Requirement for the Course Software Construction and Development**

**Bachelors of Science in Software Engineering**

**Dedication/Acknowledgment**

First of all, we would thank Allah Almighty for being able to complete this end-term project and report with success. Then, we would also like to thank to our parents who greatly cooperated and gave us time to complete this report on time.

**Abstract**

The **Smart Inventory and Order Management System (SIOMS)** is a comprehensive software solution designed to optimize inventory management, automate order processing, and enhance operational efficiency within various industries, including online pharmacies. With the growing complexity of modern software systems, robust documentation is crucial to ensure all stakeholders share a mutual understanding of the system's features and functionality.

This report outlines the core components of **SIOMS**, including automated stock tracking, bill calculation, receipt generation, and sales reporting. The system improves efficiency by automating manual processes, reducing human errors, and offering real-time updates on inventory levels and order status. The **SIOMS** system also allows customers to place online orders, check stock availability, and receive automated billing and receipts, thus improving customer experience and store productivity.

The report includes detailed use-case models and a **Requirement Traceability Matrix** to map system requirements with corresponding use cases and test cases, ensuring that all system functions align with user needs. Additionally, the **Supplementary Specification** document highlights the system’s quality attributes and design constraints, ensuring that the system meets required standards and operates effectively within its intended environment.

**SIOMS** serves as an essential tool for improving inventory and order management efficiency while ensuring accuracy, reducing operational risks, and enhancing customer satisfaction.

Contents

[**Description:** 8](#_Toc183553888)

[**1.** **Use-case Diagram** 9](#_Toc183553889)

[1. ID: UC-01 10](#_Toc183553890)

[2. ID: UC-08 10](#_Toc183553891)

[3. ID: UC-08 11](#_Toc183553892)

[**3.** **Sequence Diagram** 12](#_Toc183553893)

[**4.** **Activity Diagram** 13](#_Toc183553894)

[**5.** **Domain Diagram** 14](#_Toc183553895)

[**6.** **State Diagram** 15](#_Toc183553896)

[**7.** **Grasp MVC** 16](#_Toc183553897)

***Project Title: Smart Inventory and Order Management System***

# **Description:**

**SIOMS (Smart Inventory and Order Management System)** is an advanced software solution developed to automate and optimize the core processes of an online pharmacy management system. In a traditional pharmacy, managing inventory, processing orders, calculating bills, and generating receipts manually can be both time-consuming and prone to errors. Customers often face delays in checking product availability, which wastes their time and leads to frustration. Additionally, manual processes put a heavy burden on staff, reducing operational efficiency and increasing the chances of mistakes. **SIOMS** addresses these challenges by automating essential tasks such as stock management, order processing, billing, and report generation. The system continuously updates inventory in real-time, ensuring accurate stock information and minimizing the risk of stockouts or overstocking. Customers can place orders both online and manually, and the system automatically updates the stock levels as orders are placed. The bill calculation and receipt generation process is fully automated, making transactions faster and more efficient. **SIOMS** also generates real-time sales and income reports, providing the pharmacy with valuable insights into performance and revenue. In addition, it keeps track of supplier and revenue records, ensuring that the pharmacy maintains accurate financial and supply chain data. By automating these processes, **SIOMS** significantly enhances operational efficiency, reduces errors, and provides a more streamlined, reliable service for both customers and staff.

# **Use-case Diagram**

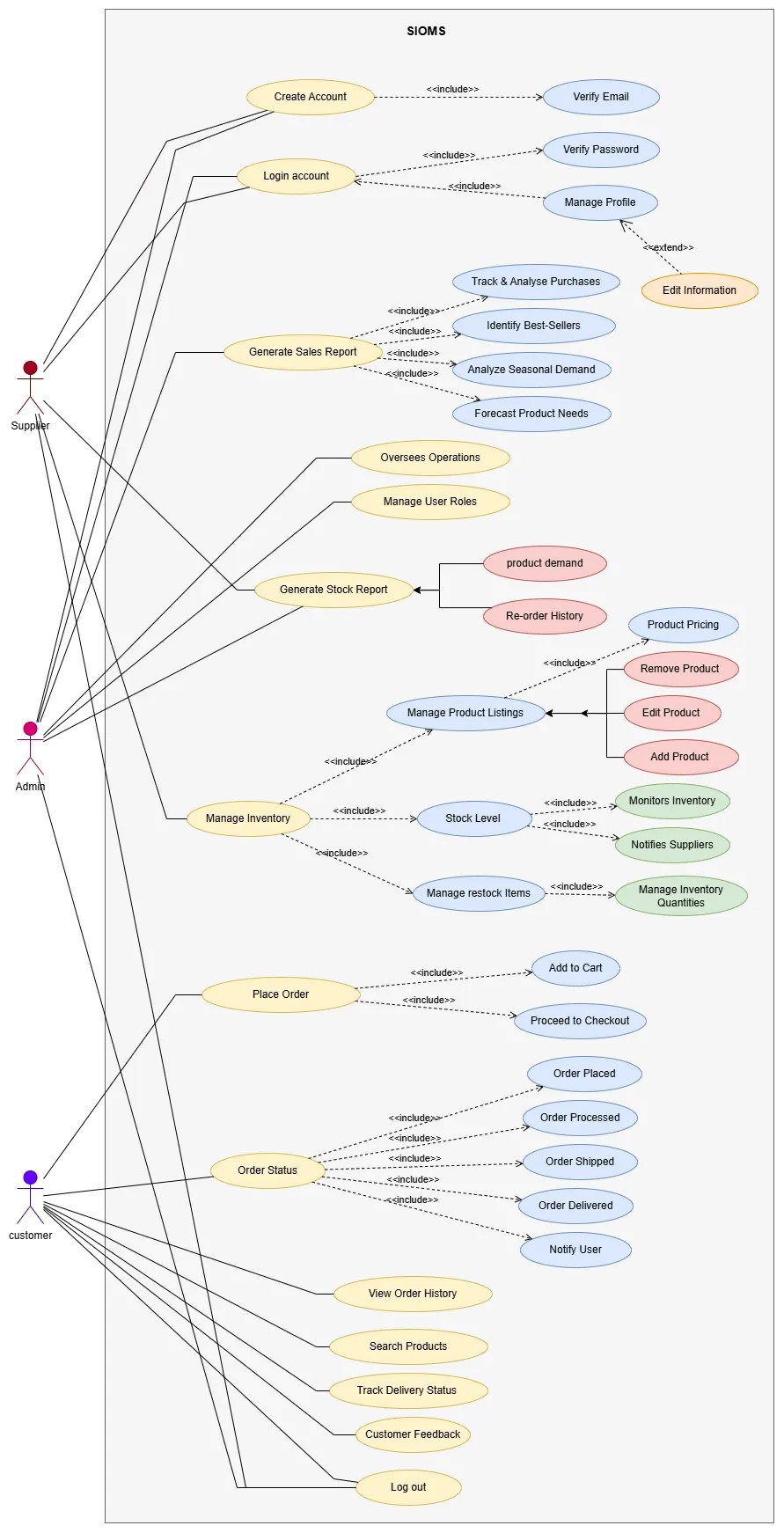


Figure Use Case Diagram

1. **Fully Dress Use Cases**

# ID: UC-01

|  |  |
| --- | --- |
| **Section** | **Content** |
| **Designation** | Smart Inventory Order Management System |
| **Name** | Order Status |
| **Authors** | Rida Fatima |
| **Priority** | High |
| **Criticality** | Critical |
| **Source** | Customer Requirements |
| **Responsible** | Development Team |
| **Description** | This use case allows customers to check the status of their orders. The system provides detailed info including order processing, shipping, delivery status, and notifications. |
| **Trigger event** | The customer logs into the system and selects the "Order Status" option. |
| **Actors** | Customer |
| **Preconditions** | 1. The customer must have access to place an order.  2. The customer must be logged into the system. |
| **PostCondition** | 1. The customer is informed about the current status of their order.  2. Notifications, if applicable, are sent to the customer. |
| **Result** | The customer successfully checks the status of their order. |
| **Main Scenario** | 1. The customer logs into their account.  2. The customer selects "Order Status" from the menu.  3. The system retrieves the order details from the database.  4. The system displays the following information:   * Order Placed * Order Processed * Order Shipped * Order Delivered * Notification sent to the user   5. The customer views the detailed order status. |
| **Alternative Scenario** | 1. If the system cannot find the order in the database:   1a.The system displays an error message: "Order not found."   1. If the customer is not logged in:   2a.The system prompts the customer to log in.   1. If there is a system error during status retrieval:   3a.The system displays an error message: "Unable to fetch order status. Please try again later." |

# ID: UC-02

|  |  |
| --- | --- |
| **Section** | **Content** |
| **Designation** | Place Order |
| **Name** | Place Order |
| **Authors** | Madiha Abbasi |
| **Priority** | High |
| **Criticality** | Critical |
| **Source** | SIOMS Project Requirements |
| **Responsible** | Development Team |
| **Description** | Allows the customer to place an order by adding items to the cart and proceeding to checkout. |
| **Trigger event** | Customer adds products to the cart and clicks on "Proceed to Checkout." |
| **Actors** | **Primary:** Customer, Admin, Supplier |
| **Precondition** | 1. Customer must be logged in. 2. Products must be available in the inventory. |
| **Postcondition** | 1. The system records the order in the database. 2. The order receives the status "Placed." |
| **Result** | The customer successfully places the order and receives confirmation, initiating further processing. |
| **Main Scenario** | 1. The customer browses products and adds desired items to the cart. 2. The system verifies product availability and updates the cart. 3. The customer proceeds to checkout and reviews their cart items and total cost. 4. The system records the order as "Placed" and generates a unique order ID. 5. The delivery team delivers the order to the customer and updates its status to "Delivered." 6. Notifications are sent to the customer confirming the order placement. 7. The customer places an order through the system. |
| **Alternative Scenario** | **2a. Product Cannot Be Added to Cart:** The system encounters an issue while adding a product to the cart. The system notifies the customer of the issue and provides recommendations (e.g., check stock availability or try again).  **3a. Product Unavailable:** A product becomes unavailable during checkout. System notifies the customer to remove or replace the unavailable product. |

# ID: UC-03

|  |  |
| --- | --- |
| **Section** | **Details** |
| **Designation** | Manage Inventory |
| **Name** | Inventory Management |
| **Authors** | Hanzla Alvi |
| **Priority** | High |
| **Criticality** | Critical |
| **Source** | Supplier documentation for inventory management |
| **Responsible** | Supplier |
| **Description** | This use case allows suppliers to manage inventory by restocking products and updating stock levels. |
| **Trigger Event** | The system sends a low-stock alert, or the supplier checks stock levels. |
| **Actors** | **Primary**: Supplier |
| **Preconditions** | - Supplier must be logged into the system. - The system must have access to stock data. |
| **PostCondition** | - Stock levels are updated in the system. - Notifications are sent to the Inventory Manager. |
| **Result** | Inventory is updated accurately, and restocking records are saved for future reference. |
| **Main Scenario** | 1. Supplier logs into the inventory system.  2. The system shows the current stock levels for all products.  3. The system highlights any products with low stock.  4. Supplier selects a product with low stock and chooses to restock it.  5. Supplier enters the quantity being restocked (e.g., "50 units").  6. The system updates the stock level for the selected product.  7. The system notifies the Inventory Manager about the restocked items.  8. Supplier logs out after completing the process. |
| **Alternative Scenario** | 2a. System Fails to Load Stock Data:    2a1. System shows an error message.    2a2. Supplier retries or contacts support. 5a. Invalid Restock Quantity:    5a1. System asks the supplier to enter a valid number. 6a. System Fails to Update Stock:    6a1. System logs the error and informs the supplier. |

# **Sequence Diagram**

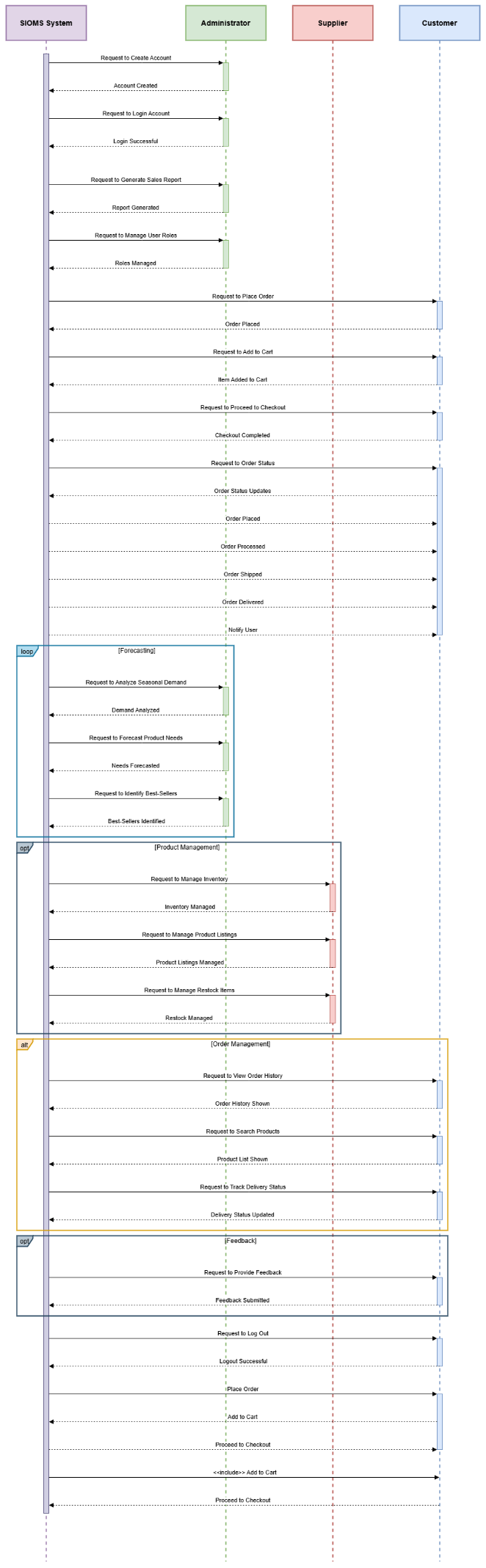


Figure Sequence Diagram

# **Activity Diagram**

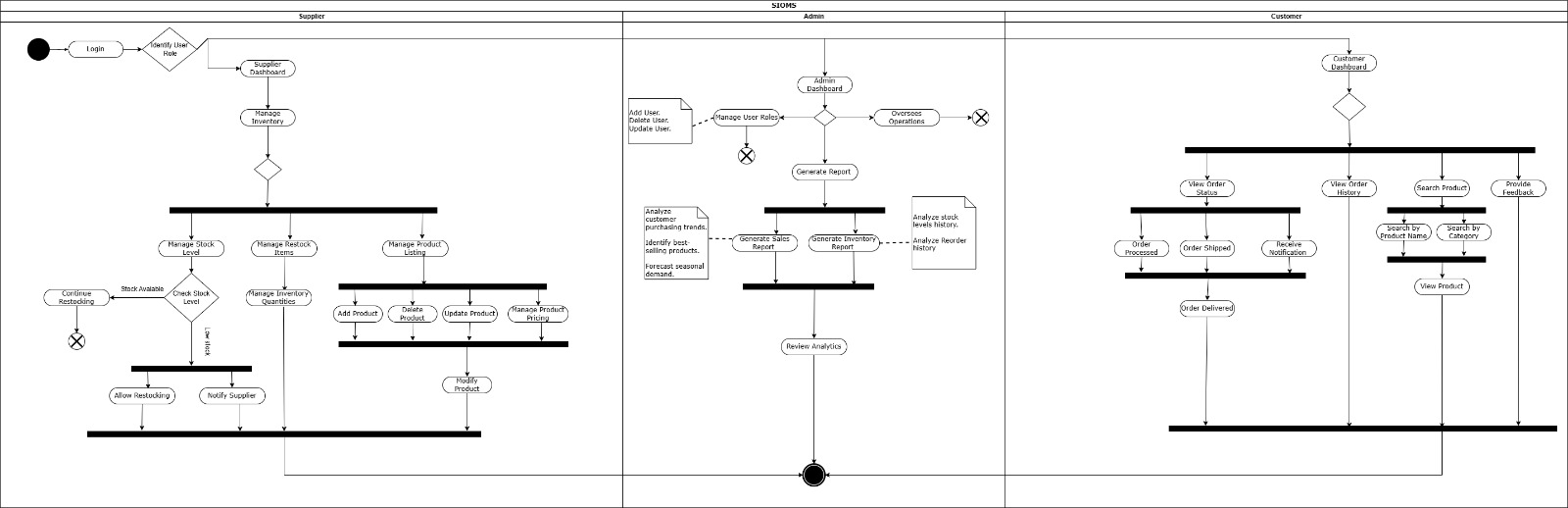


Figure Activity Diagram

# **Domain Diagram**

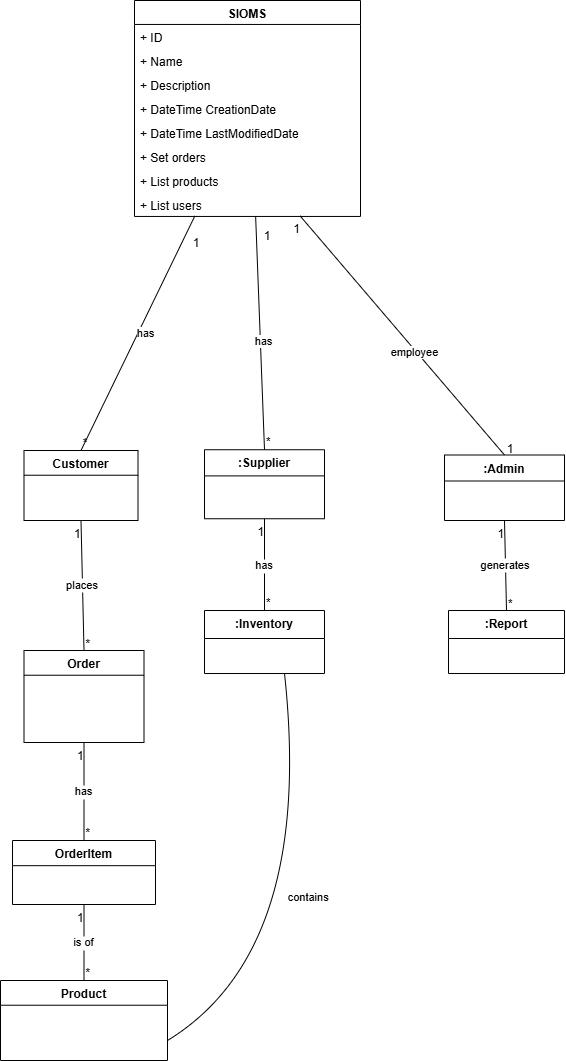


Figure Domain Diagram

# **State Diagram**

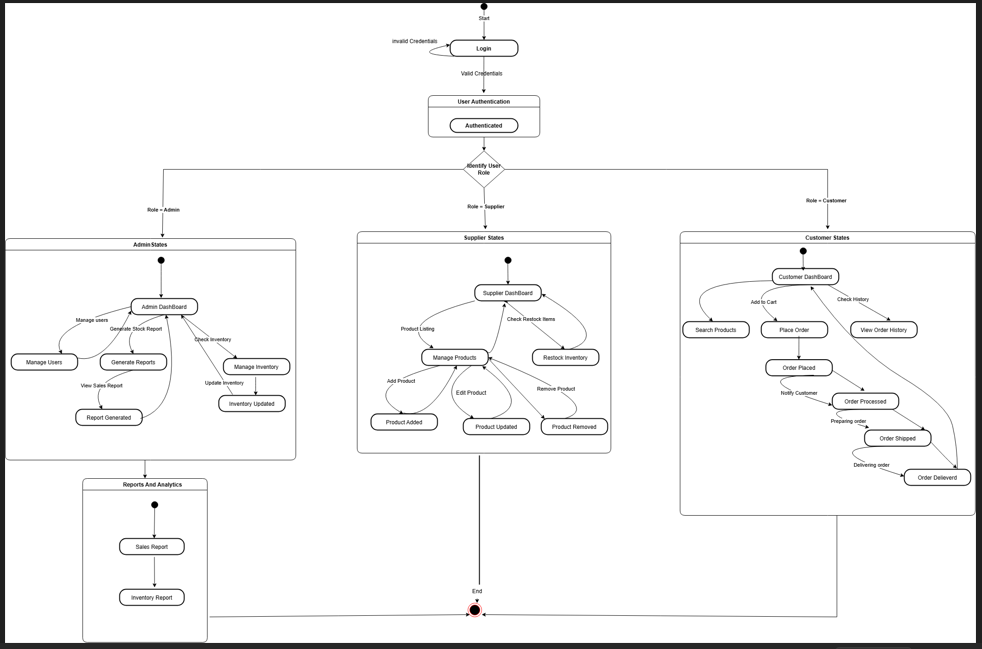


Figure State Diagram

# **Grasp MVC**