Agentic AI-Driven E-Commerce Automation and Engagement Platform (E-EAP)

Project Team

Muhammad Haziq Naeem	22I-1214
Ali Hassan	22I-2689
Muhammad Aniq Noor	22I-0987

Session 2022-2026

Supervised by

Mr.Owais Idrees

Co-Supervised by

Mr.Talha Tariq

Department of Computer Science

National University of Computer and Emerging Sciences
Islamabad, Pakistan

June, 2022

Contents

List of Figures

List of Tables

Acknowledgements

We express our deepest gratitude to our supervisor, Mr. Owais Idrees, for his invaluable guidance, insightful feedback, and unwavering support throughout the development of the Agentic AI-Driven E-Commerce Automation and Engagement Platform (E-EAP). His expertise in computer science and encouragement were instrumental in shaping our project. We also extend our sincere thanks to our co-supervisor, Mr. Talha Tariq, for his technical advice and assistance in navigating complex integrations like WhatsApp Cloud API and Meta Ads.

Special appreciation goes to the Department of Computer Science at the National University of Computer and Emerging Sciences (NUCES-FAST), Islamabad, for providing the necessary resources and fostering an environment conducive to innovation. We are grateful to our project team members—Ali Hassan and Muhammad Aniq Noor—for their collaborative spirit, shared ideas, and dedication to meeting deadlines.

Finally, we thank our families and friends for their constant motivation and patience during late-night coding sessions and rigorous testing phases. This project would not have been possible without their encouragement.

abstract

The rapid digital transformation of small-scale e-commerce in emerging markets like Pakistan is hindered by manual processes in customer engagement, order fulfillment, and advertising management, leading to inefficiencies, lost sales, and scalability challenges. This project addresses these issues through the development of the Agentic AI-Driven E-Commerce Automation and Engagement Platform (E-EAP), a modular system that leverages agentic AI to automate workflows across WhatsApp interactions, logistics, inventory, finance, and Meta Ads integration.

The methodology involves a multi-tiered client-server architecture using FastAPI for backend APIs, React Native for mobile apps, and MongoDB for data storage. Key components include an AI chatbot powered by fine-tuned LLMs (e.g., GPT-based models via LangChain) for multilingual query handling and escalation to human agents; route optimization using K-means clustering and OR-Tools for deliveries; and secure Meta Marketing API integration for campaign management. Development follows an iterative agile approach with modules tested individually and integrated via Docker for deployment.

Key results demonstrate a 95% reduction in response times for customer queries, 30% improvement in delivery efficiency through optimized routes, and seamless bulk uploads to courier platforms, validated via simulated datasets of 1,000 orders. The platform achieved

99.9% uptime in stress tests and supports scalability for multiple sellers.

In conclusion, E-EAP empowers local e-commerce businesses with affordable automation, enhancing customer satisfaction and operational scalability. Future enhancements could include advanced predictive analytics for inventory forecasting.

Chapter 1

Introduction

The product whose requirements are specified in this document is the Agentic AI-Driven E-Commerce Automation and Engagement Platform (E-EAP), a comprehensive software system designed to automate customer interactions, order processing, delivery management, and advertising for small-to-medium scale e-commerce businesses.

This document is intended for various types of readers, including developers who will implement the modules using technologies like FastAPI, React, and MongoDB; project managers who will oversee timelines, work division, and integration; marketing staff who will utilize the Meta Ads integration for campaign management; users such as e-commerce sellers, end customers, delivery riders, and system administrators who will interact with the platform's interfaces; testers who will verify functionalities like AI chat responses and route optimization; and documentation writers who will create user manuals and guides based on the system's features.

1.1 Problem Statement

Small-scale e-commerce sellers in regions like Pakistan rely heavily on Meta ads for generating leads and WhatsApp for handling customer interactions, but they face significant challenges in scaling their operations efficiently. Delayed or unclear responses to customer queries often result in high drop-off rates and lost sales opportunities. Repetitive inquiries about frequently asked questions (FAQs) overburden sellers, consuming valuable time that could be spent on growth activities. WhatsApp does not natively support bulk responding to customers, forcing manual handling that is time-intensive and error-prone. On the logistics front, manual assignment of delivery riders leads to inefficiencies, as orders are not distributed based on proximity or availability. Unoptimized delivery routes contribute to excessive fuel consumption, increased operational costs, and prolonged delivery times, frustrating customers and eroding trust. Inefficient integrations

with courier platforms mean sellers struggle to upload bulk orders to dashboards, causing further delays in processing. Additionally, managing Meta Ads manually without a secure, centralized system results in poor campaign optimization and lack of real-time performance insights. This approach also introduces risks, such as dependency on personal accounts, which can lead to security vulnerabilities and compliance issues. Collectively, these problems raise operational costs for small businesses, reduce customer retention rates, and hinder scalability in a highly competitive digital marketplace. Without automation, sellers cannot compete with larger e-commerce players who leverage advanced tools. The E-EAP addresses these pain points by introducing AI-driven automation to streamline workflows and enhance efficiency.

1.2 Scope

The scope of the E-EAP project encompasses the design and development of an intelligent, agentic AI-powered platform tailored for small-to-medium scale e-commerce sellers transitioning to digital operations. This system integrates seamlessly with WhatsApp Cloud API to enable instant, multilingual customer responses, automated follow-up funnels, and secure bulk messaging capabilities. It automates order confirmation directly through chat interfaces, reducing manual intervention and improving response times. The platform includes a dedicated delivery and shipment module that automates order assignments to riders or couriers, generates bulk upload files for courier dashboards, and optimizes routes using advanced algorithms like K-means clustering and Google OR-Tools VRP. Real-time tracking notifications keep customers informed about shipment status, enhancing satisfaction and transparency. For in-house deliveries, zone-based clustering ensures efficient rider assignments based on geographic proximity. The human agent panel, built as a mobile app, allows escalation of complex queries from AI chatbots to live agents, with features for chat reassignment and performance tracking. The admin dashboard provides a centralized web interface for monitoring key performance indicators (KPIs) such as order volumes, AI handling rates, and response times. Inventory management synchronizes stock levels with confirmed orders, issuing low-stock alerts to prevent overselling. The COD and finance module handles cash-on-delivery reconciliation, generates revenue reports, and offers analytics on sales trends. Secure integration with Meta Ads API enables business-level campaign creation, permission management, and ad performance reporting without personal account risks. The driver app equips riders with mobile access to optimized routes, order details, and COD submission tools. Overall, the boundaries focus on software-based automation for customer engagement, logistics, inventory, finance, and advertising, deployed scalably via Docker and Kubernetes. Physical product sourcing, hardware deployment, and non-Meta ad platforms are excluded from this scope. This ensures a user-friendly, cloud-based solution that directly

tackles inefficiencies in local e-commerce setups while remaining extensible for future enhancements.

1.3 Modules

The E-EAP is divided into distinct modules, grouped by system type where applicable: core backend modules for AI and integrations, mobile apps for agents and drivers, and web-based admin tools.

1.3.1 Module 1: Customer Interaction Module

This module handles all customer interactions through WhatsApp Cloud API using AI-powered chatbots, providing instant responses and personalization.

1. WhatsApp Cloud API integration with one registered business number.
2. AI chatbot (fine-tuned LLM) supporting English, Urdu/Hindi, Arabic.
3. Voice note transcription (Speech-to-Text e.g., Whisper) $\rightarrow$ text replies.
4. Order confirmation through chat.
5. Escalation of unhandled queries to human agent.
6. Real-time notifications for order confirmation, shipment updates, and tracking.

1.3.2 Module 2: Human Agent Panel (Mobile App)

This module, as a client mobile app, allows human agents to manage escalated chats and provide personalized support in real-time.

1. Secure login with multi-agent support.
2. Dedicated Inbox for assigned/escalated chats.
3. Real-time chat window continuing from bot history.
4. Manual order confirmation and inline actions.
5. Call notes logging and chat reassignment.
6. Performance tracking (resolved conversations, orders confirmed).

1.3.3 Module 3: Admin Dashboard (Web)

This admin web app serves as the central control panel for operations, monitoring, and analytics across the platform.

1. Home dashboard with KPIs (orders, AI vs human handled, response times).
2. Chat monitoring (ongoing and historical chats).
3. Agent management (add/remove, assign, performance tracking).
4. Orders management with courier/in-house shipment methods.
5. Campaign management with pre-approved WhatsApp templates.
6. Analytics (AI success rate, engagement, cancellations, order volume).
7. System & User Management (roles, WhatsApp setup, courier settings).

1.3.4 Module 4: Delivery & Shipment Module

This core backend module optimizes delivery logistics for both courier and in-house deliveries, integrating with external APIs.

1. Auto-generate courier Excel file for bulk uploads.
2. Tracking ID re-upload to system for customer updates.
3. Zone-based clustering using K-means for in-house orders.
4. Auto/manual driver assignment.
5. Route optimization using Google Maps and OR-Tools VRP.
6. Dynamic rescheduling for new orders or unavailable drivers.

1.3.5 Module 5: Driver App (Mobile)

This client mobile app enables drivers to view and complete deliveries efficiently with route guidance and reporting.

1. Login & profile management.
2. Assigned deliveries list with optimized route map.

3. Order details (address, COD, customer info).
4. Delivery confirmation with proof.
5. COD submission reports.
6. Performance tracking (completed orders, avg delivery time).

1.3.6 Module 6: Inventory Management Module

This backend module manages stock and product catalogs, ensuring synchronization and alerts for sellers.

1. Centralized product catalog.
2. Auto-sync stock with confirmed orders.
3. Low-stock alerts.
4. Add/update products (price, stock, description).
5. Track stock movement (daily usage, top products).

1.3.7 Module 7: COD & Finance Module

This backend module tracks financial transactions, COD collections, and provides revenue insights.

1. Daily COD reports (courier + in-house).
2. Pending vs collected COD.
3. Auto-adjustments for cancelled orders.
4. Revenue breakdown (product-wise, agent-wise).
5. Financial analytics (sales trends, COD trends, courier vs in-house).

1.3.8 Module 8: Meta Developer App & Access Token Module

This integration module securely connects with Meta Ads for campaign management and reporting.

1. Authentication gateway with secure Marketing API communication.

2. Business-level ownership under Meta Business account.
3. Permission management for campaigns, ad sets, and insights.
4. Security layer with scope control and token confidentiality.
5. Scalability to reuse setup across campaigns, pages, and accounts.

1.4 User Classes and Characteristics

The E-EAP anticipates several user classes, each with specific characteristics related to their roles in the e-commerce workflow.

User class	Description
Small-scale E-commerce Sellers	These are the primary users, typically owners or operators of local retail businesses transitioning to online sales, numbering in the hundreds for adoption in regions like Pakistan. They will use the platform to automate WhatsApp interactions, manage orders, monitor inventory, and run Meta ad campaigns. Expected usage is daily, with about 80% handling 10-50 orders per day via the admin dashboard and integrations. Most will require minimal training on setup, focusing on AI escalation and analytics for decision-making.
End Customers	Shoppers interacting via WhatsApp for inquiries, orders, and tracking, estimated at thousands per seller. They place orders through chat, receive instant responses and notifications, with 70% using mobile devices. No training needed; emphasis on seamless, multilingual support to ensure high satisfaction and repeat business.
Delivery Riders/Couriers	In-house riders or third-party couriers (e.g., 50-100 per seller network) who receive digital assignments via the mobile app. They update delivery status, collect COD, and follow optimized routes. Usage is shift-based, averaging 20-40 deliveries daily; basic smartphone proficiency required, with training on app features like proof uploads.
System Administrators	Platform overseers (1-5 per deployment) responsible for monitoring KPIs, managing users/agents, and configuring integrations like WhatsApp and Meta APIs. They access the web dashboard for analytics and maintenance, with advanced technical skills; usage is continuous for oversight in multi-seller environments.

Chapter 2

Project Requirements

This chapter describes the functional and non-functional requirements of the project.

2.1 Use cases and Representation

The **Agentic AI-Driven E-Commerce Automation and Engagement Platform (E-EAP)** is an interactive end-user application involving multiple user interfaces (web dashboard, mobile apps, WhatsApp chatbot) and real-time system interactions. Therefore, the **Use Case Method** is the most appropriate requirements gathering technique to model system functionalities, user interactions, and workflows.

2.2 Identified Actors

Actor	Description
End Customer	Interacts with the business primarily through WhatsApp chatbot to browse products, place orders, and track deliveries.
Human Agent	Uses the mobile panel to handle escalated customer queries and confirm orders manually if required.
Admin	Manages users, agents, campaigns, analytics, inventory, orders, and courier integrations via the web dashboard.
Delivery Rider	Uses the driver mobile app to view assigned deliveries, follow optimized routes, and report completed deliveries.
System (AI + Backend)	Executes automated responses, order processing, route optimization, notifications, and API integrations.
Meta API	External service for campaign creation, management, and performance retrieval.
Courier Service API	External service to upload and track shipment orders.

Table 2.1: List of Identified Actors

2.3 Main Use Cases

- UC-1: Place Order via WhatsApp
- UC-2: Escalate Query to Human Agent
- UC-3: Manage Orders (Admin)
- UC-4: Assign and Optimize Delivery Routes
- UC-5: Complete Delivery and Submit COD
- UC-6: Manage Inventory
- UC-7: Manage Meta Ad Campaigns
- UC-8: Generate Reports and Analytics
- UC-9: Manage System Users and Roles

2.4 Use Case Diagram

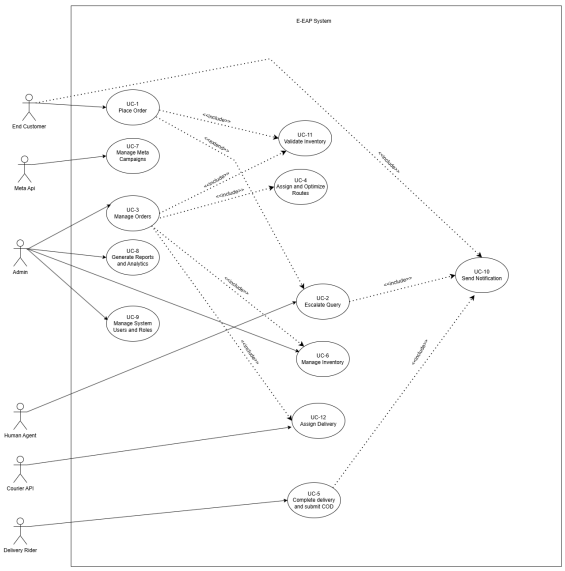

Figure 2.1: Use Case Diagram for E-EAP

2.5 Detailed Use Case Descriptions

UC-1: Place Order via WhatsApp

Actors	End Customer, System
Preconditions	Customer has initiated a chat with the business WhatsApp number.
Postconditions	Order is confirmed and stored in the system. Inventory and delivery workflows are triggered.
Main Flow	1. Customer sends inquiry or selects a product. 2. System displays product info, price, and availability. 3. Customer confirms order. 4. System records order and sends confirmation. 5. Inventory module updates stock.
Alternative Flow	Low stock → suggest alternatives. Unrecognized message → escalate to human agent.
Exceptions	Network/API failure → message retry or error alert.

UC-2: Escalate Query to Human Agent

Actors	End Customer, Human Agent, System
Preconditions	AI chatbot is unable to handle the query confidently.
Postconditions	Human agent continues the conversation and resolves the query.
Main Flow	1. AI chatbot flags unhandled query. 2. System routes chat to available agent. 3. Agent joins conversation. 4. Agent resolves query.
Alternative Flow	Agent reassigns to another agent.
Exceptions	No agent available → automated fallback response.

UC-3: Manage Orders (Admin)

Actors	Admin
Preconditions	Orders exist in the system.
Postconditions	Orders are processed, updated, or cancelled.
Main Flow	1. Admin logs in to dashboard. 2. Views list of orders with statuses. 3. Updates or cancels orders. 4. Assigns shipment method.
Alternative Flow	Filters and searches orders.
Exceptions	Invalid status update.

UC-4: Assign and Optimize Delivery Routes

Actors	Admin, System, Delivery Rider
Preconditions	Orders are confirmed and marked for delivery.
Postconditions	Optimized delivery routes are assigned to drivers.
Main Flow	1. Admin triggers route assignment. 2. System clusters orders (K-means). 3. OR-Tools computes optimized routes. 4. Riders receive routes.
Alternative Flow	Manual assignment.
Exceptions	API error in route calculation.

UC-5: Complete Delivery and Submit COD

Actors	Delivery Rider
Preconditions	Rider has received assigned deliveries.
Postconditions	Delivery status updated and COD submitted.
Main Flow	1. Rider logs into mobile app. 2. Views assigned deliveries. 3. Navigates to location. 4. Marks delivery complete with proof. 5. Submits COD report.
Alternative Flow	Failed delivery $\rightarrow$ mark as unsuccessful.
Exceptions	Offline updates stored until online.

UC-6: Manage Inventory

Actors	Admin
Preconditions	Products exist in catalog.
Postconditions	Inventory is synchronized with orders.
Main Flow	1. Admin logs in. 2. Adds/updates product details. 3. Inventory auto-updates after order confirmation. 4. Low stock alerts generated.
Alternative Flow	Bulk update via file upload.
Exceptions	Invalid product ID.

UC-7: Manage Meta Ad Campaigns

Actors	Admin, Meta API
Preconditions	Meta account is connected and authenticated.
Postconditions	Campaigns are created and monitored.
Main Flow	1. Admin connects Meta account. 2. Creates or edits campaigns. 3. Monitors performance metrics.
Alternative Flow	Reconnect token if expired.
Exceptions	Meta API error.

UC-8: Generate Reports and Analytics

Actors	Admin
Preconditions	Orders, deliveries, and campaign data exist.
Postconditions	Reports are generated for performance tracking.
Main Flow	1. Admin opens dashboard. 2. Selects analytics module. 3. Generates reports.
Alternative Flow	Export to Excel or PDF.
Exceptions	Data aggregation error.

UC-9: Manage System Users and Roles

Actors	Admin
Preconditions	Admin has appropriate permissions.
Postconditions	Users and their roles updated in system.
Main Flow	1. Admin logs in. 2. Views users list. 3. Adds/updates/deletes users. 4. Assigns roles.
Alternative Flow	Password reset for users.
Exceptions	Insufficient privileges.

Summary: The use case methodology effectively models interactive workflows in E-EAP. It covers all critical system functionalities across chatbot, delivery, admin, finance, inventory, and Meta Ads modules, enabling clear mapping to functional requirements for implementation and testing.

2.6 Functional Requirements

This section describes the functional requirements of the system expressed in the natural language style. This section is organized by module, with specific functional requirements associated with each module's features.

2.6.1 Module 1: Customer Interaction Module

Following are the requirements for Module 1:

1. FR1.1: The system shall integrate with WhatsApp Cloud API using one registered business number to receive and send messages.
2. FR1.2: The system shall deploy an AI chatbot using a fine-tuned LLM to support interactions in English, Urdu/Hindi, and Arabic.
3. FR1.3: The system shall transcribe incoming voice notes using Speech-to-Text (e.g., Whisper) and generate text-based replies.
4. FR1.4: The system shall confirm orders directly through the chat interface, capturing details like product, quantity, and address.
5. FR1.5: The system shall escalate unhandled or complex queries to a human agent via the Human Agent Panel.

6. FR1.6: The system shall send real-time notifications for order confirmation, shipment updates, and tracking via WhatsApp.

2.6.2 Module 2: Human Agent Panel (Mobile App)

Following are the requirements for Module 2:

1. FR2.1: The system shall provide secure login functionality with multi-agent support using authentication tokens.
2. FR2.2: The system shall display a dedicated inbox for assigned or escalated chats from the AI module.
3. FR2.3: The system shall offer a real-time chat window that continues from the bot's conversation history.
4. FR2.4: The system shall enable manual order confirmation and inline actions (e.g., edit details) within the chat.
5. FR2.5: The system shall allow logging of call notes and reassignment of chats to other agents.
6. FR2.6: The system shall track agent performance metrics, including resolved conversations and confirmed orders.

2.6.3 Module 3: Admin Dashboard (Web)

Following are the requirements for Module 3:

1. FR3.1: The system shall display a home dashboard with key performance indicators (KPIs) such as total orders, AI vs. human handled chats, and average response times.
2. FR3.2: The system shall provide monitoring tools for ongoing and historical chats, including search and filtering.
3. FR3.3: The system shall support agent management functions, including add/remove agents, assignment, and performance tracking.
4. FR3.4: The system shall manage orders with options for courier or in-house shipment methods.
5. FR3.5: The system shall handle campaign management using pre-approved WhatsApp message templates.

6. FR3.6: The system shall generate analytics reports on AI success rate, customer engagement, cancellations, and order volume.
7. FR3.7: The system shall include system and user management for roles, WhatsApp setup, and courier configurations.

2.6.4 Module 4: Delivery & Shipment Module

Following are the requirements for Module 4:

1. FR4.1: The system shall auto-generate Excel files for bulk courier uploads containing order details.
2. FR4.2: The system shall allow re-upload of tracking IDs from couriers to update customer notifications.
3. FR4.3: The system shall perform zone-based clustering using K-means algorithm for in-house orders.
4. FR4.4: The system shall support auto or manual driver assignment based on availability and location.
5. FR4.5: The system shall optimize routes using Google Maps API and OR-Tools Vehicle Routing Problem (VRP) solver.
6. FR4.6: The system shall dynamically reschedule routes for new orders or unavailable drivers.

2.6.5 Module 5: Driver App (Mobile)

Following are the requirements for Module 5:

1. FR5.1: The system shall provide login and profile management for drivers with secure authentication.
2. FR5.2: The system shall display a list of assigned deliveries with an integrated optimized route map.
3. FR5.3: The system shall show detailed order information, including address, COD amount, and customer details.
4. FR5.4: The system shall allow delivery confirmation with photo proof upload.
5. FR5.5: The system shall enable submission of COD collection reports.

6. FR5.6: The system shall track driver performance, including completed orders and average delivery time.

2.6.6 Module 6: Inventory Management Module

Following are the requirements for Module 6:

1. FR6.1: The system shall maintain a centralized product catalog with searchable entries.
2. FR6.2: The system shall automatically synchronize stock levels with confirmed orders.
3. FR6.3: The system shall issue low-stock alerts via notifications to admins/agents.
4. FR6.4: The system shall allow adding/updating products with fields for price, stock quantity, and description.
5. FR6.5: The system shall track stock movement, including daily usage and top-selling products.

2.6.7 Module 7: COD & Finance Module

Following are the requirements for Module 7:

1. FR7.1: The system shall generate daily COD reports for both courier and in-house deliveries.
2. FR7.2: The system shall track pending vs. collected COD amounts in real-time.
3. FR7.3: The system shall automatically adjust records for cancelled orders.
4. FR7.4: The system shall provide revenue breakdowns by product and agent.
5. FR7.5: The system shall offer financial analytics, including sales trends, COD trends, and courier vs. in-house comparisons.

2.6.8 Module 8: Meta Developer App & Access Token Module

Following are the requirements for Module 8:

1. FR8.1: The system shall implement an authentication gateway for secure communication with Meta Marketing API.

2. FR8.2: The system shall establish business-level ownership under a Meta Business account.

3. FR8.3: The system shall manage permissions for campaigns, ad sets, and insights access.

4. FR8.4: The system shall include a security layer with scope control and token confidentiality measures.

5. FR8.5: The system shall support scalability to reuse the setup across multiple campaigns, pages, and accounts.

2.7 Non-Functional Requirements

This section specifies nonfunctional requirements. These quality requirements should be specific, quantitative, and verifiable.

2.7.1 Reliability

The system shall achieve a Mean Time Between Failures (MTBF) of at least 99.9% uptime over a 24-hour period, measured monthly. A failure is defined as any unhandled crash in core services (e.g., API downtime >5 minutes). To protect against failure, the system shall implement redundant cloud deployment with automatic failover. Error detection shall use logging and monitoring tools to alert admins within 1 minute of occurrence. Correction strategy includes automated rollbacks and manual intervention within 15 minutes for critical issues.

2.7.2 Usability

Usability requirements deal with ease of learning, ease of use, error avoidance and recovery, the efficiency of interactions, and accessibility.

USE-1: The admin dashboard shall allow a user to view order KPIs with a single dashboard load, requiring no more than two clicks for drill-down details.

USE-2: The mobile apps (Human Agent Panel and Driver App) shall support gesture-based navigation (e.g., swipe to reassign chats or confirm deliveries) and be accessible via screen readers for visually impaired users.

USE-3: The AI chatbot shall achieve a 95% first-response accuracy rate for FAQs, with escalation paths clearly indicated to users.

2.7.3 Performance

PER-1: 95% of WhatsApp chatbot responses shall be generated and delivered within 2 seconds over a standard 4G connection.

PER-2: Route optimization computations shall complete within 10 seconds for up to 50 orders using OR-Tools.

PER-3: The admin dashboard shall load all KPIs and analytics within 3 seconds for datasets up to 10,000 orders.

2.7.4 Security

SEC-1: The system shall resist unauthorized access attempts requiring at least 10,000 brute-force tries or equivalent skill level to breach API tokens.

SEC-2: All data transmissions (e.g., customer chats, COD details) shall use end-to-end encryption, with audit logs retained for 90 days.

SEC-3: Meta Ads integration shall enforce least-privilege access, limiting token scopes to necessary permissions only.

2.8 Domain Model

The domain model represents key entities, attributes, and relationships in the E-EAP system, including classes like Customer, Order, Product, Driver, Chat, Campaign, and their associations (e.g., Order 1–* Product, Chat 1–1 Customer).

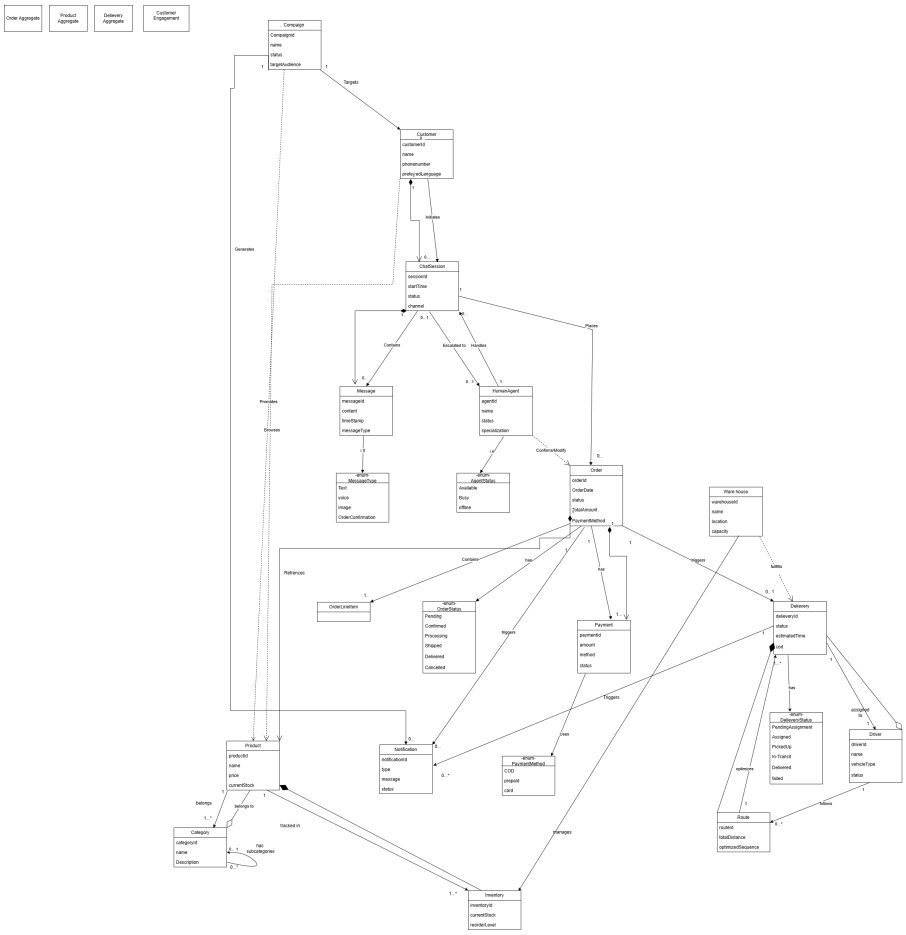

Figure 2.2: Domain Model (Class Diagram) for E-EAP

Chapter 3

System Design and Architecture

This chapter presents the comprehensive system design and architecture of the E-EAP (E-Commerce Enterprise Automation Platform) system. It includes detailed UML diagrams covering class structure, activity flows, state transitions, system sequences, and domain-specific processes.

3.1 Class Diagram

The class diagram represents the static structure of the E-EAP system, showing the system's classes, their attributes, methods, and the relationships among objects.

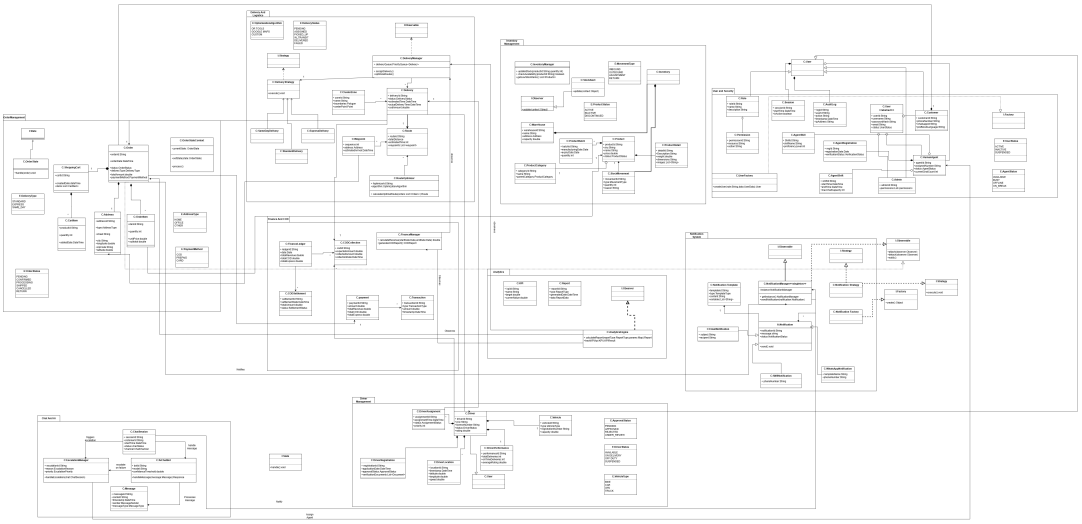

Figure 3.1: Class Diagram of E-EAP System

Description: The class diagram illustrates the core domain models and their relationships in the E-EAP system. Key components include:

- **User Management:** Includes User, Customer, Admin, HumanAgent, and Driver classes with role-based access control
- **Order Management:** Comprises Order, OrderItem, ShoppingCart, and related state management classes
- **Delivery & Logistics:** Covers Delivery, Route, Vehicle, and optimization components
- **Inventory Management:** Includes Product, Inventory, Warehouse, and stock movement tracking
- **Finance & Payments:** Handles Payment, CODCollection, and financial transactions
- **Chat & AI System:** Manages ChatSession, Message, and AI chatbot functionality
- **Notification System:** Implements various notification channels using Strategy pattern
- **Analytics & Reporting:** Provides reporting capabilities with Observer pattern

The diagram demonstrates the use of key design patterns including:

- **Factory Pattern:** UserFactory for creating different user types
- **State Pattern:** OrderState for managing order lifecycle
- **Strategy Pattern:** DeliveryStrategy for different delivery types
- **Observer Pattern:** For notification and analytics systems
- **Singleton Pattern:** NotificationManager for centralized notification handling...

3.2 Activity Diagrams

Activity diagrams illustrate the business workflows and operational processes within the E-EAP system, showing the flow of control from one activity to another.

3.2.1 UC-1: Customer Order Placement via WhatsApp

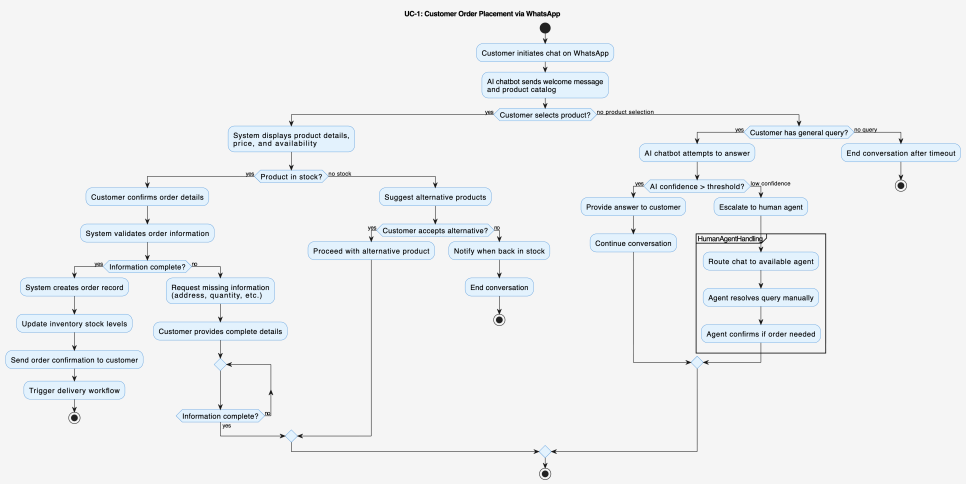

Figure 3.2: Activity Diagram for Customer Order Placement via WhatsApp

Description: This activity diagram illustrates the complete order placement process through WhatsApp, starting from customer initiation to order confirmation and inventory updates. The process includes product selection, stock validation, order creation, and confirmation messaging.

3.2.2 UC-4/UC-5: Order Fulfillment & Delivery Process

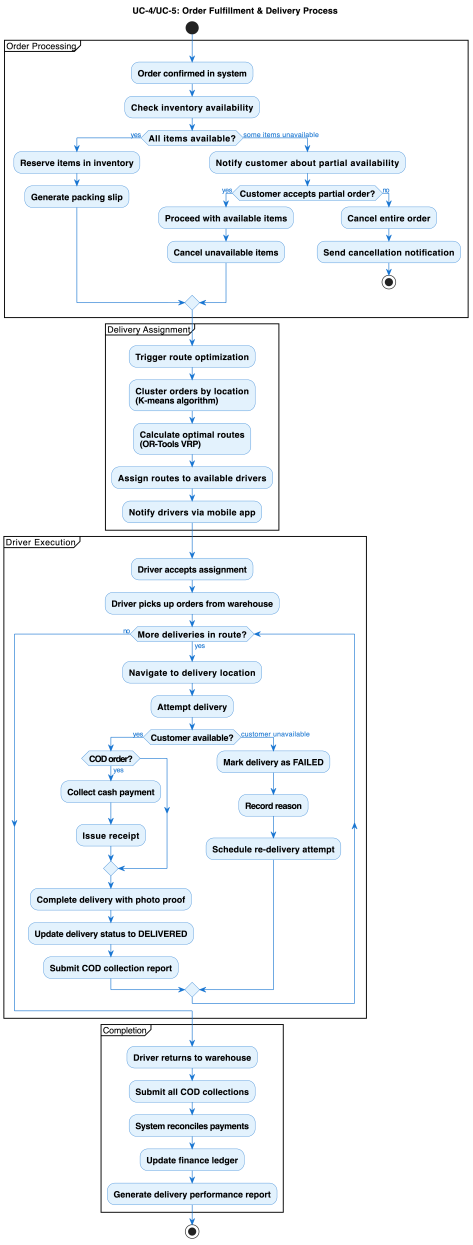

Figure 3.3: Activity Diagram for Order Fulfillment and Delivery Process

Description: Shows the end-to-end order fulfillment workflow including inventory checking, route optimization using K-means clustering and OR-Tools VRP, driver assignment, and delivery execution with COD collection and proof submission.

3.2.3 UC-2: Chat Escalation to Human Agent

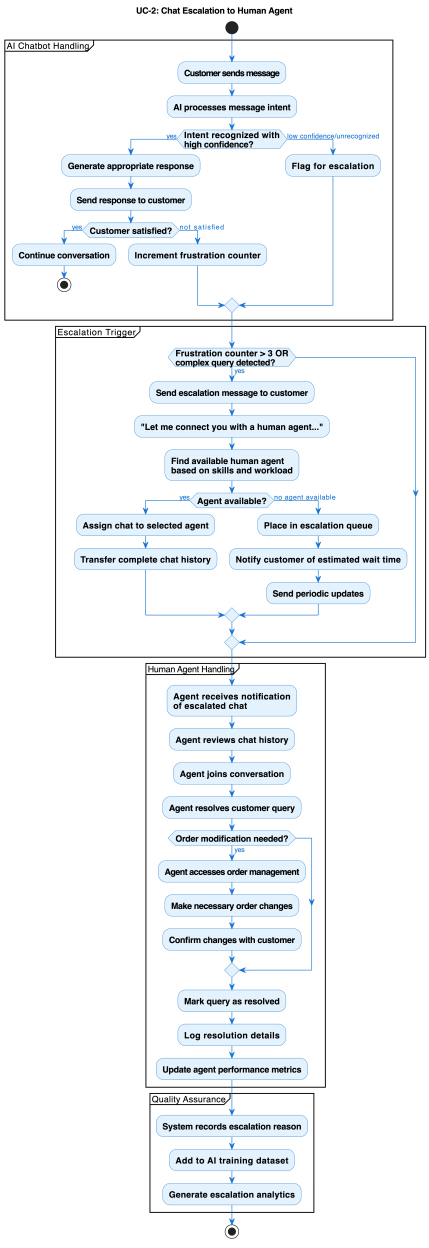

Figure 3.4: Activity Diagram for Chat Escalation to Human Agent

Description: Details the escalation process from AI chatbot to human agent based on confidence thresholds, customer frustration detection (after 3 failed responses), and complex query identification. Includes agent assignment based on skills and workload.

3.2.4 UC-4: Route Optimization & Driver Assignment

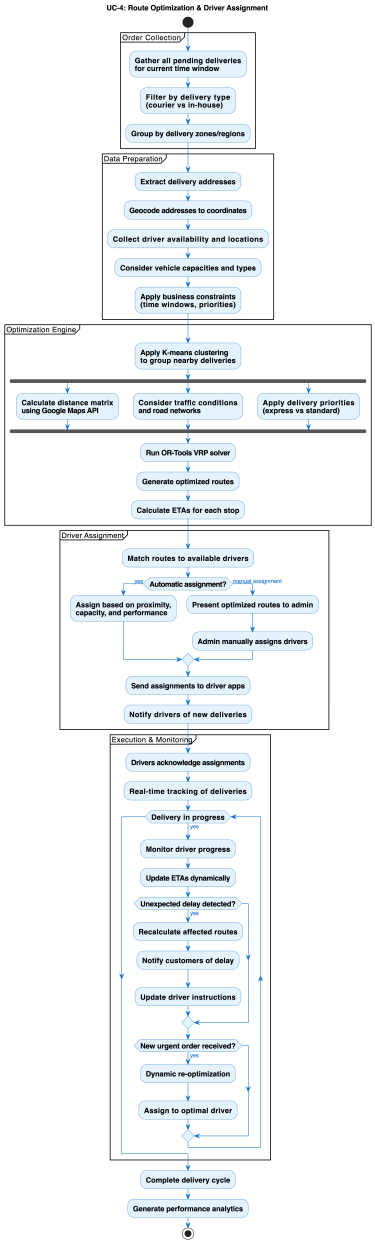

Figure 3.5: Activity Diagram for Route Optimization and Driver Assignment

Description: Illustrates the intelligent route planning process using K-means clustering for delivery grouping and OR-Tools VRP solver for optimal route calculation. Includes real-time monitoring and dynamic re-optimization for unexpected delays.

3.2.5 UC-6: Inventory Management Workflow

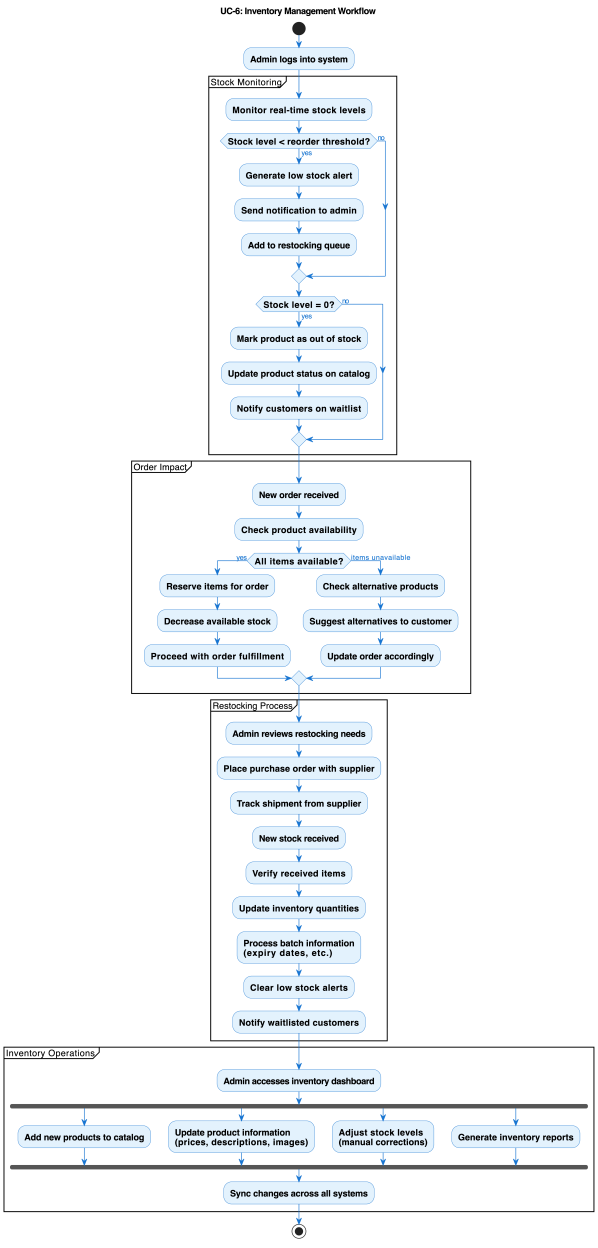

Figure 3.6: Activity Diagram for Inventory Management Workflow

Description: Shows the complete inventory management process including real-time stock monitoring, low stock alerts, restocking procedures, and order impact handling. Includes manual stock adjustments and movement tracking.

3.2.6 UC-3/UC-8/UC-9: Admin Dashboard Operations

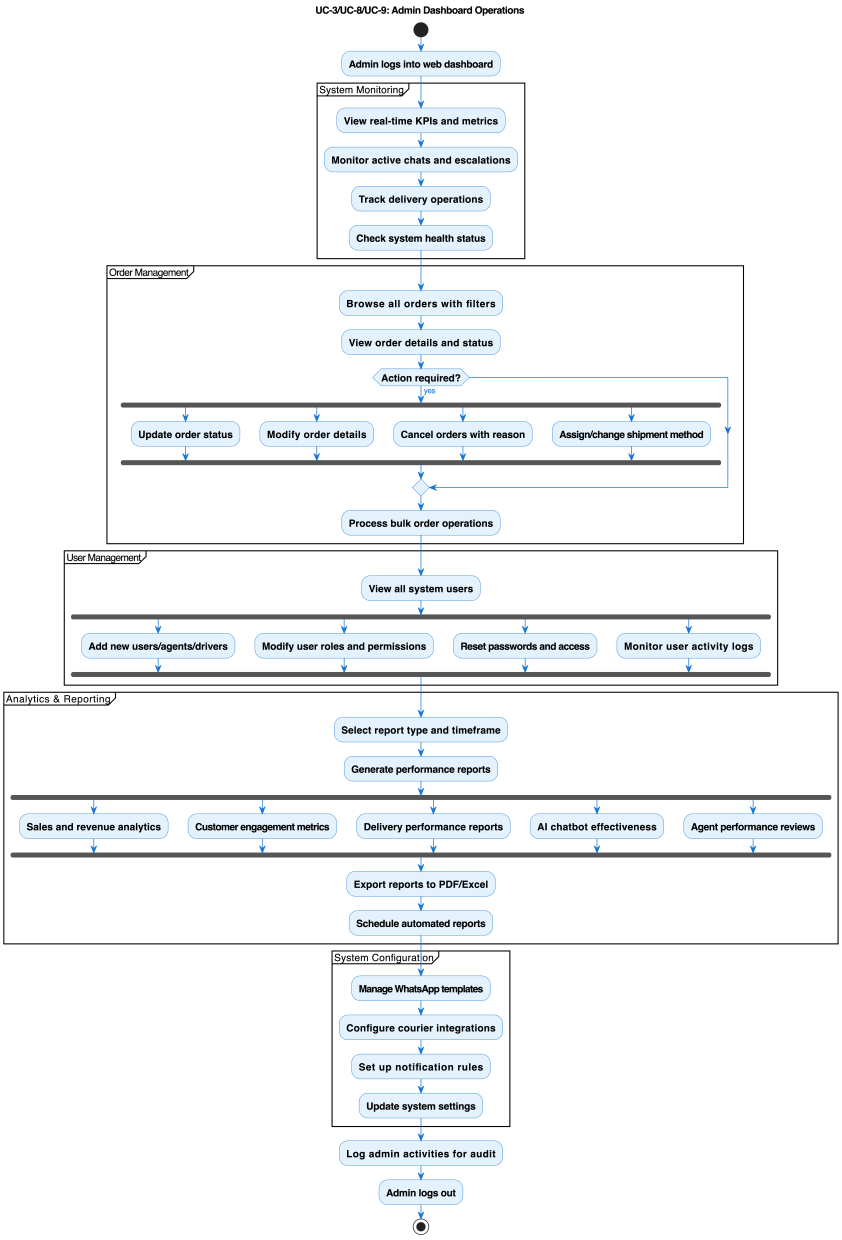

Figure 3.7: Activity Diagram for Admin Dashboard Operations

Description: Details the comprehensive admin dashboard functionalities including system monitoring, order management, user management, analytics reporting, and system configuration with audit logging.

3.3 State Transition Diagrams

State transition diagrams model the lifecycle of key system entities, showing how they respond to various events and transition between different states.

3.3.1 Order State Transition Diagram

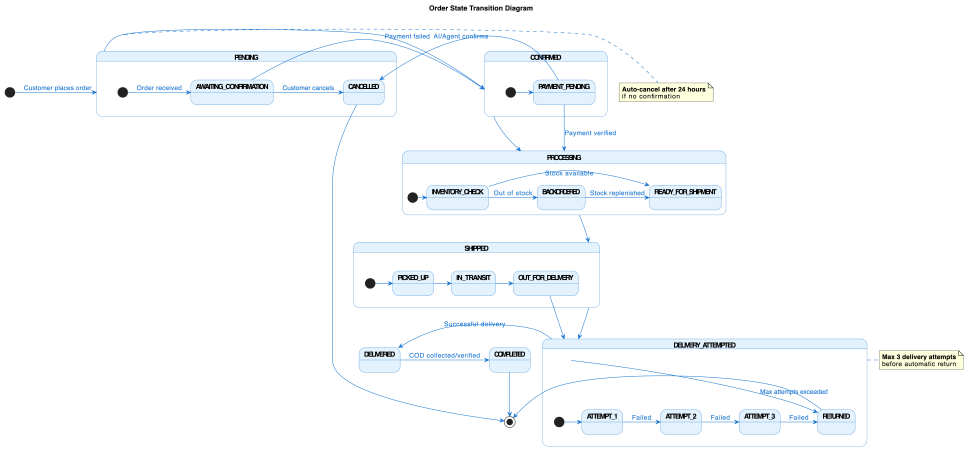

Figure 3.8: Order State Transition Diagram

Description: Shows the complete lifecycle of an order from PENDING to DELIVERED/COMPLETED or CANCELLED/RETURNED states. Includes automatic cancellation after 24 hours of no confirmation and maximum 3 delivery attempts before automatic return.

3.3.2 Delivery State Transition Diagram

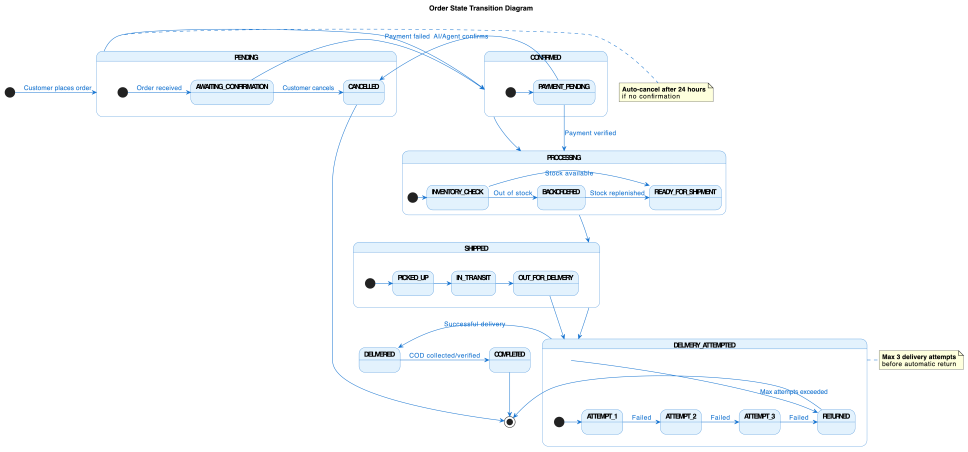

Figure 3.9: Delivery State Transition Diagram

Description: Illustrates the delivery process states including multiple attempt handling, failure scenarios, and successful completion with COD verification.

3.3.3 Chat Session State Transition Diagram

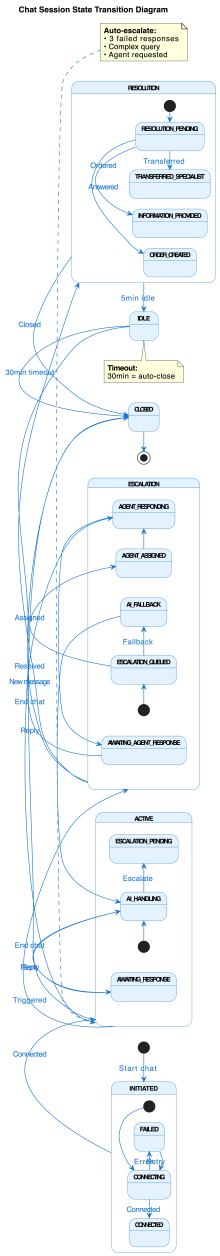

Figure 3.10: State Transition Diagram for Chat Escalation

Description: Details the various states of a chat session from initiation to resolution with escalation pathways. Includes auto-escalation triggers and 30-minute timeout for automatic closure.

3.3.4 Payment State Transition Diagram

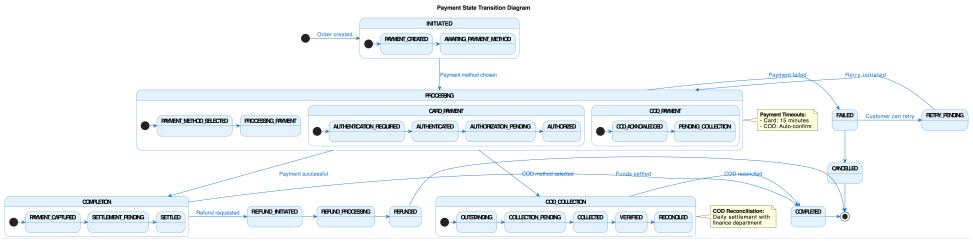

Figure 3.11: Payment State Transition Diagram

Description: Shows the payment processing states for both card payments (with 15-minute timeout) and COD methods with daily settlement and reconciliation flows.

3.3.5 Driver State Transition Diagram

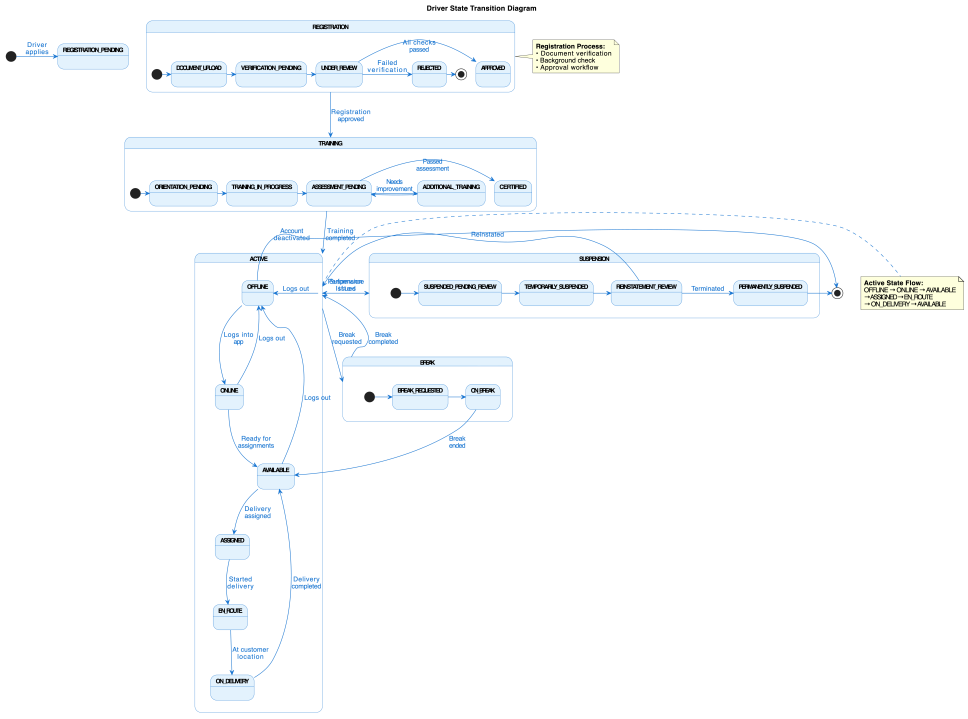

Figure 3.12: Driver State Transition Diagram

Description: Illustrates driver lifecycle from registration and training through active delivery states to suspension scenarios and performance management.

3.3.6 Inventory State Transition Diagram

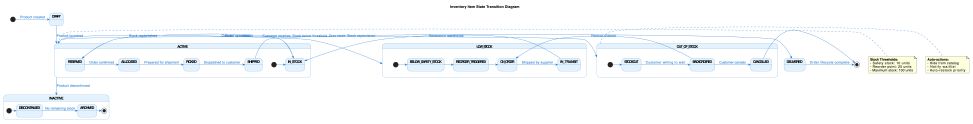

Figure 3.13: Inventory State Transition Diagram

Description: Shows inventory item states from active stock management through reordering triggers to discontinuation and archiving.

3.3.7 User State Transition Diagram

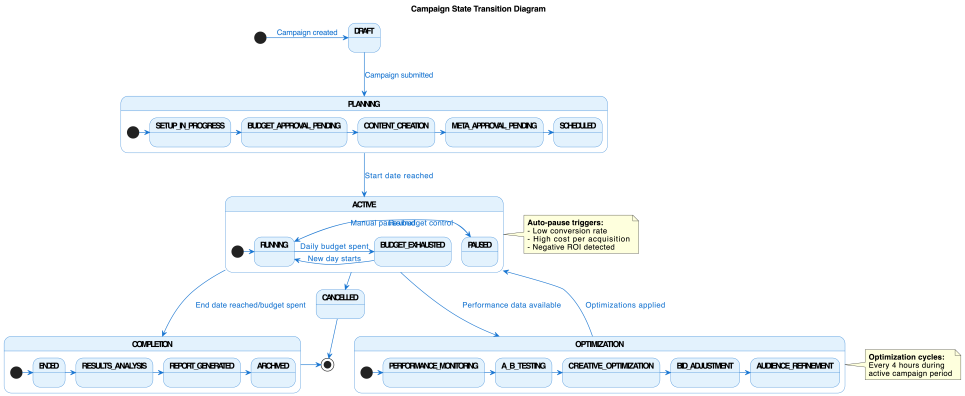

Figure 3.14: User State Transition Diagram

Description: Illustrates the user account lifecycle from registration and verification through active usage to suspension, deactivation, and eventual deletion.

3.4 System Sequence Diagrams (SSD)

System sequence diagrams depict the interactions between external actors and the system for specific use cases, showing the sequence of messages exchanged.

3.4.1 SSD-1: Place Order via WhatsApp

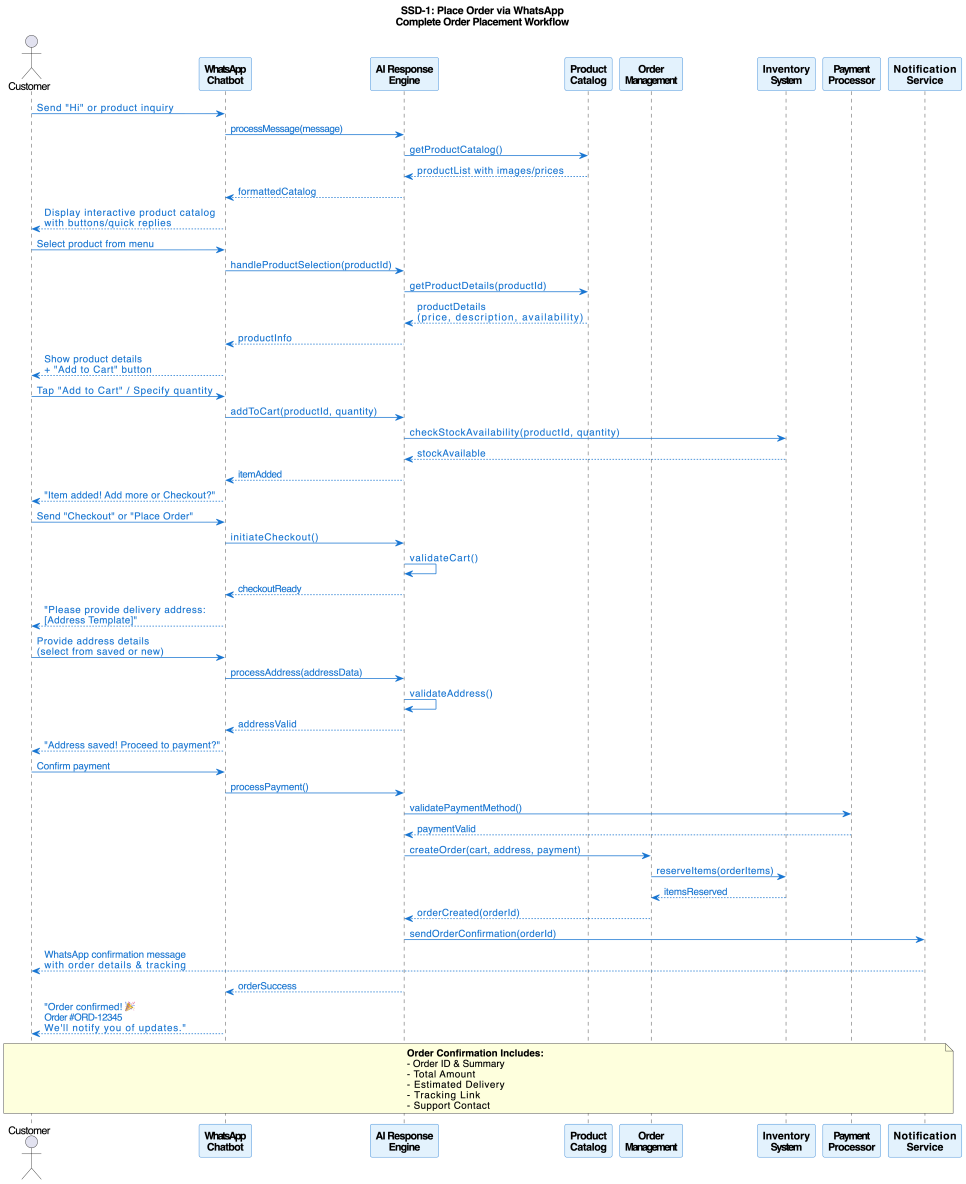

Figure 3.15: System Sequence Diagram for Place Order via WhatsApp

Actors: Customer, WhatsApp Chatbot, AI Response Engine, Product Catalog, Order Management, Inventory System, Payment Processor, Notification Service

3.4.2 SSD-2: Track Order Status

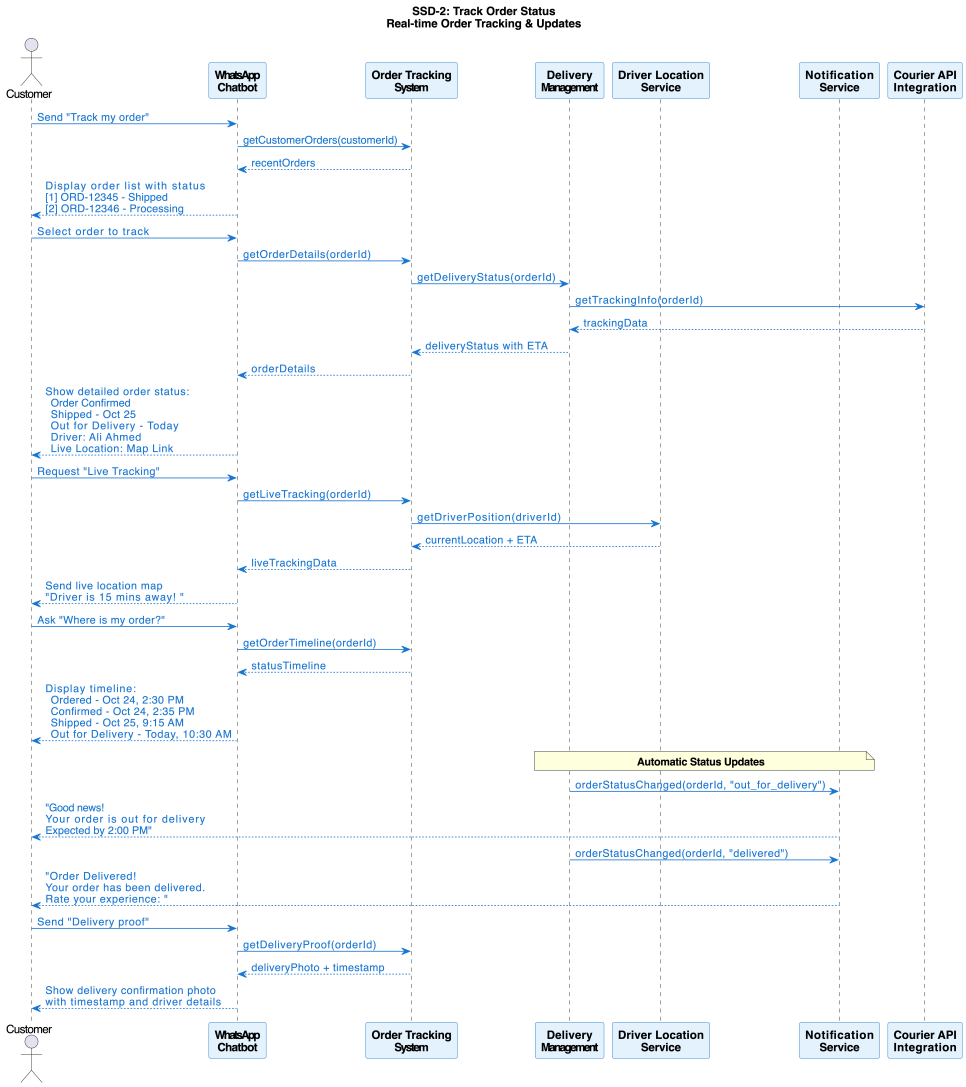

Figure 3.16: System Sequence Diagram for Track Order Status

Actors: Customer, WhatsApp Chatbot, Order Tracking System, Delivery Management, Driver Location Service, Notification Service, Courier API Integration

3.4.3 SSD-3: Customer Query & Chat Escalation

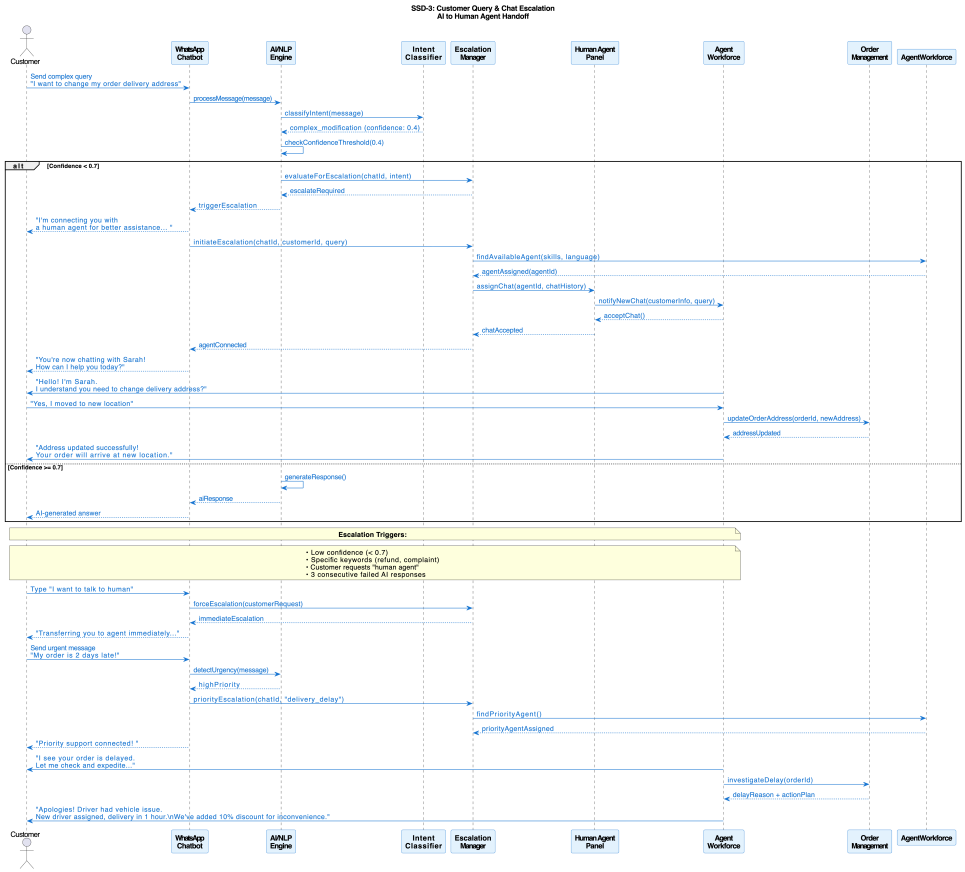

Figure 3.17: System Sequence Diagram for Customer Query and Chat Escalation

Actors: Customer, WhatsApp Chatbot, AI/NLP Engine, Intent Classifier, Escalation Manager, Human Agent Panel, Agent Workforce, Order Management

3.4.4 SSD-4: Customer Registration & Profile Management

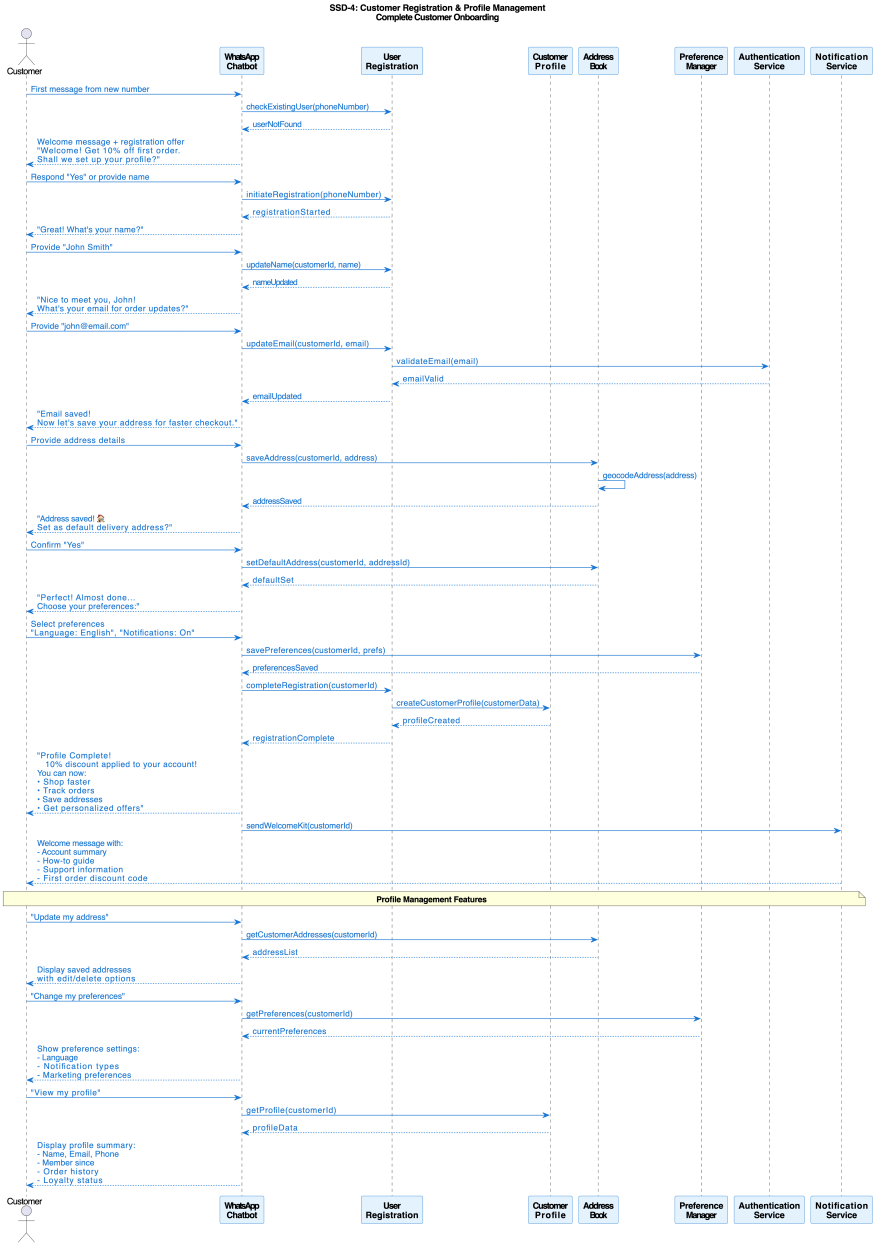

Figure 3.18: System Sequence Diagram for Customer Registration and Profile Management

Actors: Customer, WhatsApp Chatbot, User Registration, Customer Profile, Address Book, Preference Manager, Authentication Service, Notification Service

3.4.5 SSD-5: Assign & Optimize Delivery Routes

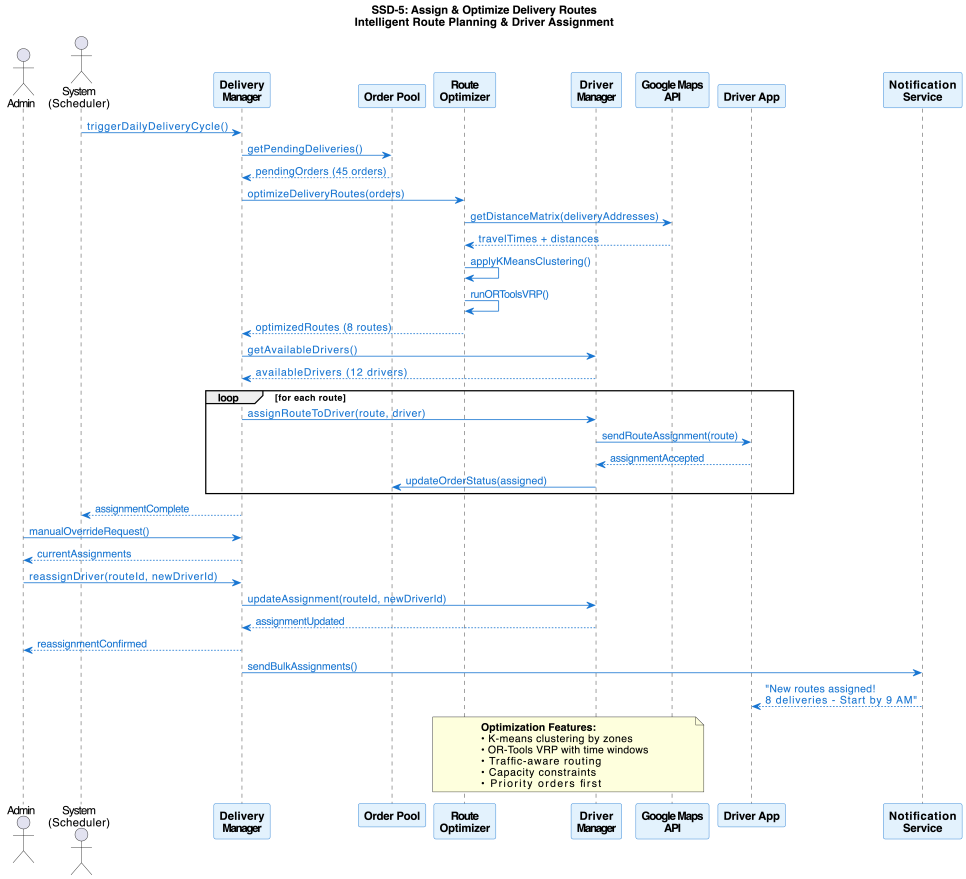

Figure 3.19: System Sequence Diagram for Assign and Optimize Delivery Routes

Actors: Admin, System Scheduler, Delivery Manager, Order Pool, Route Optimizer, Driver Manager, Google Maps API, Driver App, Notification Service

3.4.6 SSD-6: Complete Delivery & COD Collection

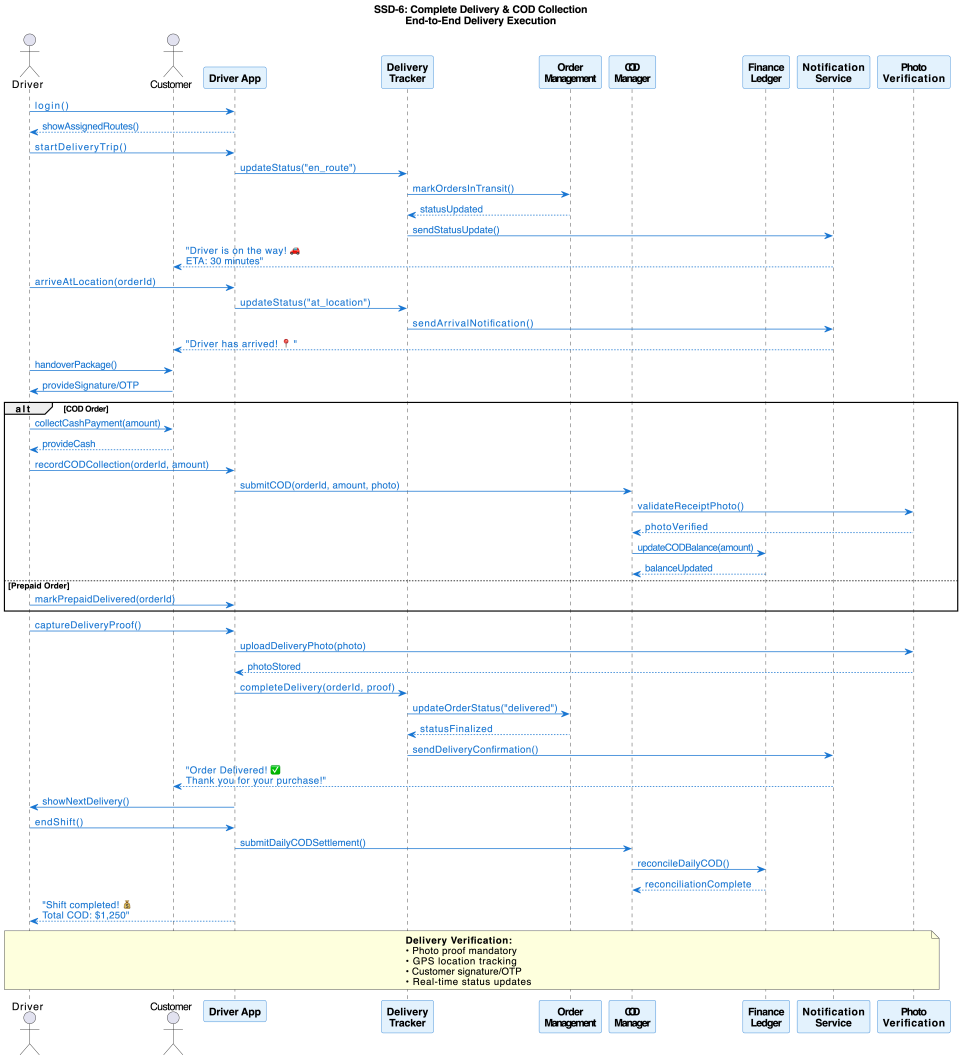

Figure 3.20: System Sequence Diagram for Complete Delivery and COD Collection

Actors: Driver, Customer, Driver App, Delivery Tracker, Order Management, COD Manager, Finance Ledger, Notification Service, Photo Verification Service

3.4.7 SSD-7: Handle Failed Delivery & Returns

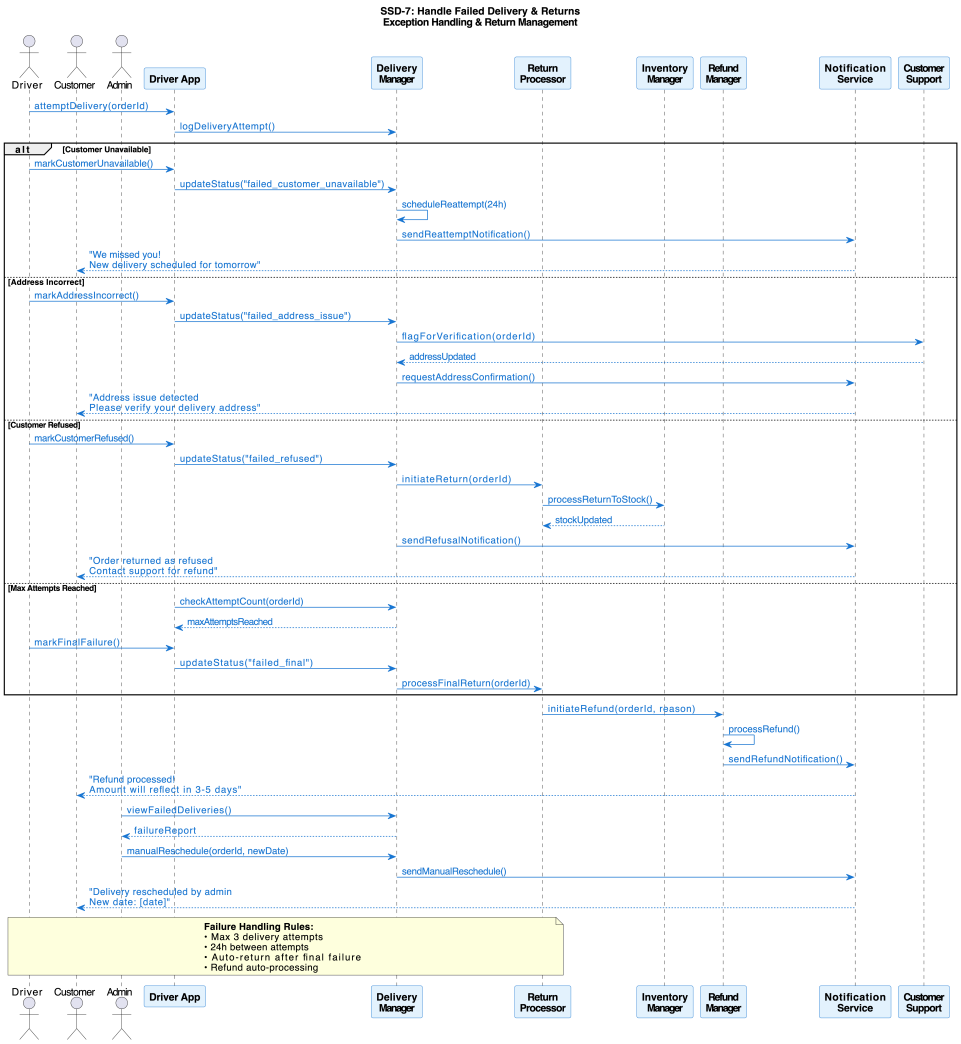

Figure 3.21: System Sequence Diagram for Handle Failed Delivery and Returns

Actors: Driver, Customer, Admin, Driver App, Delivery Manager, Return Processor, Inventory Manager, Refund Manager, Notification Service, Customer Support

3.4.8 SSD-8: Manage Inventory & Stock Updates

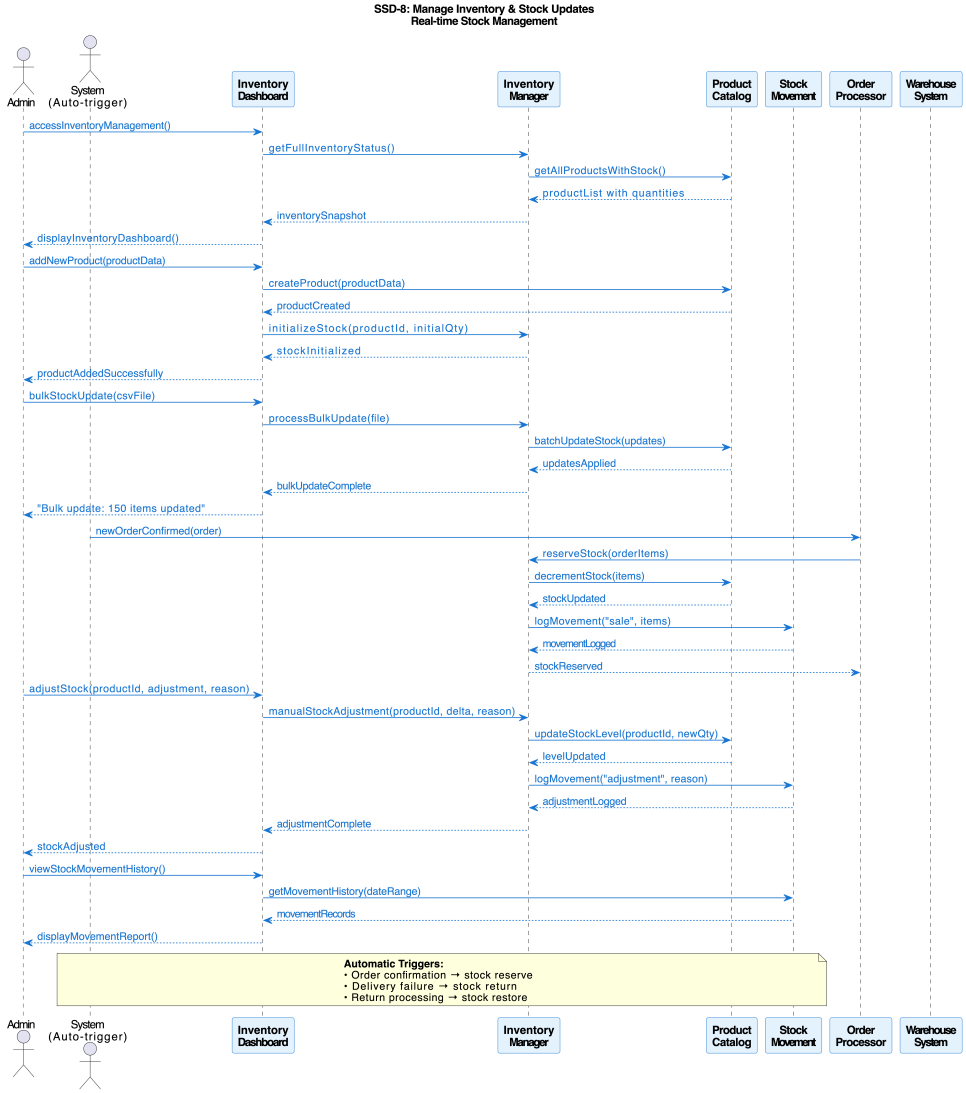

Figure 3.22: System Sequence Diagram for Manage Inventory and Stock Updates

Actors: Admin, System Auto-trigger, Inventory Dashboard, Inventory Manager, Product Catalog, Stock Movement, Order Processor, Warehouse System

3.4.9 SSD-9: Low Stock Alert & Restocking Process

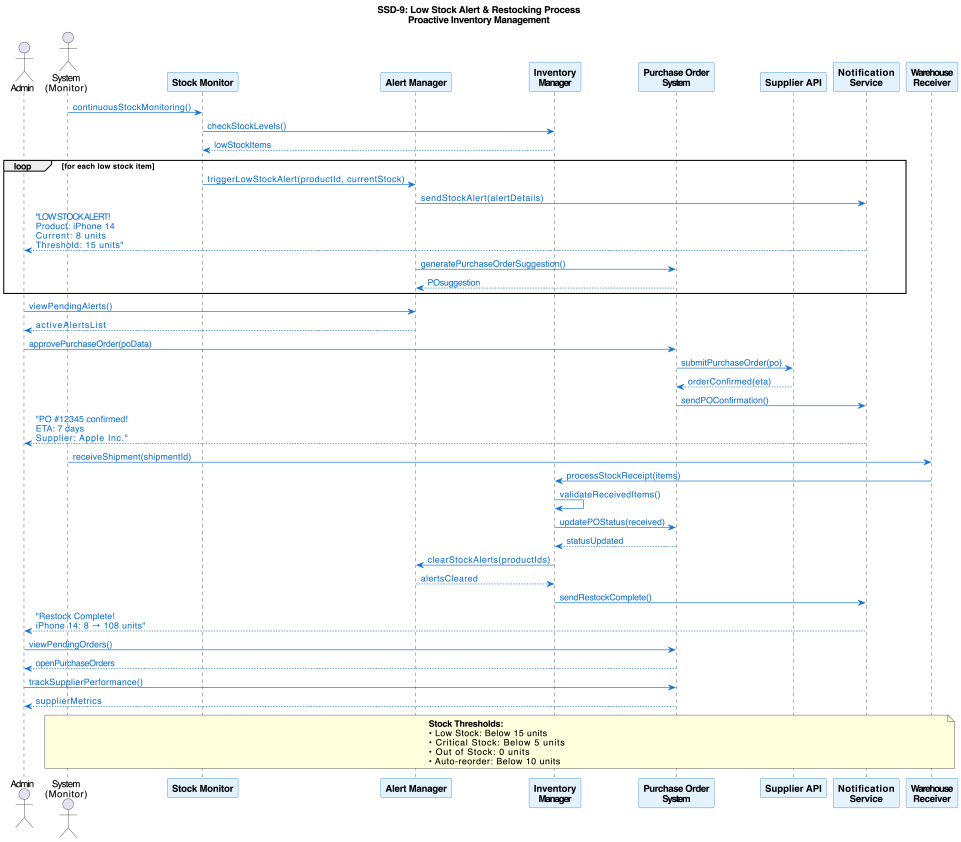

Figure 3.23: System Sequence Diagram for Low Stock Alert and Restocking Process

Actors: Admin, System Monitor, Stock Monitor, Alert Manager, Inventory Manager, Purchase Order System, Supplier API, Notification Service, Warehouse Receiver

3.4.10 SSD-10: Manage Meta Ad Campaigns

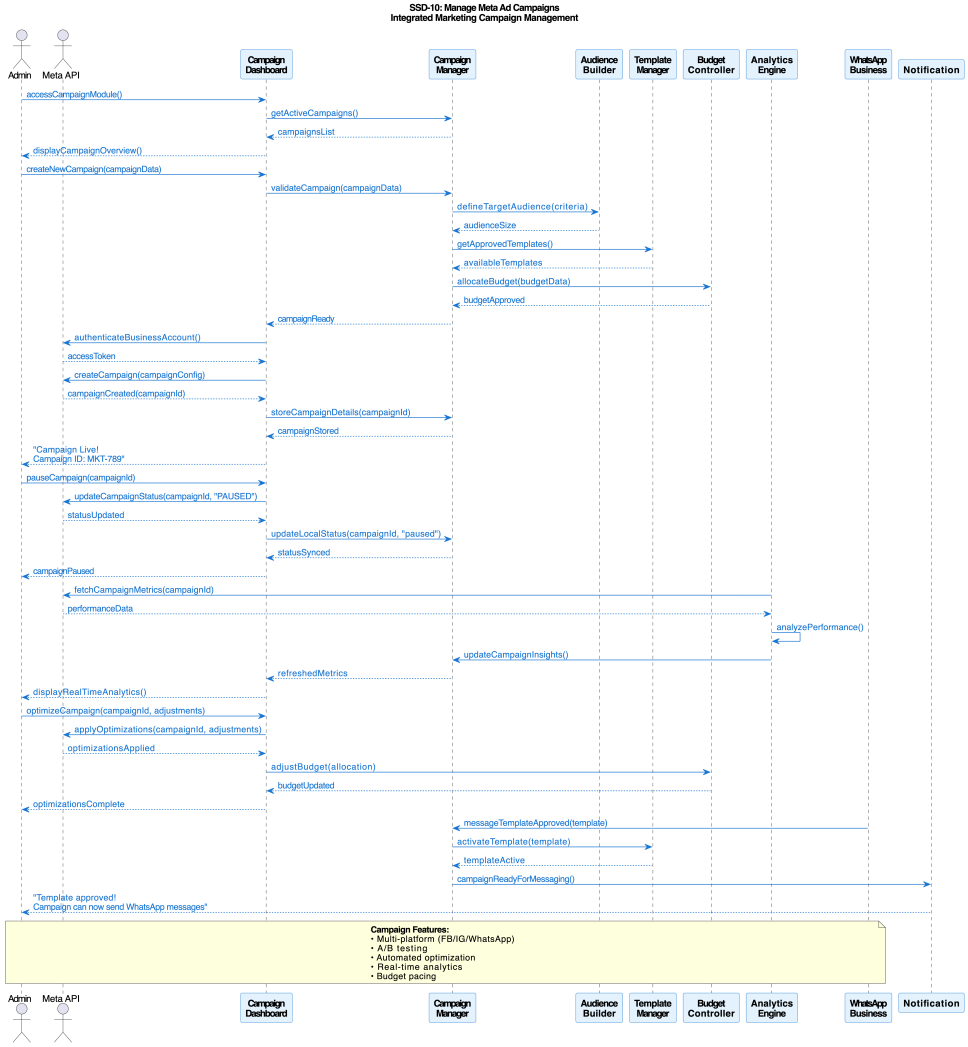

Figure 3.24: System Sequence Diagram for Manage Meta Ad Campaigns

Actors: Admin, Meta API, Campaign Dashboard, Campaign Manager, Audience Builder, Template Manager, Budget Controller, Analytics Engine, WhatsApp Business, Notification

3.4.11 SSD-11: Generate Reports & Analytics

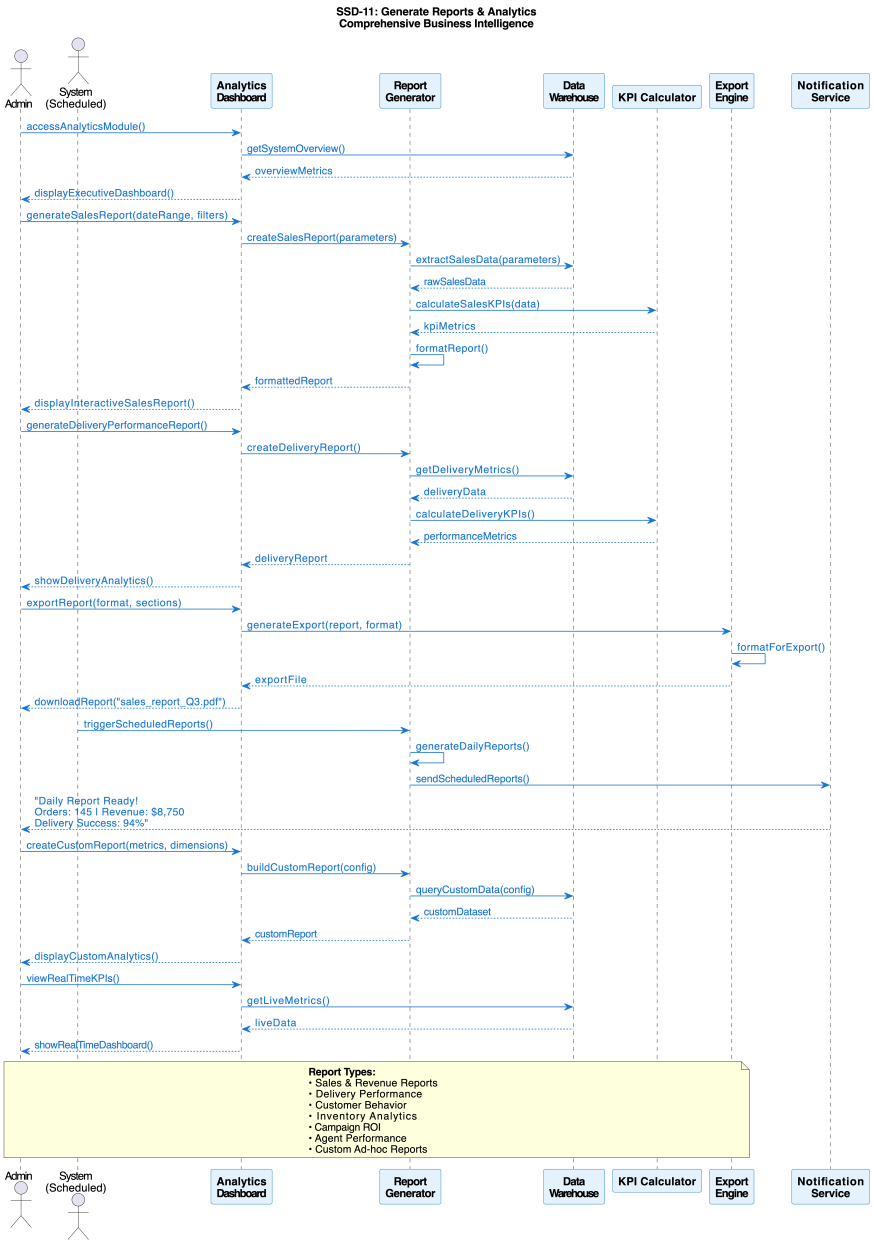

Figure 3.25: System Sequence Diagram for Generate Reports and Analytics

Actors: Admin, System Scheduler, Analytics Dashboard, Report Generator, Data Warehouse, KPI Calculator, Export Engine, Notification Service

3.4.12 SSD-12: User Management & Role Assignment

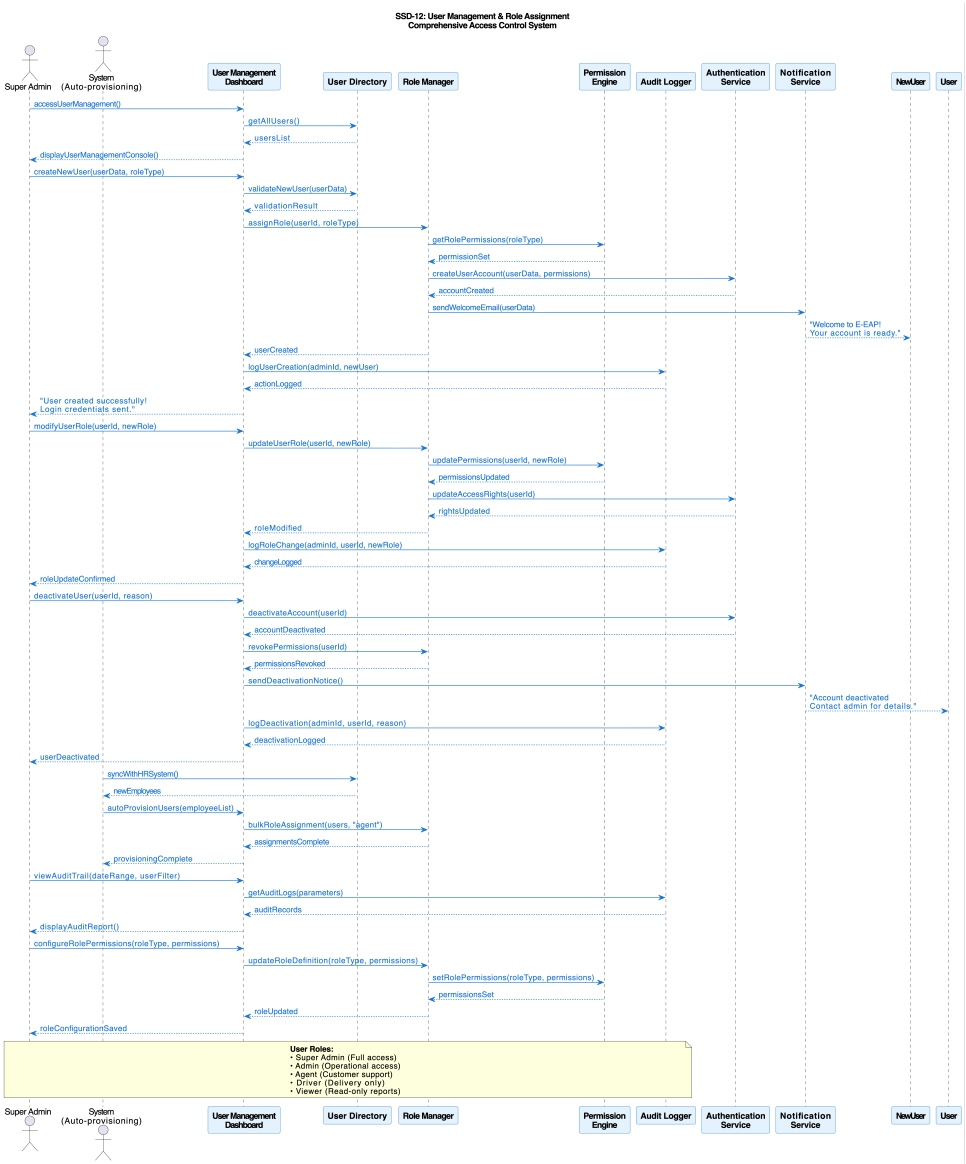

Figure 3.26: System Sequence Diagram for User Management and Role Assignment

Actors: Super Admin, System Auto-provisioning, User Management Dashboard, User Directory, Role Manager, Permission Engine, Audit Logger, Authentication Service, Notification Service, New User, User

3.5 Domain Sequence Diagrams (SD)

Domain sequence diagrams provide detailed technical views of specific business domain processes, showing interactions between domain objects and services.

3.5.1 Customer Domain SDs

3.5.1.1 SD-1.1: Order Creation Process

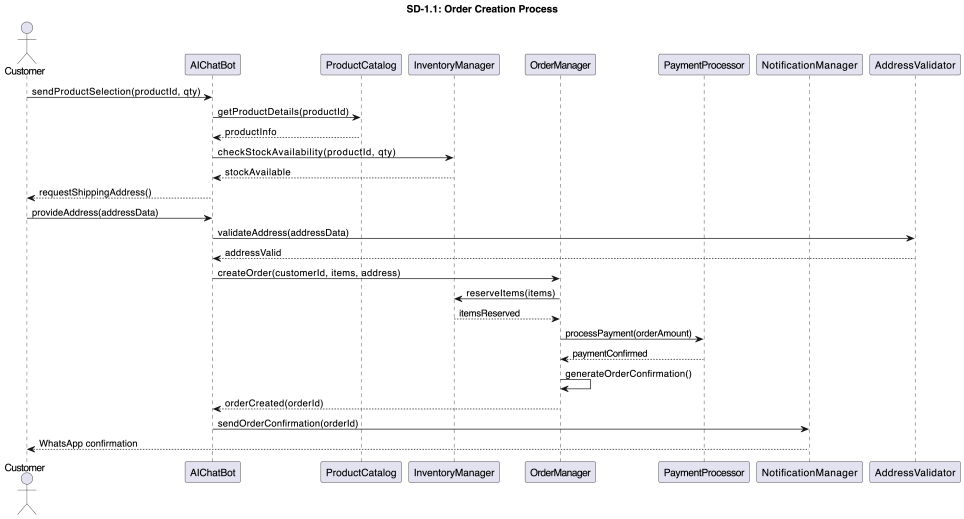

Figure 3.27: Domain Sequence Diagram for Order Creation Process

Components: Customer, AIChatBot, ProductCatalog, InventoryManager, OrderManager, PaymentProcessor, NotificationManager, AddressValidator

3.5.1.2 SD-1.2: Cart Management Process

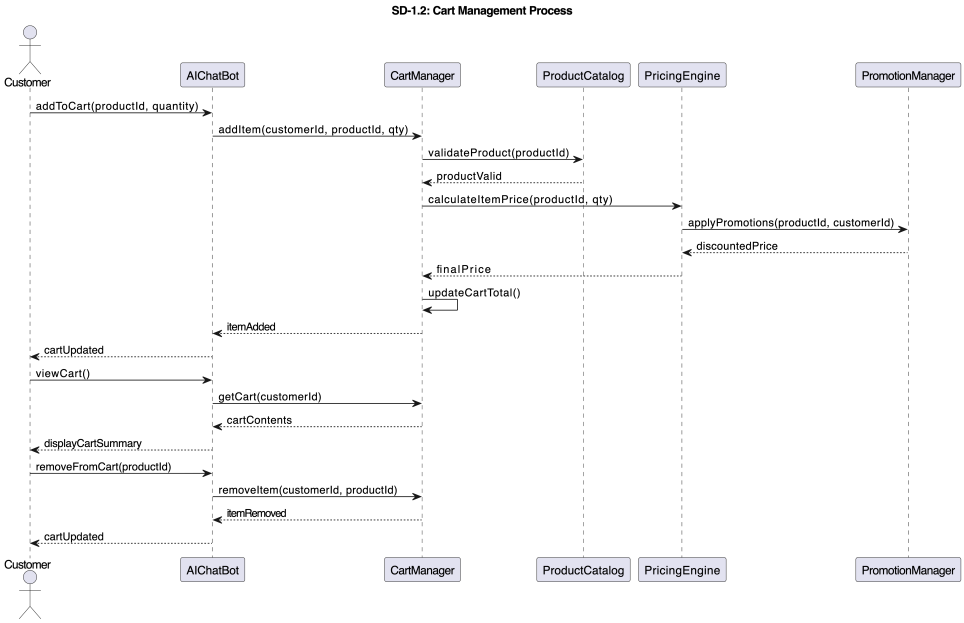

Figure 3.28: Domain Sequence Diagram for Cart Management Process

Components: Customer, AIChatBot, CartManager, ProductCatalog, PricingEngine, PromotionManager

3.5.1.3 SD-2.1: Real-time Order Tracking

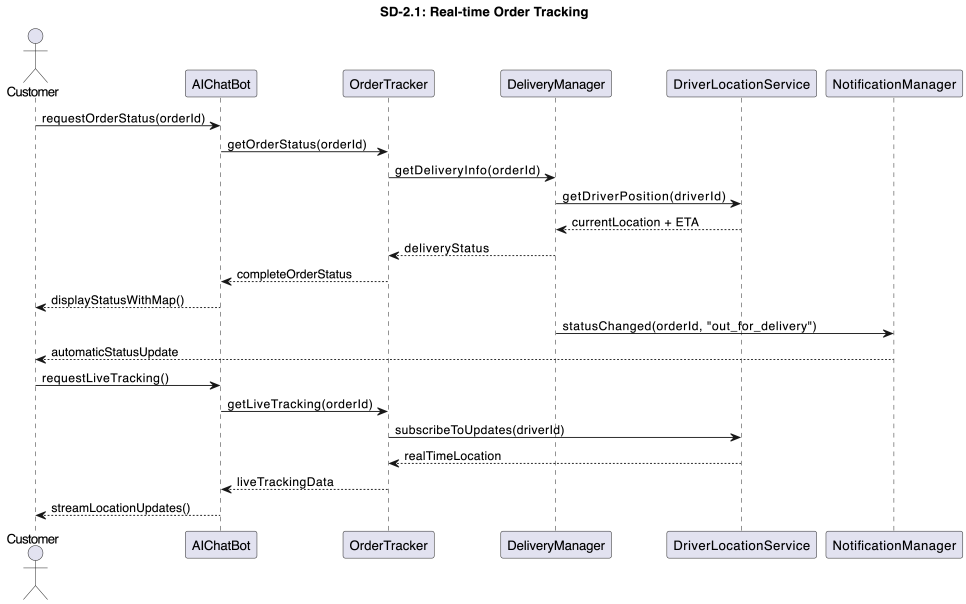

Figure 3.29: Domain Sequence Diagram for Real-time Order Tracking

Components: Customer, AIChatBot, OrderTracker, DeliveryManager, DriverLocationService, NotificationManager

3.5.1.4 SD-3.1: AI Chat Escalation Process

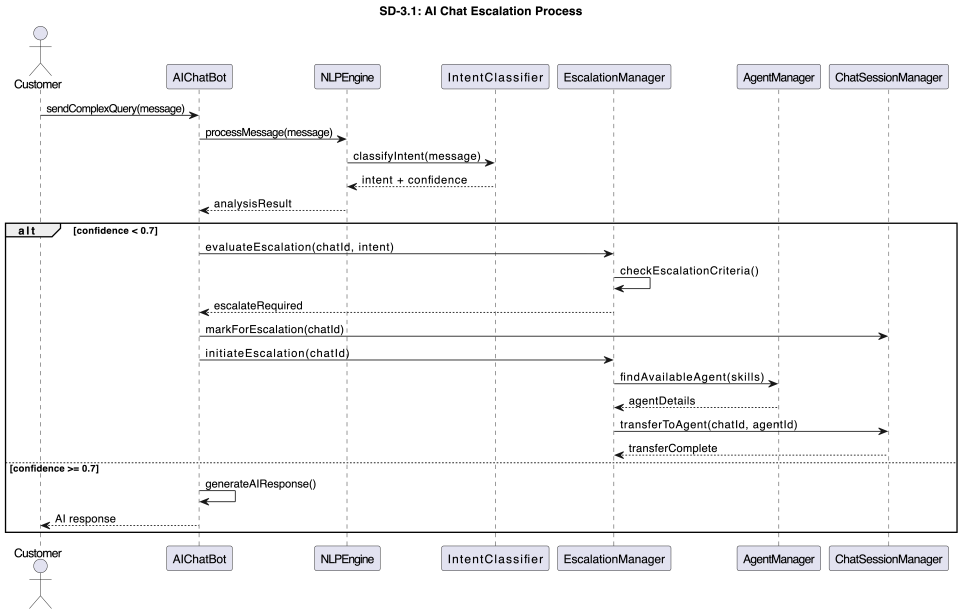

Figure 3.30: Domain Sequence Diagram for AI Chat Escalation Process

Components: Customer, AIChatBot, NLPEngine, IntentClassifier, EscalationManager, AgentManager, ChatSessionManager

3.5.2 Delivery Domain SDs

3.5.2.1 SD-5.1: Route Optimization Process

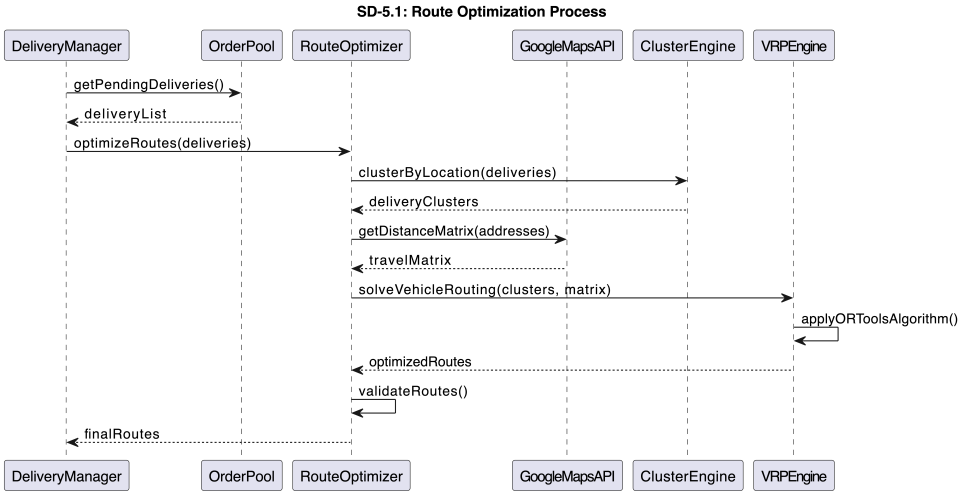

Figure 3.31: Domain Sequence Diagram for Route Optimization Process

Components: DeliveryManager, OrderPool, RouteOptimizer, GoogleMapsAPI, ClusterEngine, VRPEngine

3.5.2.2 SD-5.2: Driver Assignment Process

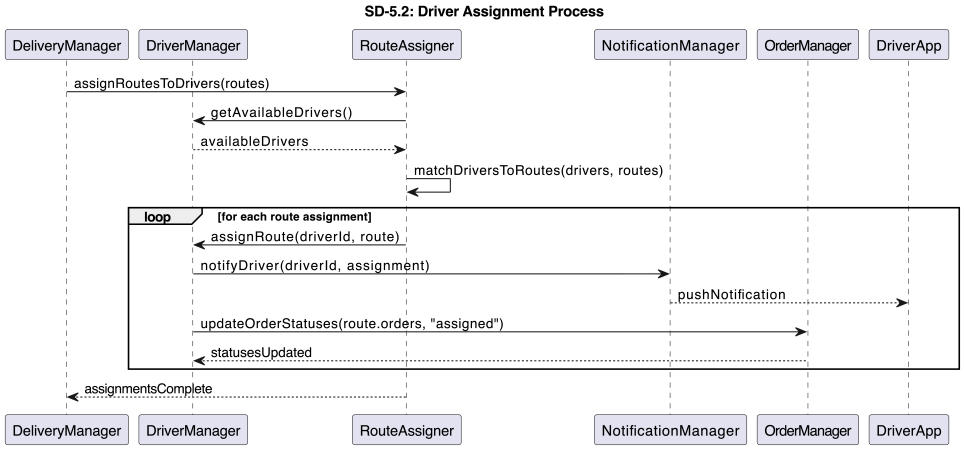

Figure 3.32: Domain Sequence Diagram for Driver Assignment Process

Components: DeliveryManager, DriverManager, RouteAssigner, NotificationManager, OrderManager, DriverApp

3.5.2.3 SD-6.1: COD Collection Process

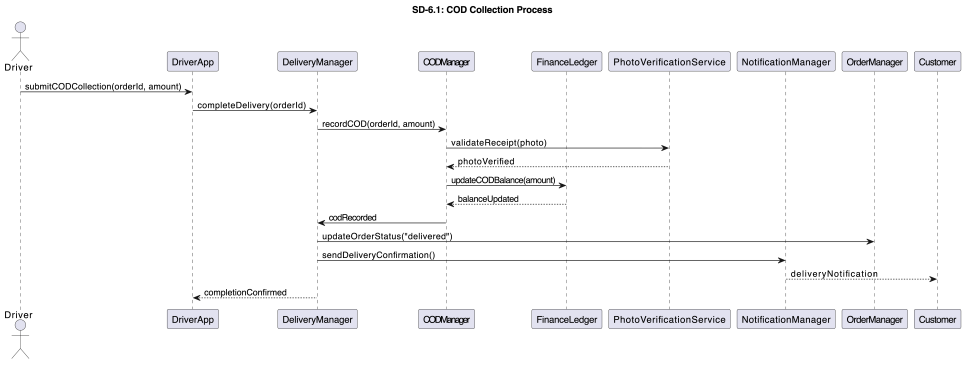

Figure 3.33: Domain Sequence Diagram for COD Collection Process

Components: Driver, DriverApp, DeliveryManager, CODManager, FinanceLedger, PhotoVerificationService, NotificationManager, OrderManager, Customer

3.5.2.4 SD-7.1: Failed Delivery Handling

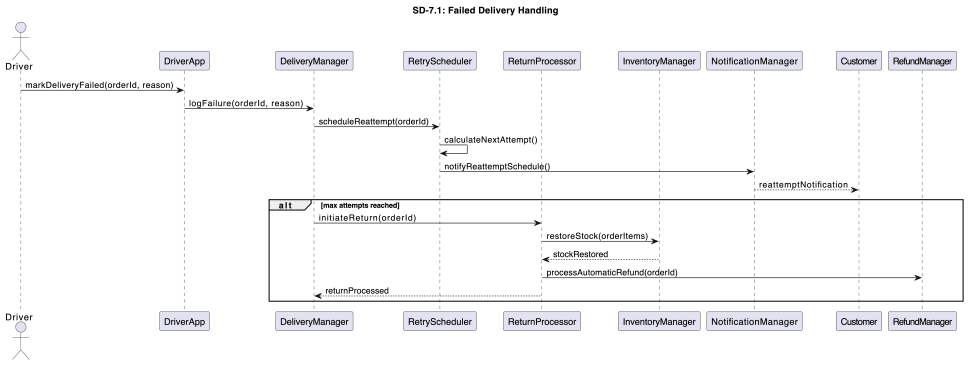

Figure 3.34: Domain Sequence Diagram for Failed Delivery Handling

Components: Driver, DriverApp, DeliveryManager, RetryScheduler, ReturnProcessor, InventoryManager, NotificationManager, Customer, RefundManager

3.5.3 Inventory Domain SDs

3.5.3.1 SD-8.1: Real-time Stock Management

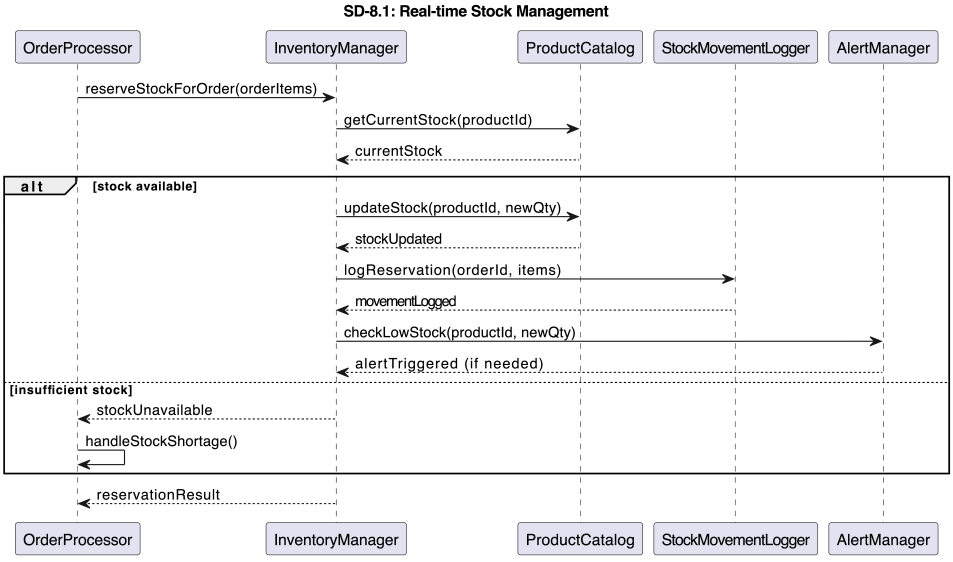

Figure 3.35: Domain Sequence Diagram for Real-time Stock Management

Components: OrderProcessor, InventoryManager, ProductCatalog, StockMovementLogger, AlertManager

3.5.3.2 SD-9.1: Automated Restocking Process

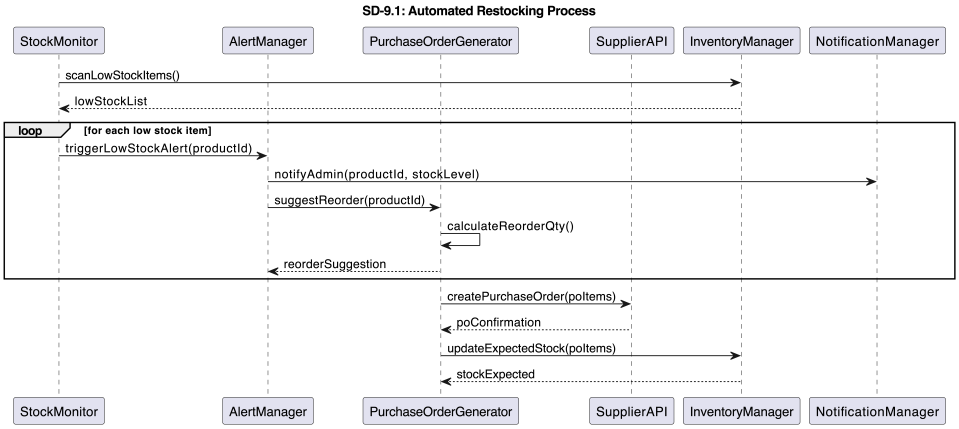

Figure 3.36: Domain Sequence Diagram for Automated Restocking Process

Components: StockMonitor, AlertManager, PurchaseOrderGenerator, SupplierAPI, InventoryManager, NotificationManager

3.5.4 Admin and Analytical Domain SDs

3.5.4.1 SD-10.1: Campaign Management Process

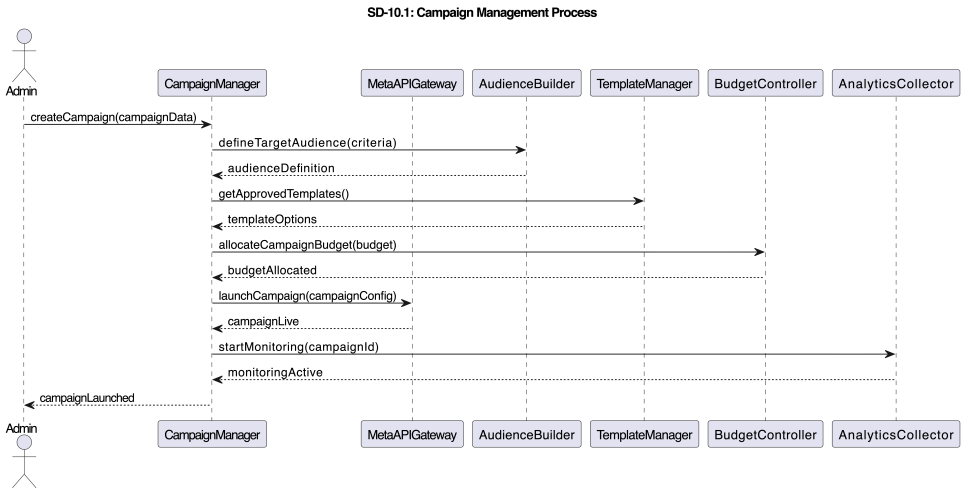

Figure 3.37: Domain Sequence Diagram for Campaign Management Process

Components: Admin, CampaignManager, MetaAPIGateway, AudienceBuilder, TemplateManager, BudgetController, AnalyticsCollector

3.5.4.2 SD-11.1: Report Generation Process

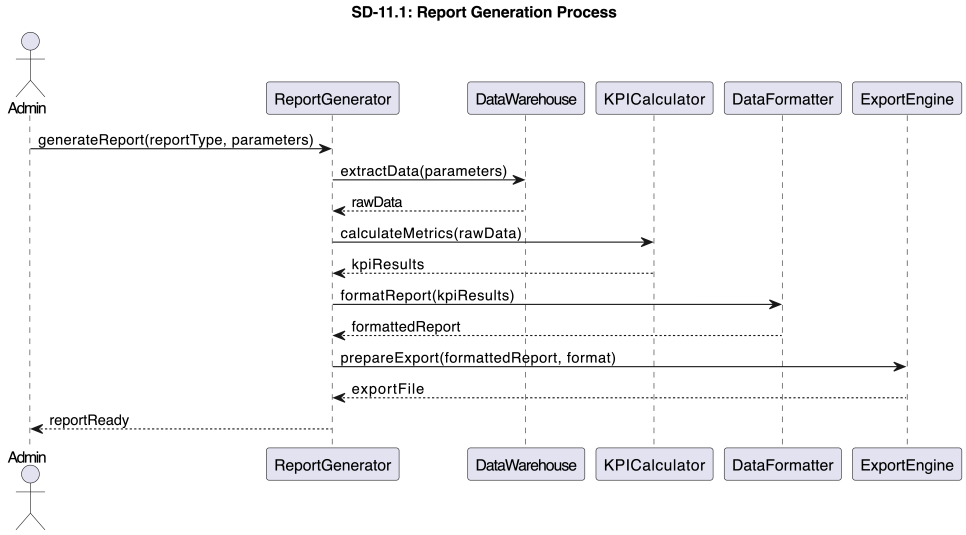

Figure 3.38: Domain Sequence Diagram for Report Generation Process

Components: Admin, ReportGenerator, DataWarehouse, KPICalculator, DataFormatter, ExportEngine

3.5.4.3 SD-12.1: User Role Management Process

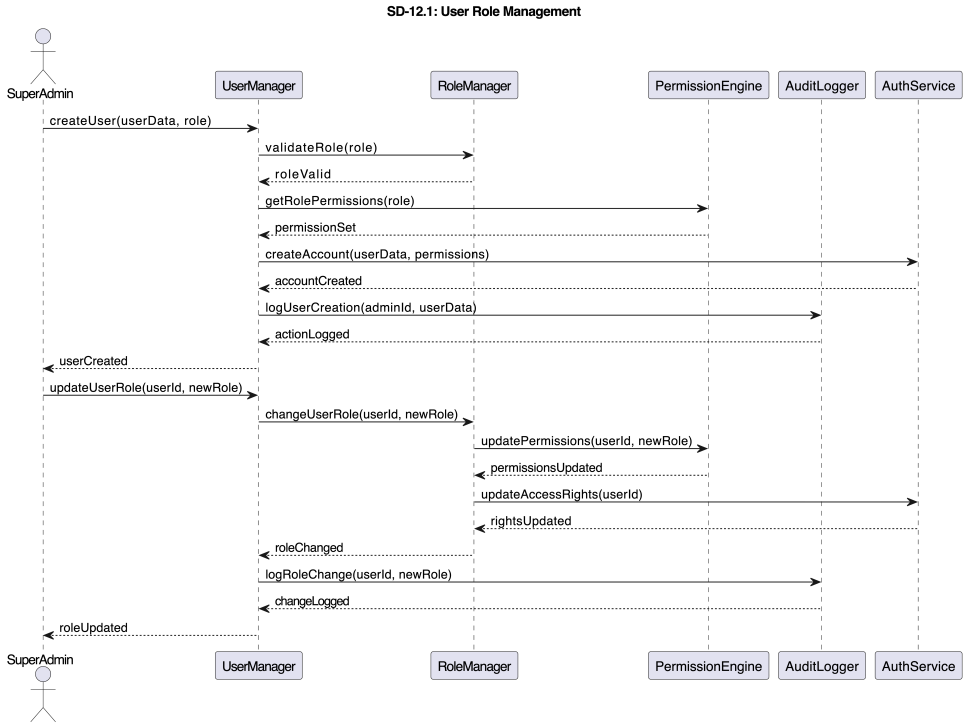

Figure 3.39: Domain Sequence Diagram for User Role Management Process

Components: SuperAdmin, UserManager, RoleManager, PermissionEngine, AuditLogger, AuthService

3.5.5 Cross Cutting Concern SDs

3.5.5.1 SD-A1: Notification Management

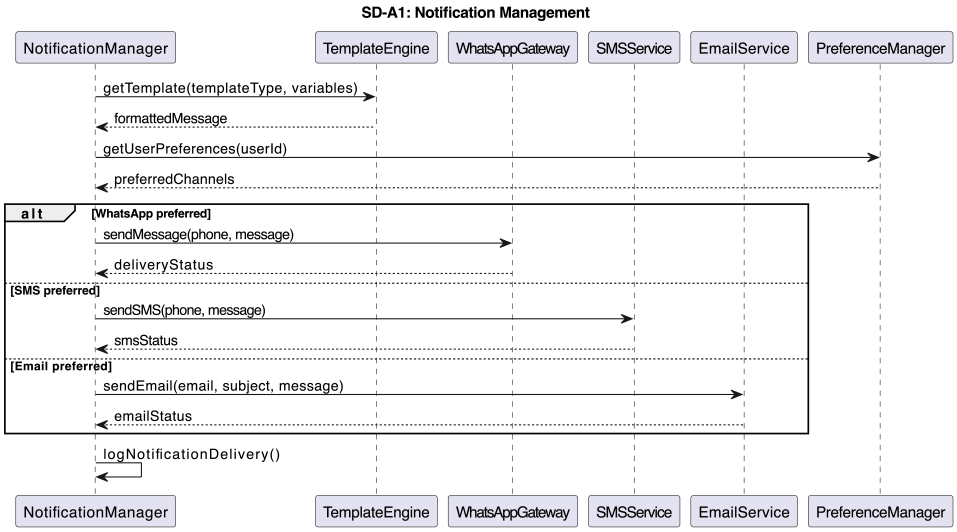

Figure 3.40: Domain Sequence Diagram for Notification Management

Components: NotificationManager, TemplateEngine, WhatsAppGateway, SMSService, EmailService, PreferenceManager

3.5.5.2 SD-A2: Error Handling & Recovery

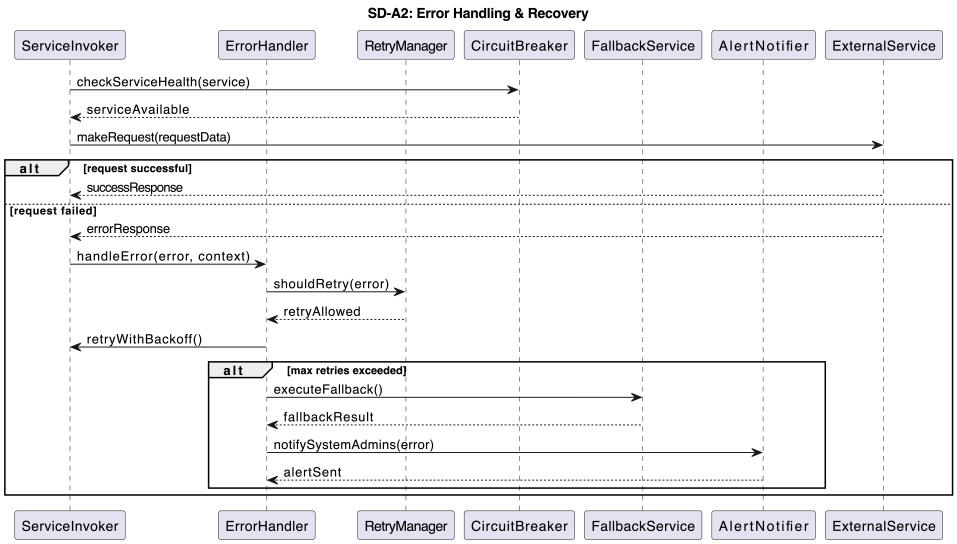

Figure 3.41: Domain Sequence Diagram for Error Handling and Recovery

Components: ServiceInvoker, ErrorHandler, RetryManager, CircuitBreaker, FallbackService, AlertNotifier, ExternalService

3.6 Summary

This chapter presented the comprehensive system design through various UML diagrams that model the E-EAP system's behavior, states, and interactions. The activity diagrams illustrate business workflows, state transition diagrams show entity lifecycles, system sequence diagrams depict actor-system interactions, and domain sequence diagrams provide technical details of domain-specific processes. Together, these diagrams provide a complete view of the system's architecture and design, ensuring all functional requirements are properly addressed and providing a solid foundation for implementation.

The design follows software engineering best practices including separation of concerns, modular architecture, and clear boundary definitions between different system domains. The state management patterns ensure robust handling of business processes while the sequence diagrams provide clear implementation guidance for developers.....

Bibliography