Group tasks: B+ please refer to your indv. task for your indv. mark.

Birzeit University

Department of Computer Science

COMP433: SOFTWARE ENGINEERING

Software Requirement Specifications and Design SRSD

Mobile Phones Shop

Group 3

1.	Ahmad Hamdan	1210241	Section 3
2.	Mohammad Fareed	1212387	Section 3
3.	Omar Hussain	1212739	Section 3
4.	Ismail Tarteer	1211243	Section 3
5.	Sohaib Badaha	1210472	Section 3

Date January 9, 2026

Group Members

Mohammad Fareed

Ahmad Hamdan

Omar Hussain

Ismail Tarteer

Sohaib Badaha

Contents

1. Project Planning and Management

1.1 Writers of the Report

Mohammad Fareed and Ahmad Hamdan

1.2 Business Title

Mobile Phones Shop

1.3 Group Name

G3

1.4 Name of Students

- Mohammad Fareed 1212387
- Ahmad Hamdan 1210241
- Ismail Tarteer 1211243
- Sohaib Badaha 1210472
- Omar Hussain 1212739

1.5 Role of Students

- Project Manager: Mohammad Fareed
- Secretary: Ahmad Hamdan
- Programmer: Sohaib Badaha
- Technical Architect: Ismail Tarteer
- Requirement Engineer: Omar Hussain

1.6 Project Management Strategy

The project team adopted a structured yet flexible project management strategy that emphasized effective communication and collaboration. Several tools were used to support coordination among team members throughout the project lifecycle. Facebook Messenger was used as the primary group communication channel for daily updates and quick discussions, while Google Meets was utilized for online meetings. In addition, the team frequently conducted in-person gatherings on the university campus, which proved especially useful for detailed discussions, requirement clarifications, and resolving complex design issues.

The team held regular weekly meetings to monitor progress, distribute tasks, and address any challenges encountered during development. During critical periods, such as phase submissions or tight deadlines, the number of meetings increased as needed to ensure timely completion of deliverables. Decision-making was handled collaboratively, where all team members were encouraged to share their opinions and suggestions. For decisions requiring consensus, team input was carefully considered, while the project manager made the final decision when necessary to maintain consistency and keep the project on track.

Regarding the development process, the team followed a hybrid approach combining elements of both the Waterfall and Agile software process models. Early phases of the project, such as requirement analysis and overall planning, followed a more structured Waterfall approach to establish a solid foundation. Later phases adopted Agile practices, allowing for iterative refinement, flexibility in task execution, and adaptation to feedback. For system modeling and diagram creation, the team used the Astah software tool, which provided strong UML support and facilitated clear and consistent documentation.

1.7 Project Manager Report

As the project manager, I, Mohammad Fareed, am pleased to reflect on the effort, collaboration, and overall outcome of this project. Leading the team throughout the different phases was both challenging and rewarding, as each member played a vital role in achieving the project objectives and delivering a well-structured and complete system.

Ahmad Hamdan, serving as the Secretary, was responsible for organizing documentation, ensuring consistency across reports, and maintaining clarity in written deliverables. His attention to detail and commitment to formatting, structure, and correctness significantly improved the overall quality of the submitted work. Sohaib Badaha, acting as the Programmer, contributed strongly to the implementation-oriented aspects of the project and supported the translation of requirements and designs into logical system behavior. Ismail Tarteer, as the Technical Architect, played a key role in shaping the system architecture and design decisions, providing valuable technical insights that enhanced the robustness and coherence of the overall solution. Omar Hussain, the Requirement Engineer, was instrumental in analyzing and documenting system requirements, ensuring that user needs and system functionalities were clearly defined and consistently reflected across use cases and diagrams.

Managing the project was not without challenges. One of the primary difficulties was coordinating schedules among team members, as differences in availability sometimes delayed discussions or task completion. Ensuring consistent communication and alignment across all phases also required continuous follow-up and review. Additionally, balancing workload distribution while meeting strict academic deadlines demanded careful planning and flexibility. In several instances, I had to closely monitor progress, resolve misunderstandings, and step in to guide tasks to maintain momentum and ensure deadlines were met.

Despite these challenges, I strongly believe the project was successful. The team demonstrated strong collaboration, adaptability, and commitment throughout the course. The final outcome reflects a clear understanding of software engineering principles, realistic system modeling, and effective teamwork. Overall, this project stands as a valuable learning experience that strengthened both our technical and managerial skills, and I am proud of the quality and completeness of the work we achieved together.

1.8 Group Members Report

1.8.1 Ahmad Hamdan

I believe our project was successful due to strong collaboration and consistent effort from all team members. As the secretary of the group, I was responsible for organizing and maintaining project documentation, ensuring consistency in report structure, formatting, and clarity across all phases. I actively contributed to writing and refining multiple report sections, reviewing diagrams, and participating in discussions related to system analysis and design. I also contributed to sequence diagram development and provided feedback on class and component diagrams. This project enhanced my technical writing, documentation, and review skills, and helped me better understand how clear documentation supports effective software development.

1.8.2 Ismail Tarteer

From my perspective, the project was successful in delivering a coherent and realistic system design that reflects real-world requirements. As the technical architect, I focused on shaping the overall system architecture and ensuring that design decisions were technically sound and consistent. I contributed by leading architectural discussions, defining system components, reviewing class and component diagrams, and ensuring proper separation of responsibilities within the system. I also participated in sequence diagram analysis and design goal discussions. This project strengthened my system design and architectural thinking skills, and allowed me to apply theoretical concepts.

1.8.3 Sohaib Badaha

I consider this project a successful learning experience that resulted in a clear and well-modeled system. As the programmer in the team, I contributed by focusing on system behavior, logic flow, and translating requirements into structured designs. I participated in developing and reviewing sequence diagrams, class diagrams, and activity flows, ensuring that system operations were logically consistent and complete. I also took part in discussions related to implementation feasibility and design refinement. This project helped me improve my understanding of system logic, diagram interpretation, and teamwork.

1.8.4 Omar Hussain

I believe the project achieved its objectives by clearly capturing user needs and translating them into well-defined system requirements and models. As the requirement engineer, I was responsible for analyzing and documenting user and system requirements, ensuring clarity and completeness. I led and contributed to use-case analysis, participated in actor identification, and reviewed use-case specifications and diagrams to ensure consistency across the project. I also contributed to discussions related to system behavior and validation of requirements against design artifacts. This project enhanced my requirement analysis skills and reinforced the importance of accurate requirement documentation.

2. Requirement Elicitation, Analysis and Modeling

2.1 Business Description

The business is an online mobile phone store that specializes in selling high-quality mobile phones and accessories from multiple manufacturers. Customers can browse products through a web-based platform, view detailed product information, and place orders online. The system supports home delivery services within the West Bank and provides a complete online shopping experience from product selection to order fulfillment. The store operates continuously to display products and receive orders, requiring only an internet connection and access through modern web browsers on desktop or mobile devices.

The business offers a variety of services including selling mobile phones and accessories, providing product recommendations based on customer needs such as daily use, photography, or gaming, and supporting multiple payment methods such as cash on delivery, bank card payments, and online payment services. The delivery process is handled through delivery companies, with an expected delivery time of one to two days. The business generates revenue through direct product sales, accessory sales, and seasonal promotions aimed at increasing customer engagement and sales volume.

Operationally, the business manages product inventory, pricing, order processing, payment handling, and customer support services. Store administrators are responsible for managing products, tracking orders, and organizing inventory, while customer support staff assist users with payment or delivery-related issues. The business collaborates with suppliers and shipping companies to ensure product availability and timely delivery. With estimated monthly orders ranging between 50 and 150 and a product catalog of 30 to 40 phone models, the system is designed to efficiently support the store's operational and business needs.

2.2 User and System Requirements MODIFIED

2.2.1 User Requirements

Lead: Mohammad Fareed
Contributors: Ahmad Hamdan (discussion), Omar Husien (review), Ismail Tarteer (writing), Sohaib Badaha (finalization)

UR-01 The system shall allow customers to search for products using keywords.

UR-02 The system shall allow customers to view complete product details including specifications, price, images, and availability status.

UR-03 The system shall allow customers to add products to a shopping cart.

UR-04 The system shall allow customers to select a payment method during checkout, including cash on delivery and bank card payment.

UR-05 The system shall allow customers to track the current order status, including placed, processed, shipped, delivered, canceled, or refunded.

UR-06 The system shall allow customers to enter delivery address information for supported delivery areas within the West Bank.

UR-07 The system shall allow the store manager (admin) to manage inventory quantities.

UR-08 The system shall allow customers to submit customer support requests related to payment or delivery issues.

UR-9 The system shall allow customer support staff to respond to support requests and track each case until it is resolved.

2.2.2 System Requirements MODIFIED

Lead: Ahmad Hamdan & Sohaib Badaha
Contributors: Mohammad Fareed (review), Omar Husien (discussion), Ismail Tarteer (finalization)

SR-01 (The system shall allow customers to search for products using keywords.)

- SR-01.1 The system shall store product information, including name, category, brand, price, and availability status.
- SR-01.2 The system shall allow users to search for products by product name, brand, or model.
- SR-01.3 The system shall allow users to filter product results by category, brand, price range, and availability.
- SR-01.4 The system shall mark products with zero inventory quantity as out of stock.

SR-02 (UR-02 The system shall allow customers to view complete product details including specifications, price, images, and availability status.)

- SR-02.1 The system shall provide a product details page containing specifications, price, images, and available quantity.
- SR-02.2 The system shall provide related accessories associated with the selected product.
- SR-02.3 The system shall recommend products based on categories of the product.

SR-03 (The system shall allow customers to add products to a shopping cart.)

- SR-03.1 The system shall allow customers to alter product quantities or remove products from the shopping cart.
- SR-03.2 The system shall calculate and display the cart subtotal, delivery cost depend on area, and final total amount.
- SR-03.3 The system shall validate product availability before allowing order confirmation.
- SR-03.4 The system shall notify the customer about the confirmed order.

SR-04 (The system shall allow customers to select a payment method during checkout, including cash on delivery and bank card payment.)

- SR-04.1 The system shall validate the entered coupon code and apply the corresponding discount if the code is valid depending on the expiry date.
- SR-04.2 The system shall integrate with an external payment gateway to support card payments.
- SR-04.3 The system shall generate and store an electronic invoice for each confirmed order.
- SR-04.4 The system shall forget sensitive payment data such as full card numbers only the first four digits.

SR-05 (The system shall allow customers to track the current order status, including placed, processed, shipped, delivered, canceled, or refunded.)

- SR-05.1 The system shall generate a unique order identifier for each confirmed order.
- SR-05.2 The system shall assign one of the following statuses to each order: placed, processing, shipped, delivered, canceled, or refunded.
- SR-05.3 The system shall allow admin to update the order status.
- SR-05.5 The system shall record timestamps for each order status change.

SR-06 (The system shall allow customers to enter delivery address information for supported delivery areas within the West Bank.)

- SR-06.1 The system shall allow customers to enter delivery address details including city, street, and phone number.
- SR-06.2 The system shall validate that the delivery address is within supported areas in the West Bank.
- SR-06.3 The system shall calculate an estimated delivery time depending on the city.
- SR-06.4 The system shall store delivery details as part of the order record.

SR-07 (The system shall allow the store manager (admin) to manage inventory quantities.)

- SR-07.1 The system shall provide role-based authorization for store administrators.
- SR-07.2 The system shall allow administrators to add, edit, or remove product records.
- SR-07.3 The system shall allow administrators to update product prices and inventory quantities.
- SR-07.4 The system shall log inventory changes with date, time, and administrator identifier.

SR-08 (The system shall allow customers to submit customer support requests related to payment or delivery issues.)

- SR-08.1 The system shall provide a support request submission interface for customers.
- SR-08.2 The system shall generate a unique support ticket identifier for each request.
- SR-08.2 The system shall allow user to update his support ticket or respond to it.
- SR-08.3 The system shall allow support staff to respond to customer tickets.
- SR-08.4 The system shall allow support staff to update ticket status as open, in progress, or resolved.
- SR-08.5 The system should store all support ticket interactions for future reference.

SR-09 (The system shall allow customer support staff to respond to support requests and track each case until it is resolved.)

- SR-09.1 The system shall allow support staff to respond to customer tickets.
- SR-09.2 The system shall allow support staff to update ticket status as open, in progress, or resolved.
- SR-09.3 The system should store all support ticket interactions for future reference.

2.3 Scenario Analysis

2.3.1 Scenario 1 – Submit return request MODIFIED

Writer: Mohammad Fareed

Initial Assumptions: Customer Ahmad has recently received an order purchased through the system. The delivery has been completed successfully, and the order status is marked as Delivered. The return management service is operational.

Normal Flow: Customer Ahmad receives his mobile phone order and determines that it should be returned due to an issue such as a defect or receiving an incorrect model. Ahmad navigates to his order history, and selects the delivered order containing the mobile phone. The system presents a return request form where Ahmad selects a predefined return reason and may provide additional details. Upon submission, the system registers the return request and forwards it to the return staff for review. The return staff reviews the request through the system and decides whether to accept or reject the return based on delivery date and return policy constraints. If accepted, the system displays the expected refund timeline based on the original payment method, generates return instructions, and updates the item status to Return in Progress.

Alternative Flow 1 Customer Ahmad receives an order containing multiple products. After inspecting the delivered products, the customer decides to return only a specific product. He decides to return only the mobile phone and submits a return request for the selected item through the system. The system forwards the request to the return staff, who reviews it, and the system calculates the refundable amount for the phone, while leaving the remaining items unaffected.

→ **Successful Output?** Yes

Alternative Flow 2 Customer Ahmad receives his delivered order and, after inspection, discovers an extra mobile phone that he did not order. He decides to submit a return request for the extra phone and selects the option for misdelivered item, providing a description explaining that the phone was not part of his order. The system forwards the request to the return staff, who accept it, and the system generates return instructions, didn't calculate refundable amount and updates the case status to Return in Progress.

→ **Successful Output?** Yes

Error Flow Customer Ahmad receives his delivered order and attempts to submit a return request for a phone after the allowed return period has expired . The system reject the request directly and notifies Ahmad of the rejection and displays the reason with a reference to the store's return policy.

→ **Successful Output?** No

System State on Completion: For successful returns, the system stores the return request with associated metadata, including reason, and selected return type. The item status is updated to Return in Progress. For unsuccessful attempts, no return record is created, and the event is logged without affecting the order.

2.3.2 Scenario 2 – Customer Support Ticket Handling

Writer: Ahmad Hamdan

Initial Assumptions: Customer Ahmad is using the system and encounters an issue while interacting with it. If the issue is related to an order, the referenced order exists in the system database. The customer support module is operational, and support staff are available.

Normal Flow: Customer Ahmad encounters a delivery-related issue and logs into the system. He navigates to the Customer Support section and selects the option to submit a new support request. The system presents a support form where Ahmad selects the issue category and provides a description of the problem. After submission, the system creates a support ticket, assigns it a unique identifier, and sets its status to Open. A support staff member reviews the ticket, provides guidance or resolution steps through the system, and updates the ticket status to In Progress. Once the issue is resolved, the support staff updates the ticket status to Resolved, and the system notifies Ahmad of the resolution.

Alternative Flow 1: Customer Ahmad submits a support ticket describing an issue but provides limited details. The system creates the ticket and assigns it an Open status. A support staff member reviews the ticket and requests additional information through the system. Ahmad replies with the required details, which are appended to the ticket. The ticket remains in the In Progress state until the issue is resolved and then is marked as Resolved.

→ **Successful Output?** Yes

Alternative Flow 2: Customer Ahmad submits a support request related to a known issue with predefined resolution steps. The system automatically presents a suggested solution to Ahmad. Ahmad accepts the solution, and the system closes the ticket and marks it as Resolved without further staff intervention.

→ **Successful Output?** Yes

Error Flow: Customer Ahmad attempts to submit a support request without selecting an issue category or providing a description. The system blocks the submission and displays a message indicating the missing required information. No support ticket is created until Ahmad corrects the input.

→ **Successful Output?** No

System State on Completion: For successful cases, the system stores the support ticket with its full interaction history, timestamps, and final status. Resolved tickets are archived but remain accessible to both customers and support staff for future reference and auditing. Notifications are logged and delivered to the appropriate users upon status changes. In error cases, no ticket is created, and the failed submission attempt is logged without affecting system data integrity.

2.3.3 Scenario 3 – Process Payment

Writer: Ismail Tarteer

Initial Assumptions: Customer Sara is logged into the system and has at least one valid order ready for checkout. The order exists in the system database and has not yet been paid. The checkout module is operational. For bank card payments, the external payment gateway is available unless otherwise specified. The invoicing subsystem is active and capable of generating electronic invoices.

Normal Flow: Customer Sara is ready to pay for her order during checkout. The system displays the available payment methods, including cash on delivery and bank card payment. Sara selects bank card payment and enters the required card-payment details in the checkout payment step. After confirming the payment, the system sends the payment request to the external payment gateway. The payment gateway approves the transaction and returns a success response. The system confirms the payment and generates and stores an electronic invoice for the order.

Alternative Flow: Customer Sara chooses cash on delivery as the payment method instead of bank card payment. The system records the selected payment method as cash on delivery and confirms the checkout payment step. The system generates and stores an electronic invoice for the confirmed order.

→ **Successful Output?** Yes

Error Flow 1: Customer Sara selects bank card payment and enters the required payment details. The system sends the payment request to the external payment gateway. The payment gateway declines the transaction due to reasons such as insufficient funds or invalid authorization. The system displays a payment failure message and does not confirm the payment. Sara is given the option to retry the payment or select cash on delivery instead.

→ **Successful Output?** No

Error Flow 2: Customer Sara selects bank card payment and submits the payment details. The system attempts to communicate with the external payment gateway, but the gateway is unavailable due to a network or timeout issue. The system displays an error message asking Sara to try again later. The payment process is not completed, and no invoice is generated.

→ **Successful Output?** No

System State on Completion: For successful cases, the system records the completed checkout with the selected payment method. The order status is updated to Paid for bank card payments or Pending Payment on Delivery for cash on delivery orders. The generated electronic invoice is stored with timestamps, and payment confirmation details are logged for auditing purposes. For error cases, the order remains unpaid, no invoice is generated, and the failed payment attempt is logged without affecting order data integrity.

2.3.4 Scenario 4 – Deliver Order

Writer: Sohaib Badaha

Initial Assumptions: The delivery personnel is authenticated and logged into the chocolate shop system with proper permissions. Orders with status "Shipped" are already assigned by the Delivery Coordinator. Customer addresses are valid within the West Bank. Orders are packaged and ready, the delivery personnel has access to a vehicle, and for cash on delivery orders, sufficient change is available.

Normal Flow: Sohaib logs into the system at 10:00 AM and views five assigned orders. He selects order ORD-2025-1223-045, confirms its details, and marks it as "Out for Delivery" at 10:15 AM. He arrives at the customer's address at 10:45 AM, verifies the customer's identity, delivers the package in good condition, and collects a digital signature. He updates the order status to "Delivered" at 10:50 AM. The system records the delivery timestamp and sends a confirmation notification to the customer.

→ **Successful Output?** Yes

Alternative Flow 1: Sohaib delivers a cash-on-delivery order worth 150 NIS. After the customer inspects the items, the customer pays in cash and receives change and a receipt. The customer signs digitally, and Sohaib updates the order to "Delivered – Cash Collected". The system records the payment and notifies the customer and accountant.

→ **Successful Output?** Yes

Alternative Flow 2: Sohaib arrives at the delivery address but the customer is unavailable. After contacting the customer, the delivery is rescheduled to a later time. The system updates the order to "Delivery Rescheduled" and sends a notification. The order is later delivered successfully and marked as "Delivered."

→ **Successful Output?** Yes

Error Flow: Sohaib cannot complete a delivery due to an incorrect address and an unreachable customer. He marks the order as "Delivery Failed" with detailed notes. The system creates a support ticket, sends notifications to relevant staff, and keeps the order pending until the issue is resolved and delivery is rescheduled.

→ **Successful Output?** No

System State on Completion: The system maintains an audit trail for deliveries including timestamps, customer signatures, and cash-on-delivery payment records, updates the final order status, and sends automatic customer notifications. In case of delivery failure, it logs all attempts and reasons, creates support tickets, notifies relevant staff, and keeps the order active until the issue is resolved

2.3.5 Scenario 5 – Place Order

Writer: Omar Hussain

Initial Assumptions: The customer is browsing the online mobile phones and accessories catalog and intends to complete a purchase by placing an order online. The system is operational, all required services such as inventory, payment, and delivery estimation are available, and the customer has items already added to the shopping cart.

Normal Flow: The customer opens the shopping cart and reviews the selected items, such as a mobile phone and a phone accessory. The customer clicks on the *Checkout* button to continue with the purchasing process. The system prompts the customer to enter the delivery address details. The customer enters a valid delivery address located within the supported delivery areas in the West Bank, and the system displays the estimated delivery time. The customer then enters a valid coupon code and selects a preferred payment method, either cash on delivery or bank card payment, and confirms the order. The system verifies product availability, successfully places the order, generates a unique order number, and displays a confirmation message to the customer.

Alternative Flow Before confirming the order, the customer decides to modify the quantity of one of the selected products. The system updates the shopping cart accordingly and allows the customer to continue the checkout process without any issues.

→ **Successful Output?** Yes

Error Flow 1 During the checkout process, one of the selected products becomes out of stock. The system notifies the customer that the item is no longer available and prompts the customer to either remove the product from the cart or select an alternative product.

→ **Successful Output?** No

Error Flow 2 The customer enters an invalid or expired coupon code during checkout. The system rejects the coupon code and displays an error message, allowing the customer to enter a valid code or proceed without applying a discount.

→ **Successful Output?** No

System State on Completion: For successfully placed orders, the system updates the inventory quantities, generates an electronic invoice, and stores the order in the database with a unique order ID. The order status is marked as *Placed* and becomes visible in the customer's order history, ready to be processed for delivery. For unsuccessful attempts, the order is not created, and no inventory changes occur.

2.4 Effort & Time Estimation Calculation

Lead: Omar Husien
Contributors: Mohammad Fareed (discussion), Ahmad Hamdan (review), Ismail Tarteer (finalization), Sohaib Badaha (writing)

This estimation uses a simplified person-week method (pw). 1 pw = one person working full-time for one week. A 30% buffer is added to the schedule to the weekend since the week is 5 days.

UR	**Estimated Effort**	**Estimated No. of Developers**	**Total Effort**
UR-01	2 pw	1	$= 2 \times 1 = 2$ pw
UR-02	2 pw	1	$= 2 \times 1 = 2$ pw
UR-03	3 pw	2	$= 3 \times 2 = 6$ pw
UR-04	2 pw	1	$= 2 \times 1 = 2$ pw
UR-05	2 pw	1	$= 2 \times 1 = 2$ pw
UR-06	2 pw	1	$= 2 \times 1 = 2$ pw
UR-07	4 pw	2	$= 4 \times 2 = 8$ pw
UR-08	2 pw	1	$= 2 \times 1 = 2$ pw
Total effort / avg	$(2+2+3+2+2+2+4+2) = 19$ **pw**	$(1+1+2+1+1+1+2+1)/8 = 1.25$ **dev on avg**	$(2+2+6+2+2+2+8+2) = 26$ **pw**
Schedule time (30%)	$19 \times 1.30 = 24.7 \approx 25$ **w (min time)**		$26 \times 1.30 = 33.8 \approx 34$ **w (max time)**
Cost		**Avg salary = 250 USD / week**	$250 \times 34 = 8500$ **USD**
Profit margin min = 10% max = 30%			**Min cost:** $8500 \times 1.10 = 9350$ USD **Max cost:** $8500 \times 1.30 = 11050$ USD

2.5 Actors Analysis MODIFIED

Lead: Ahmad Hamdan
Contributors: Sohaib Badaha (review), Mohammad Fareed (review), Omar Hussain (discussion), Ismail Tarteer (discussion)

- **Customer** Represents the primary role of the system. Can browse and search mobile phones and accessories, view product information, add items to the shopping cart, apply coupon codes, make payments, provide delivery address details, view order history, track order status, initiate return or exchange requests for delivered products within policy limits, and submit customer support requests.
- **Customer Support Staff** Represents authorized staff responsible for handling customer inquiries and issues through the support module. Can view submitted support requests, respond to customers, update ticket statuses (open, in progress, resolved), and track each case until resolution.
- **Dispatch Staff** Represents staff responsible for preparing confirmed orders for shipment. Can review confirmed orders, pack and label items, and confirm handoff to the delivery staff by updating the order status to shipped.
- **Delivery Staff** Represents staff responsible for delivering shipped orders to customers. Can deliver orders, update the order status to delivered, and if cash on delivery is selected, confirm payment collection by marking the payment as received.
- **Return Processing Staff** Represents authorized staff responsible for handling return and exchange requests through the system. Can review submitted return requests, accept or reject them based on return policies, update return statuses, and trigger follow-up actions such as refund processing or replacement shipment linked to an existing order return.
- **Payment Gateway** Represents an external service used to process online card payments securely. Responsible for validating payment information and returning a success or failure response to the system. The system does not store sensitive payment data such as full card numbers.
- **Email Service** Represents an external notification mechanism used to send transactional messages such as order confirmations, delivery updates, electronic invoices, and support ticket status notifications to customers.

2.6 Use-Case Diagram MODIFIED

Lead: Mohammad Fareed
Contributors: Ahmad Hamdan (modify drawing), Omar Husien (review), Ismail Tarteer (writing), Sohaib Badaha (finalization)

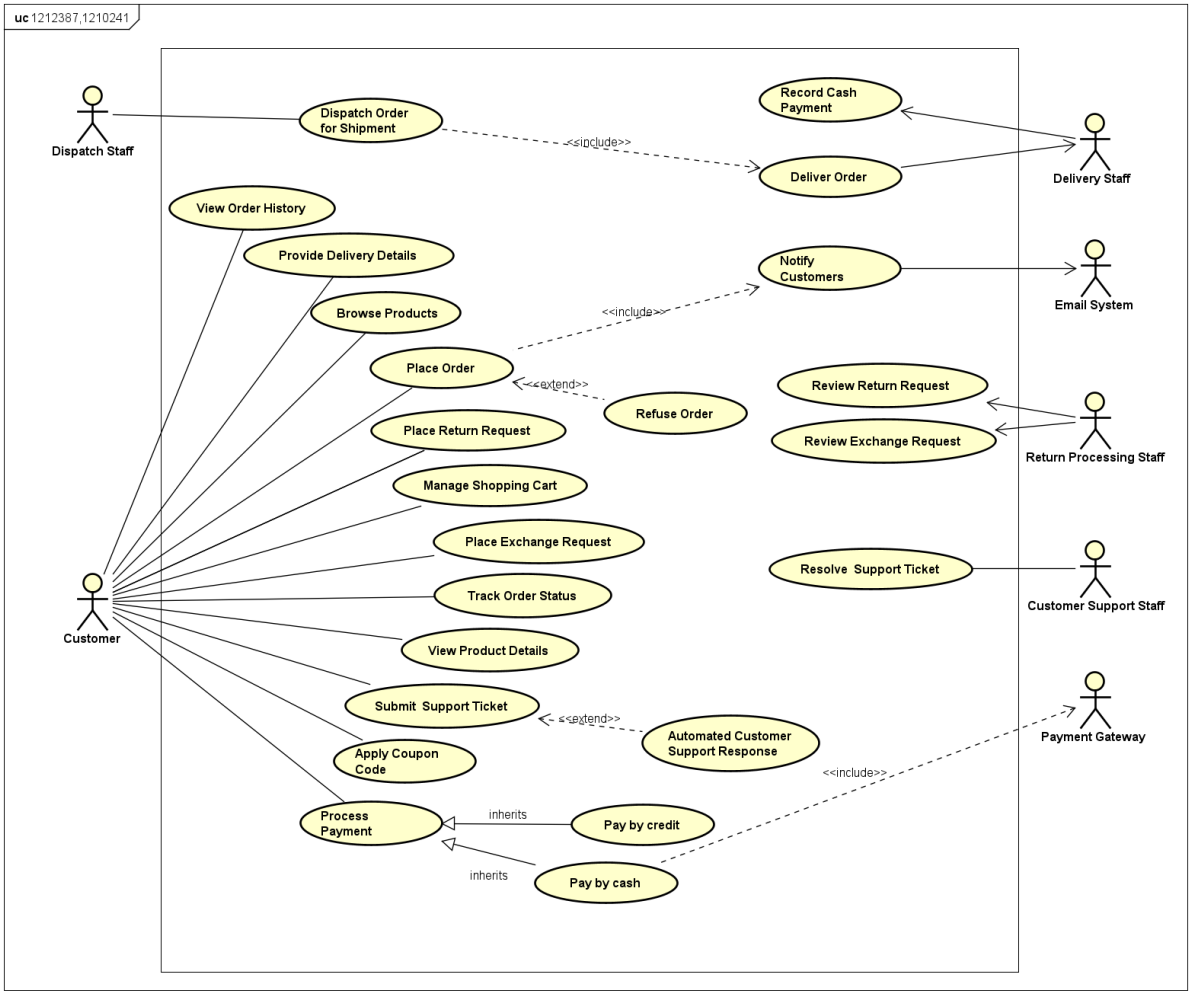


Figure 4: Use Case Diagram for the Mobile Phones Shop System

2.7 Use-Case Specifications MODIFIED

2.7.1 UC16 – Place Return Request

Writer: Mohammad Fareed

Use Case Title	Place Return Request
Description	This use case describes how a customer submits a return request for a delivered product within the store's return policy period. The system forwards the request to return staff for review, and updates the return status based on the staff's decision.
Actors	**Primary Actor:** Customer and Return Processing Staff
Data	• Order details (order ID, delivery status, delivery date) • Product details (product ID, return eligibility, non-returnable flag) • Return request details (reason for return, additional notes, return type) • Store return policy rules
Trigger	Customer initiates a return request for a delivered product through the system.
Pre-conditions	1. Customer has an existing order in the system. 2. The order status is marked as Delivered.
Work Flows	**Normal Flow** 1. Customer receives the delivered mobile phone. 2. Customer inspects the product and decides it should be returned due to a defect or incorrect model. 3. Customer navigates to the order history section. 4. Customer selects the delivered order containing the mobile phone. 5. Customer selects a predefined return reason and optionally enters additional details. 6. The system calculates the refundable amount. 7. Customer submits the return request. 8. The system registers the return request. 9. The system forwards the request to the return staff for review. 10. The system updates the item status depending on the decision of the staff.

	Alternative Flow 1 - Partial Return 1. Customer receives the delivered order. 2. The delivered order contains multiple products. 3. Customer selects items for return. 4. Customer submits the return request for the selected item. 5. The system forwards the request to return staff. 6. The system calculates the refundable amount for the mobile phone only. 7. The remaining items in the order remain unaffected. 8. The system updates the phone status depending on the decision of the staff and keep the others unchanged. **Alternative Flow 2 - Misdelivered Item** 1. Customer receives the delivered order. 2. Customer discovers an extra item that was not part of his order. 3. Customer initiates a return request and selects the option Misdelivered Item. 4. Customer provides a description explaining the issue. 5. The system accepts the request. 6. The system forwards the accepted request details to return staff. 7. The system updates the case status to Return in Progress. **Error Flow** 1. Customer receives the delivered order. 2. Customer inspects the product. 3. Customer submits a return request after the return period has expired. 4. The system logs the rejected request. 5. The system displays the rejection reason with a reference to the store's return policy.
Post-conditions / System State	• Request status is changed to in pending state until the return staff respond. • For automated accepted requests as alternative flow 2: the return request status is changed to Return in Progress. • For automated rejected requests as error flow: no return record is created.
Comments	• The system strictly enforces return policy constraints directly.

2.7.2 UC20 – Submit Support Ticket

Writer: Ahmad Hamdan

Use Case Title	Submit Support Ticket
Description	This use case allows a customer to submit a support ticket for an issue related to payment, delivery, return or exchange requests, or other system-related problems. The system validates the submitted information, creates a ticket with a unique identifier, sets the ticket status to Open, and makes it available for customer support staff to handle.
Actors	**Primary Actor:** Customer and Customer Support Staff
Data	• Ticket category (payment, delivery, return, exchange, other) • Ticket description • Optional order ID reference • Customer contact information
Stimulus / Trigger	The customer navigates to the Customer Support section and selects Submit Support Ticket (or New Support Request).
Pre-Conditions	• The customer is logged in and has an active session. • The customer support module is operational. • If the ticket references an order, the referenced order exists in the system.

Work Flows	**Normal Flow** • The customer navigates to the Customer Support section and selects Submit Support Ticket. • The system displays the support ticket submission form. • The customer selects an issue category and enters a description of the issue. • The customer enters a reference order ID if the issue is related to an order. • The customer submits the support ticket. • The system validates the required fields and verifies the reference order ID if provided. • The system creates a new support ticket, generates a unique ticket identifier, and sets the ticket status to Open. • The system stores the ticket with its creation timestamp and associated customer information. • The system confirms successful submission and displays the ticket identifier to the customer. • The system notifies customer support staff that a new support ticket is available. **Alternative Flow 1 - Submission with Missing Details** • The customer selects an issue category and enters a brief or unclear description. • The system creates a support ticket with status Open. • Customer support staff requests additional information through the ticket interface. • The customer provides the requested information, which is appended to the ticket history. **Alternative Flow 2 - Known Issue with Suggested Solution** • The customer enters a description that matches a known issue. • The system displays suggested resolution steps. • The customer accepts the suggested solution and cancels the submission. • The system does not create a support ticket and returns the customer to the support section.

	Error Flow - Missing Required Fields • The customer attempts to submit the form without selecting an issue category or entering a description. • The system blocks submission and displays validation error messages. • No support ticket is created and the system remains in the submission state.
Post-Conditions / Response	On Success: • A support ticket exists with a unique identifier. • The ticket status is Open. • The ticket is linked to the customer account. • Customer support staff is notified. On Failure: • No support ticket is created. • Validation feedback is displayed to the customer.
Comments	• Ticket status transitions after submission are handled in UC18. • Ticket interaction history is stored for auditing and tracking purposes.

2.7.3 UC06 – Process Payment

Writer: Ismail Tarteer

Use Case Title	Process Payment
Description	This use case allows a customer to select a payment method during checkout and complete the payment step using either cash on delivery or bank card payment via an external payment gateway. The system confirms the payment step and generates and stores an electronic invoice for the order. Sensitive payment data such as full card numbers are not stored by the system.
Actors	**Primary Actor:** Customer **Secondary Actor:** External Payment Gateway
Data	• Selected payment method • Card payment authorization response (if applicable) • Order payment status • Electronic invoice
Stimulus / Trigger	The customer proceeds to the payment step during the checkout process.
Pre-Conditions	• The customer has items ready for checkout and is confirming an order. • The system displays available payment options (cash on delivery and bank card).

Work Flows	**Normal Flow** • The system displays the available payment methods during checkout. • The customer selects a payment method. • If cash on delivery is selected, the system records the payment method. • If bank card payment is selected, the system communicates with the external payment gateway. • The payment gateway processes the transaction and returns an approval response. • The system confirms the payment step. • The system generates and stores an electronic invoice for the order. **Alternative Flow - Change Payment Method** • Before confirming the payment step, the customer changes the selected payment method. • The system updates the selected payment method. • The customer continues the checkout process successfully. **Error Flow - Payment Failure or Gateway Unavailable** • The payment gateway declines the transaction or is unreachable. • The system displays a payment failure message. • The payment step is not confirmed. • The customer may retry the payment or select cash on delivery.
Post-Conditions / Response	On Success: • The selected payment method is recorded. • Card payment is approved if bank card payment is used. • An electronic invoice is generated and stored. On Failure: • The payment step is not completed or confirmed. • No invoice is generated.
Comments	• Sensitive payment information is handled by the external payment gateway only. • Payment confirmation is required before order processing continues.

2.7.4 UC14 – Deliver Order

Writer: Sohaib Badaha

Use Case Title	Deliver Order
Description	A delivery personnel delivers assigned chocolate orders to customers, collects payment if required, confirms delivery, and updates the order status in the system.
Actors	**Primary Actor:** Delivery, Customer
Data	• Order and customer details • Delivery address and contact information • Payment method and amount • Delivery status and timestamp • Customer confirmation (signature)
Stimulus / Trigger	• Orders are assigned to the delivery personnel by the Delivery Coordinator. • Delivery personnel accesses assigned orders with status ”Shipped”.
Pre-Conditions	• Delivery Personnel is authenticated with proper permissions. • Orders are assigned and in ”Shipped” status. • Orders are packaged and delivery address is valid. • Delivery vehicle and cash change (if needed) are available.

Work Flows	**Normal Flow** • Delivery Personnel logs into the system and selects an assigned order. • The order is marked as ”Out for Delivery”. • Delivery Personnel arrives at the customer location and verifies identity. • The order is handed over and customer confirms receipt digitally. • Delivery Personnel updates the order status to ”Delivered”. • The system records the delivery timestamp and notifies the customer. **Alternative Flow - 1 Cash on Delivery** • Customer pays the required cash amount after receiving the order. • Delivery Personnel provides change and receipt if needed. • Order is marked as ”Delivered - Cash Collected”. • System records payment and notifies the accountant. **Alternative Flow - 2 Reschedule Delivery** • Customer is unavailable at delivery time. • Delivery Personnel contacts the customer and agrees on a new delivery time. • Order status is updated to ”Delivery Rescheduled”. • System sends a confirmation notification to the customer. **Error Flow - Delivery Failed** • Delivery fails due to incorrect address or unreachable customer. • Order is marked as ”Delivery Failed” with notes. • System creates a support ticket and notifies relevant staff. • Failed attempt is logged and package is returned to the shop.
Post-Conditions / Response	• Order status is updated (Delivered, Cash Collected, Rescheduled, or Failed). • Delivery timestamp and customer confirmation are recorded. • Payment transactions are logged if applicable. • Notifications are sent to customers and relevant staff.
Comments	• The system supports prepaid and cash on delivery orders. • All delivery attempts are logged for tracking and reporting.

2.7.5 UC05 – Place Order

Writer: Omar Hussain

Use Case Title	Place Order
Description	This use case describes how a customer completes the checkout process by reviewing the shopping cart, entering delivery details, applying a coupon if available, selecting a payment method, and confirming the order. The system validates the request, places the order, and prepares it for delivery processing.
Actors	**Primary Actor:** Customer
Data	• Shopping cart details (selected products, quantities) • Customer delivery address • Coupon code (optional) • Payment method details • Product availability and inventory data
Trigger	Customer initiates the checkout process by clicking the *Checkout* button from the shopping cart.
Pre-conditions	1. Customer has items added to the shopping cart. 2. The system and related services are operational. 3. The customer is within a supported delivery area.

Work Flows	**Normal Flow** 1. Customer opens the shopping cart. 2. Customer reviews the selected products and quantities. 3. Customer proceeds to checkout. 4. The system prompts the customer to enter delivery address details. 5. Customer enters a valid delivery address. 6. The system displays the estimated delivery time. 7. Customer enters a valid coupon code (optional). 8. Customer selects a payment method. 9. Customer confirms the order. 10. The system verifies product availability. 11. The system places the order successfully. 12. The system generates a unique order ID. 13. The system displays an order confirmation message. **Alternative Flow - Modify Cart Before Checkout** 1. Customer updates the quantity of a selected product. 2. The system recalculates the total order amount. 3. Customer continues with the checkout process. . **Error Flow 1 - Product Out of Stock** 1. The system detects that one or more products are out of stock. 2. The system notifies the customer of unavailable items. 3. The system prompts the customer to remove or replace the product. **Error Flow 2 - Invalid Coupon Code** 1. The system detects that the coupon is invalid or expired. 2. The system displays an error message. 3. Customer may enter a valid coupon or continue without a discount.
Post-conditions / System State	• The order is saved in the system with status *Placed*. • Inventory quantities are updated. • An electronic invoice is generated. • The order appears in the customer's order history and is ready for processing.
Comments	• The system validates inventory and coupon data before order confirmation. • Errors are handled gracefully and allow the customer to correct issues.

2.8 Activity Diagram MODIFIED

Lead: Ismail Tarteer
Contributors: Mohammad Fareed (modify drawing), Ahmad Hamdan (modify drawing), Omar Husien (review),Sohaib Badaha (writing),

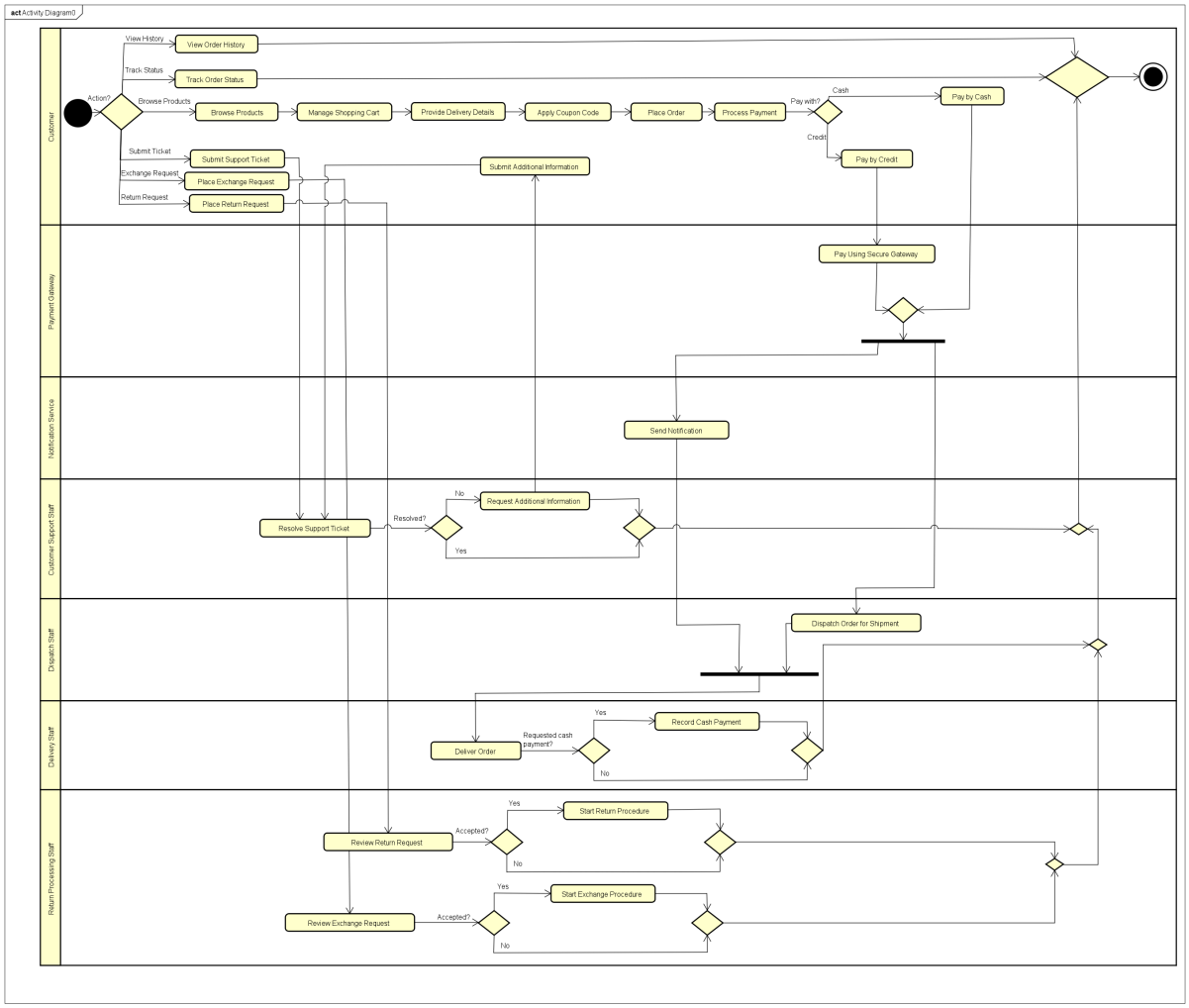


Figure 5: Activity Diagram for the Mobile Phones Shop System

2.9 Instance Activity Diagrams

2.9.1 UC16 - Place Return Request MODIFIED

Writer: Mohammad Fareed

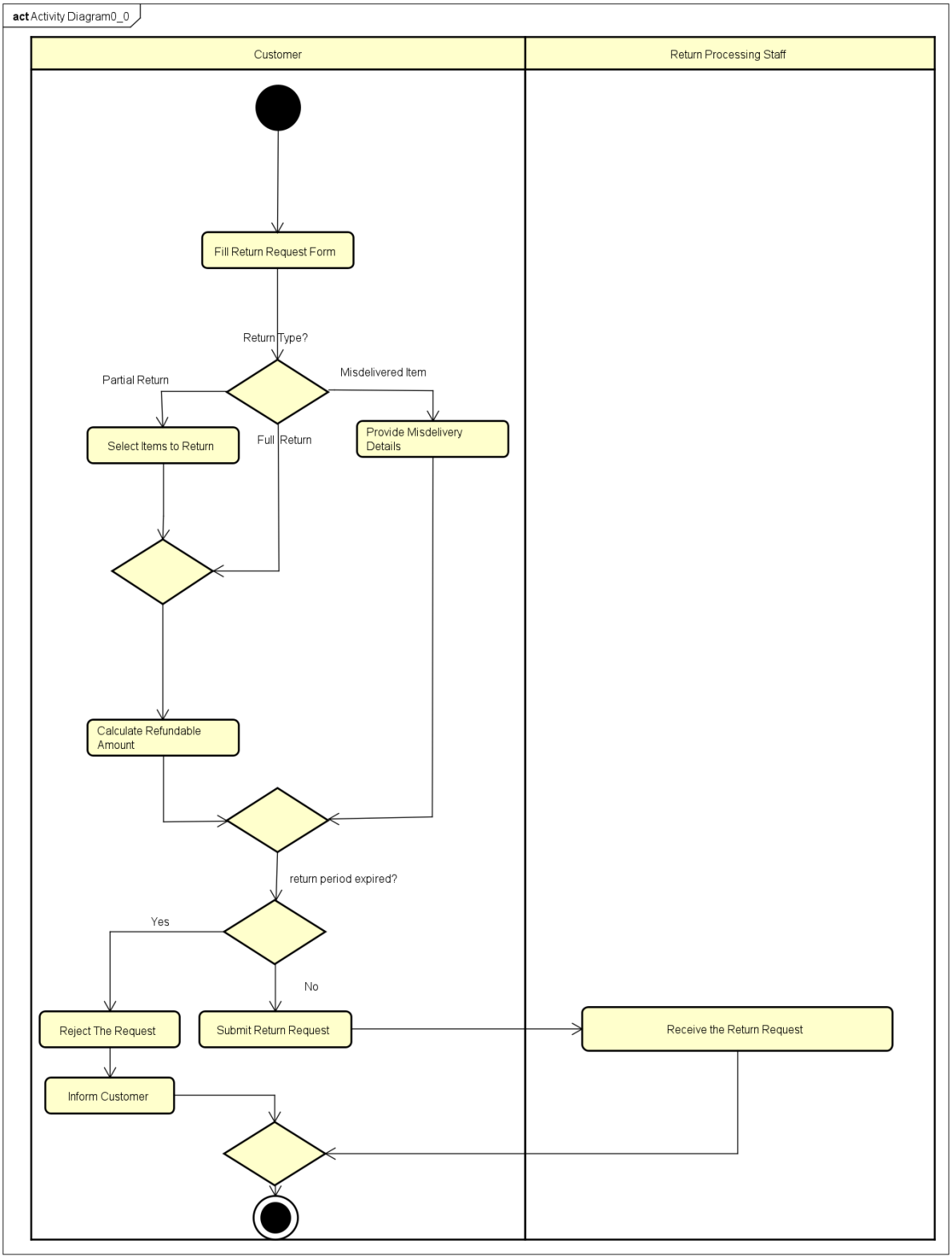


Figure 6: Activity Diagram for Place Return Request Use Case

2.9.2 UC20 - Submit Support Ticket

Writer: Ahmad Hamdan

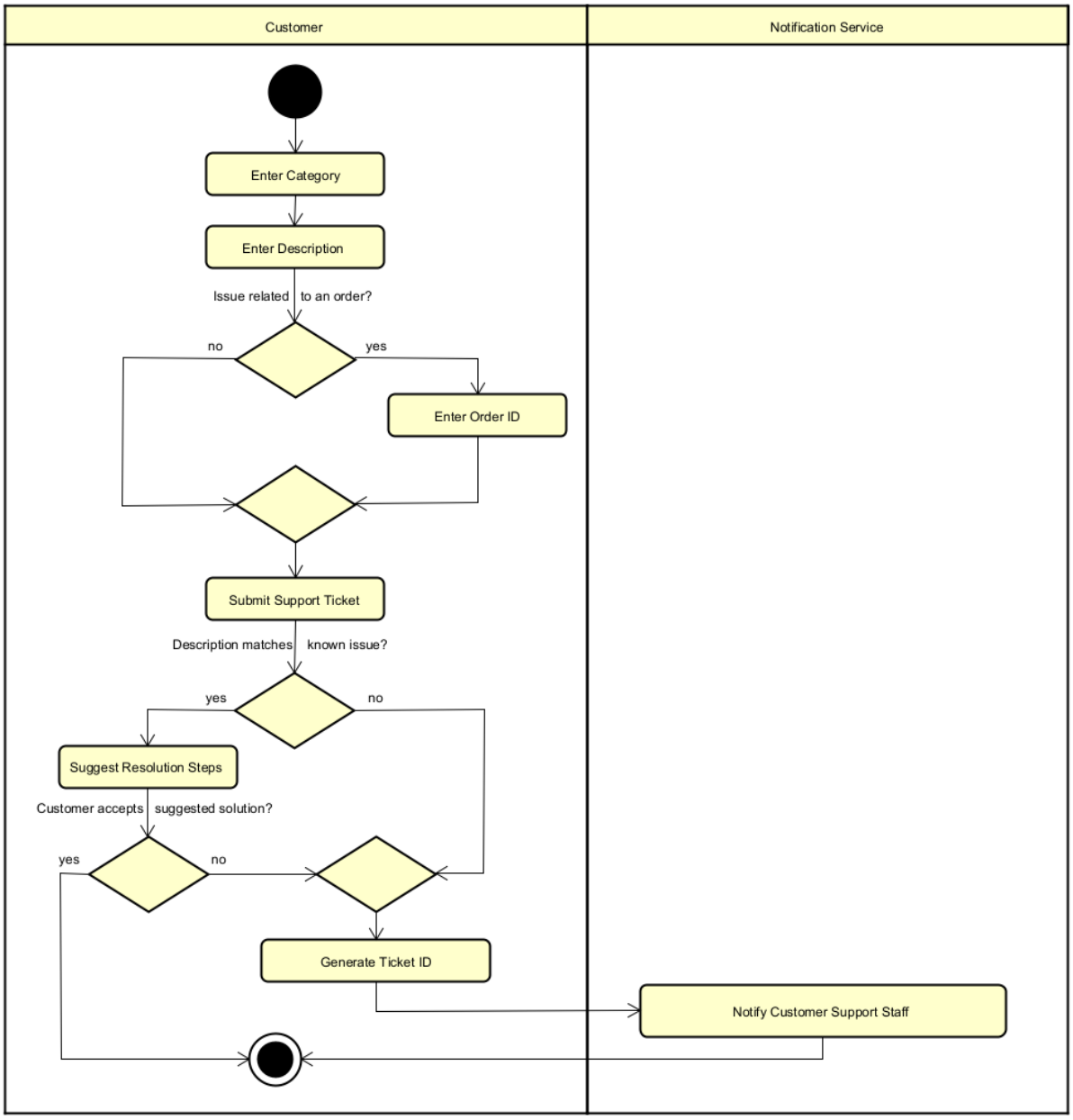


Figure 7: Activity Diagram for Submit Support Ticket Use Case

2.9.3 UC06 - Process Payment

Writer: Ismail Tarteer

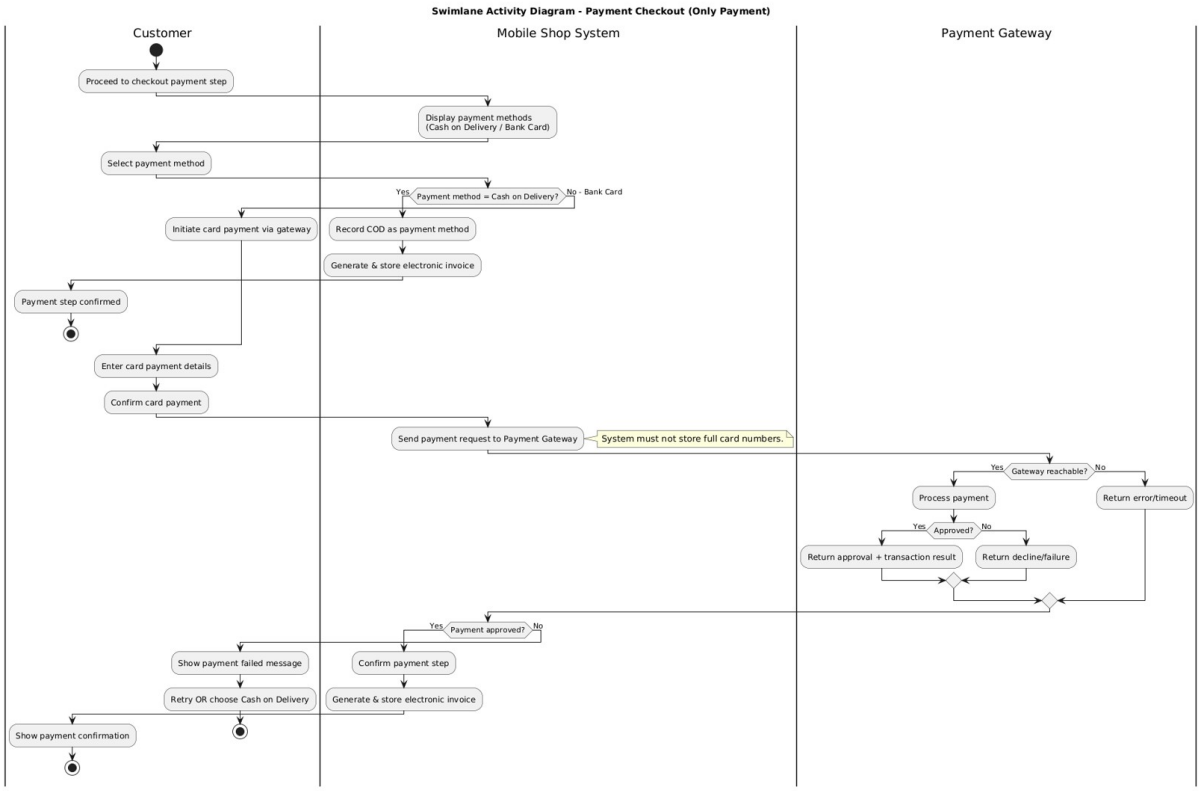


Figure 8: Activity Diagram for Process Payment Use Case

2.9.4 UC14 - Deliver Order

Writer: Sohaib Badaha

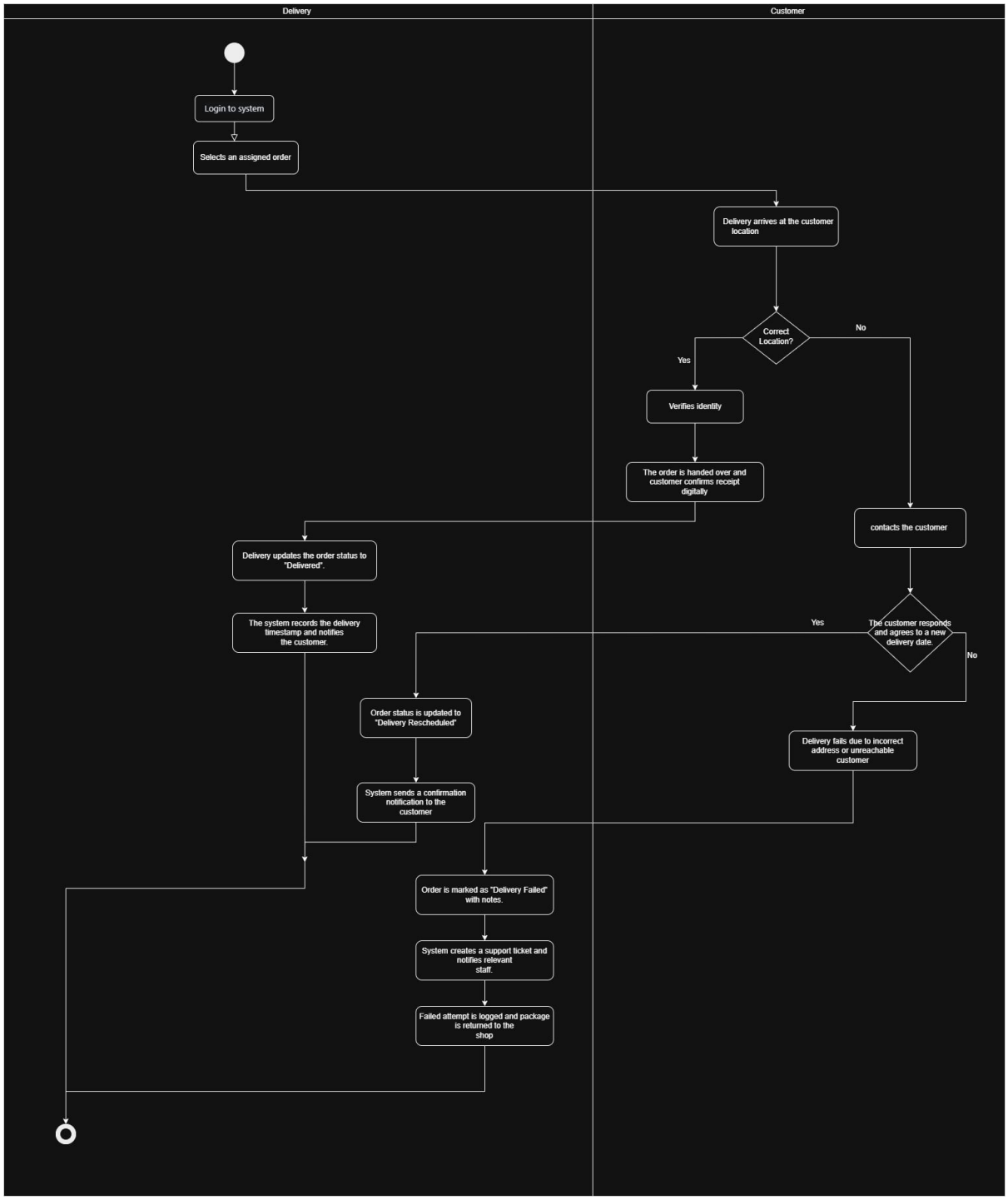


Figure 9: Activity Diagram for Deliver Order Use Case

2.9.5 UC05 - Place Order

Writer: Omar Hussain

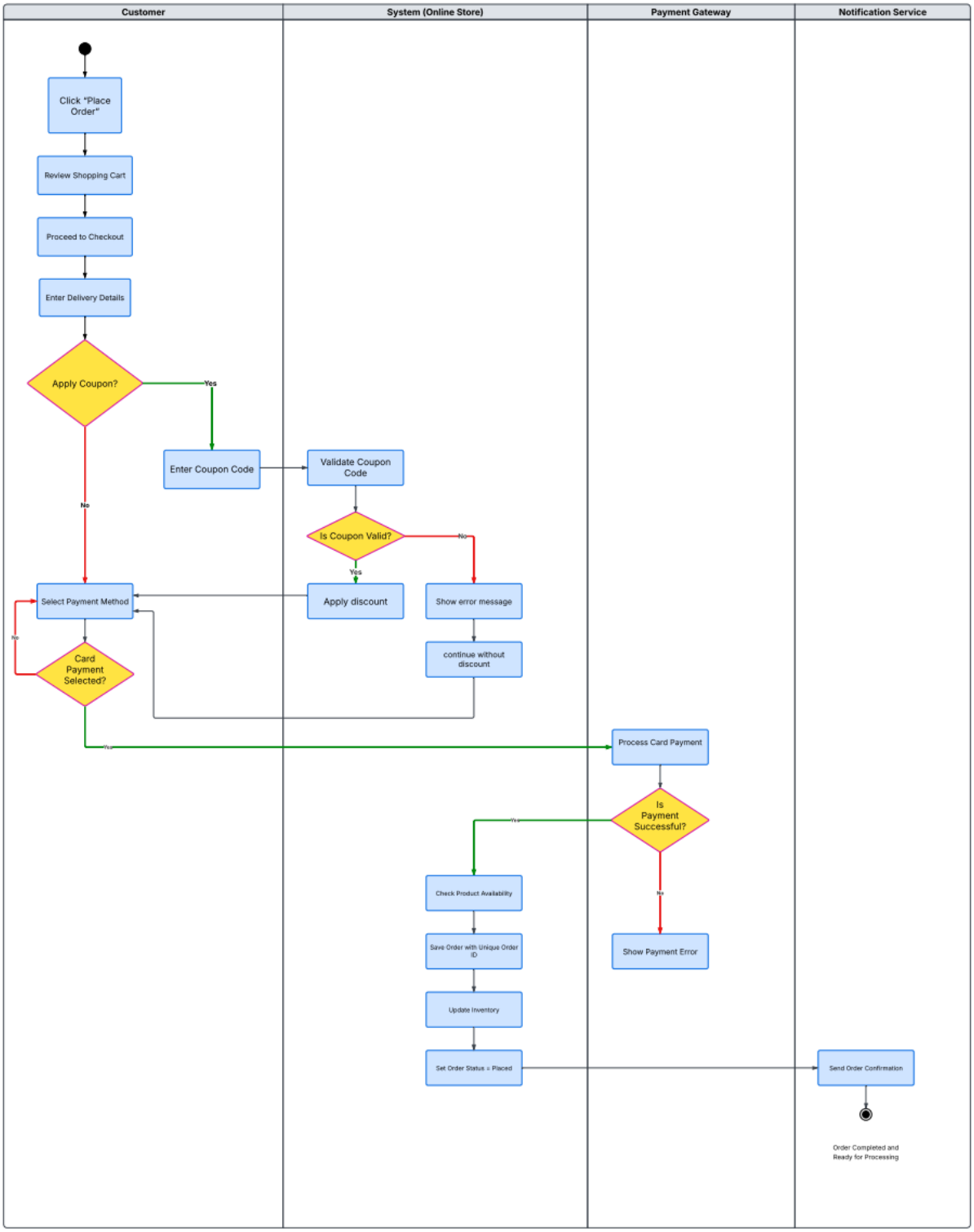


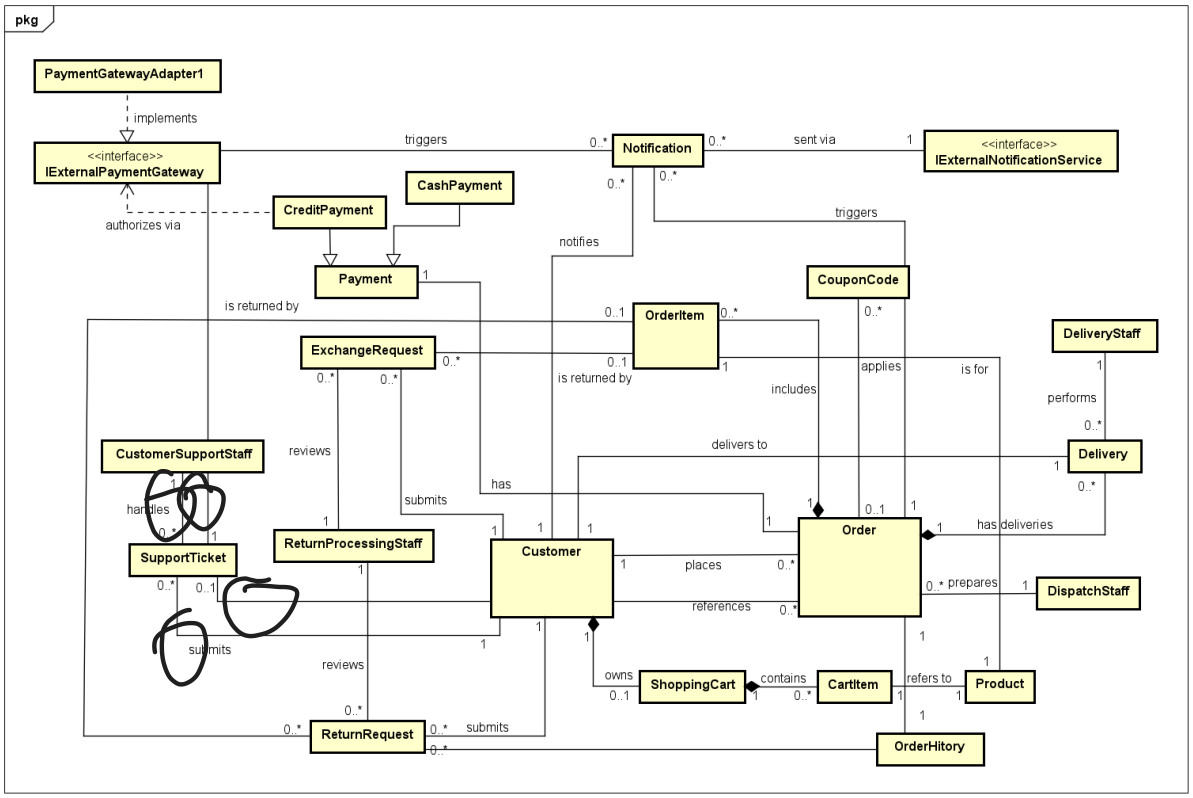
Figure 10: Activity Diagram for Place Order Use Case

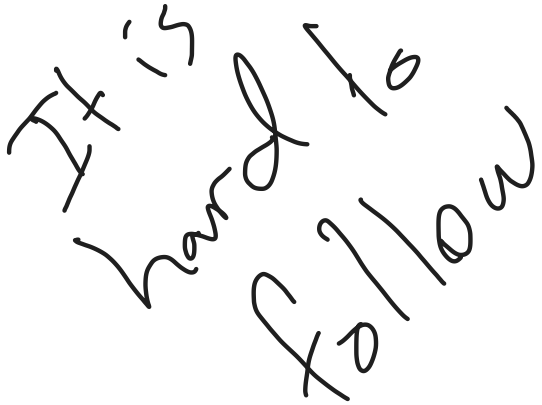
3. System Analysis and Modeling

3.1 System Class Diagrams

3.1.1 Analysis Class Model

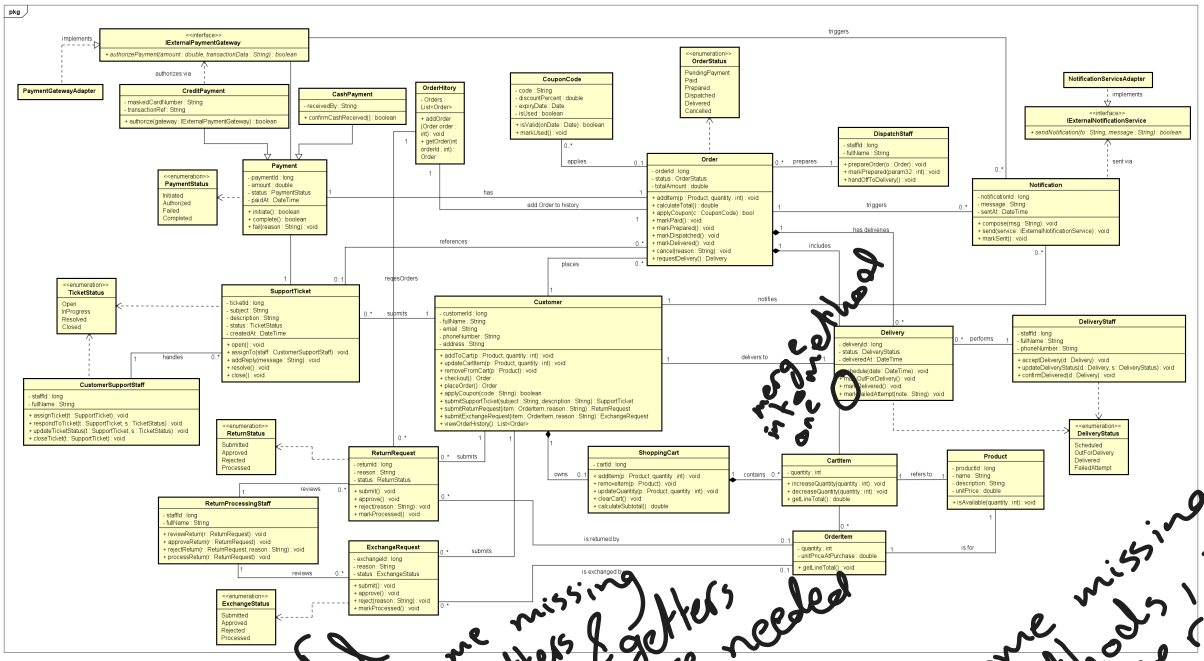
Lead: Sohaib Badaha
Contributors: Omar Hussain (review), Ismail Tarteer (review), Ahmad Hamdan (discussion), Mohammad Fareed (modify drawing)





3.1.2 Detailed Class Model

Lead: Ahmad Hamdan
Contributors: Sohaib Badaha (review), Mohammad Fareed (modify drawing), Omar Hussain (discussion), Ismail Tarteer (discussion)
Due to page size limitations, the full-resolution diagram is provided at the following link: `https://drive.google.com/file/d/11Mznm_HTGr3tOgF-2r7SwJLblRtGGA6R/view?usp=sharing`



3.2 Sequence Diagram

3.2.1 UC16 - Place Return Request

Writer: Mohammad Fareed

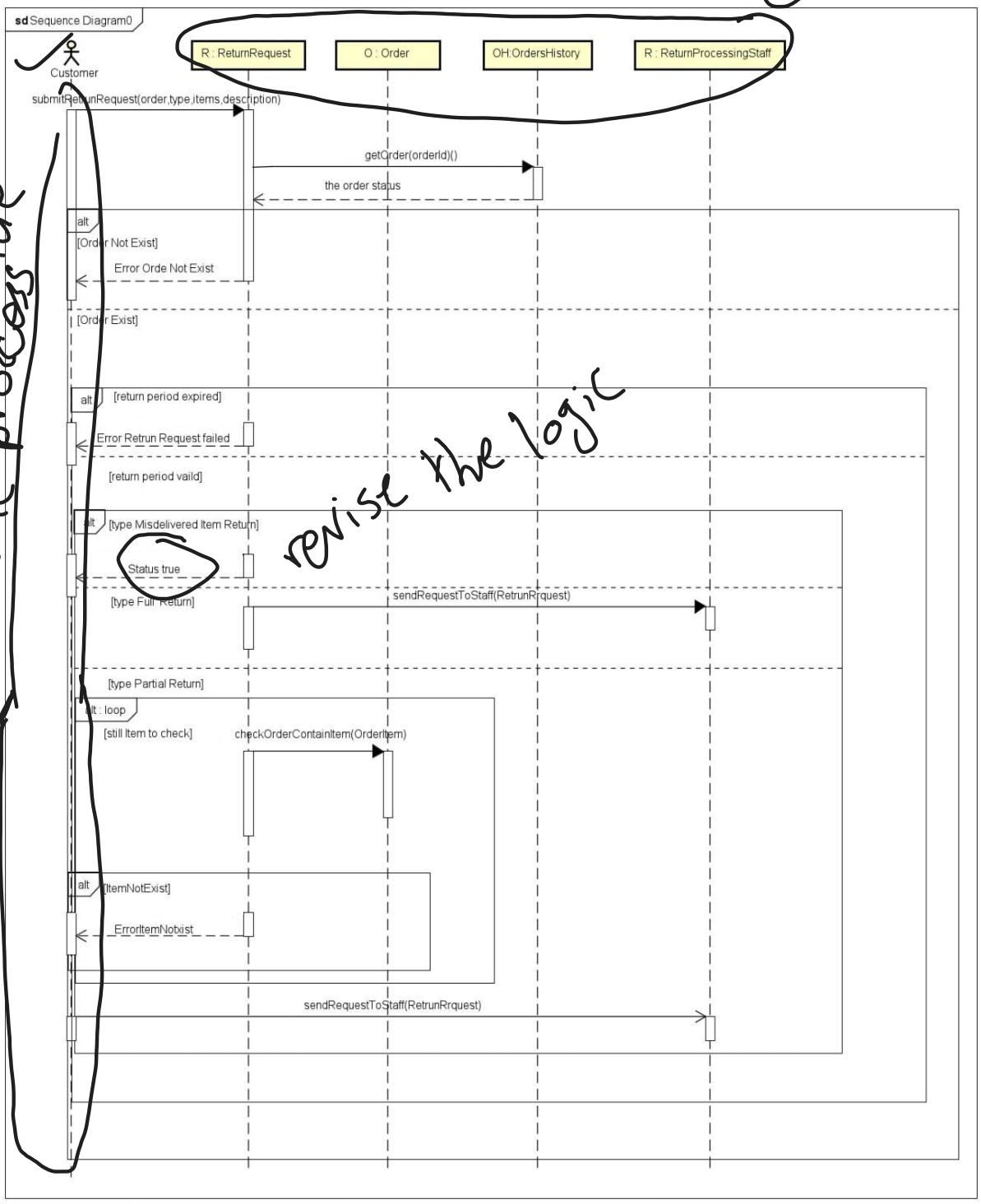
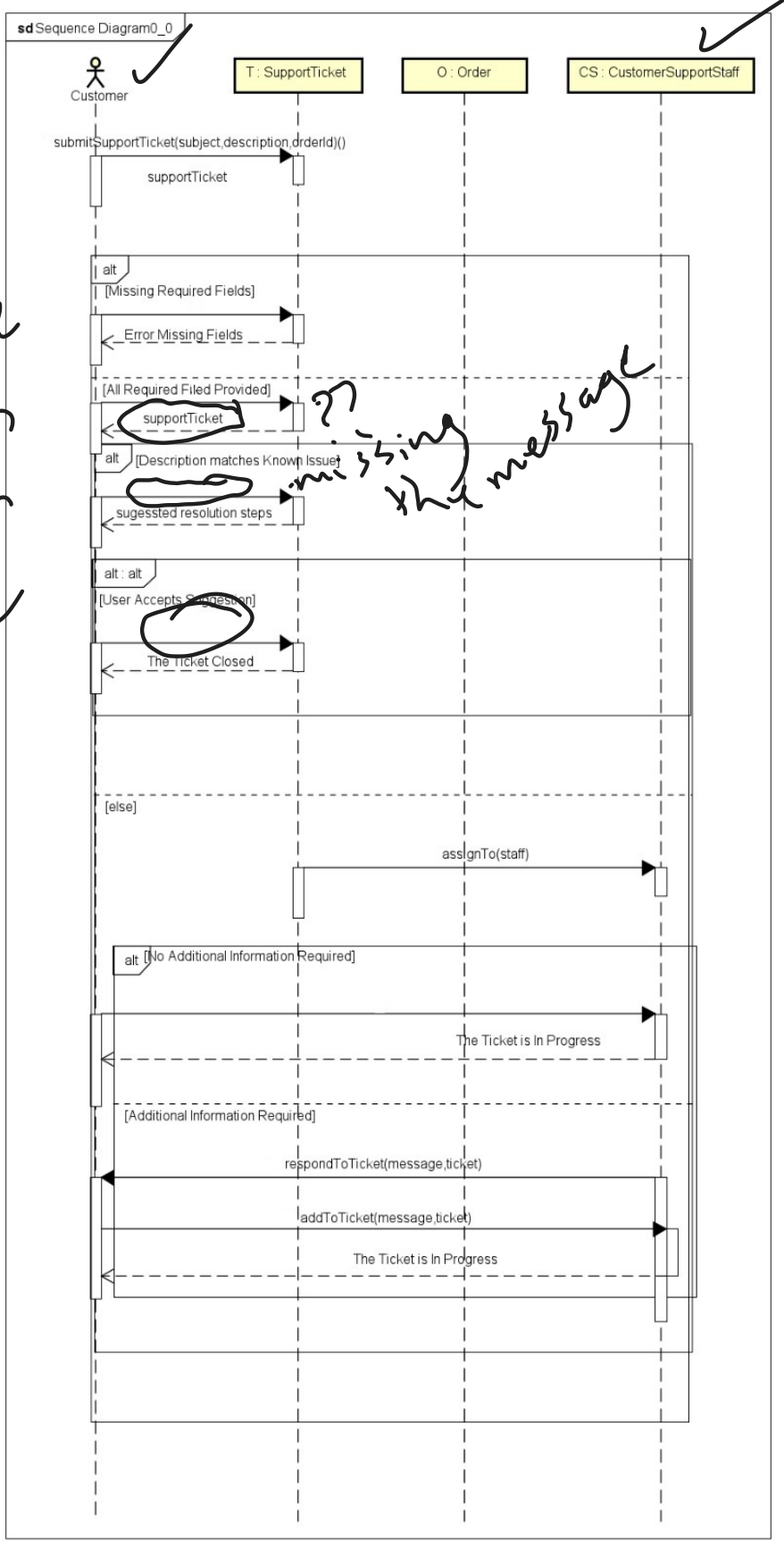


Figure 11: Activity Diagram for Place Return Request Use Case

3.2.2 UC20 - Submit Support Ticket

Writer: Ahmad Hamdan



3.2.3 UC06 - Process Payment

Writer: Ismail Tarteer

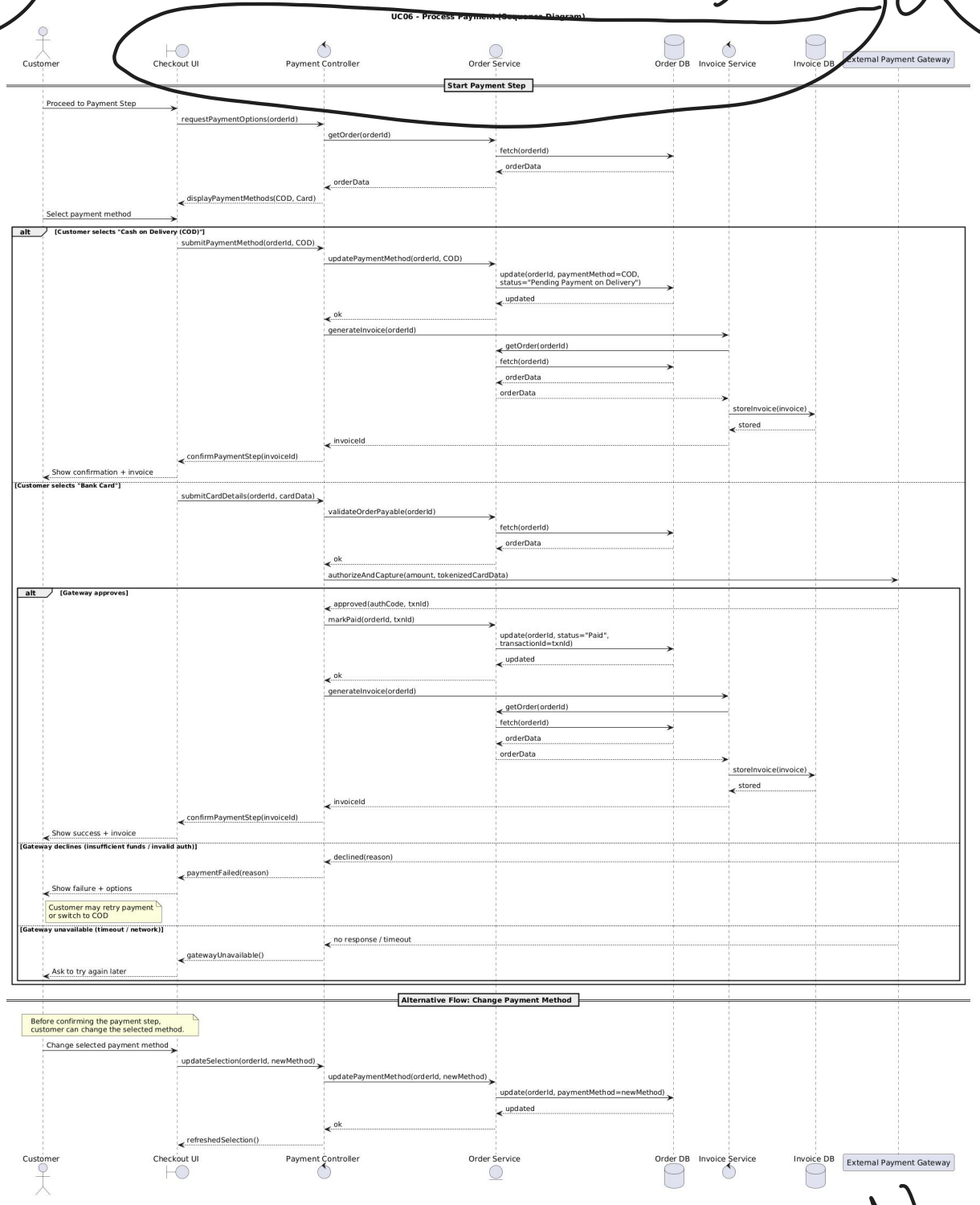


Figure 12: Activity Diagram for Process Payment Use Case

3.2.4 UC14 - Deliver Order

Writer: Sohaib Badaha

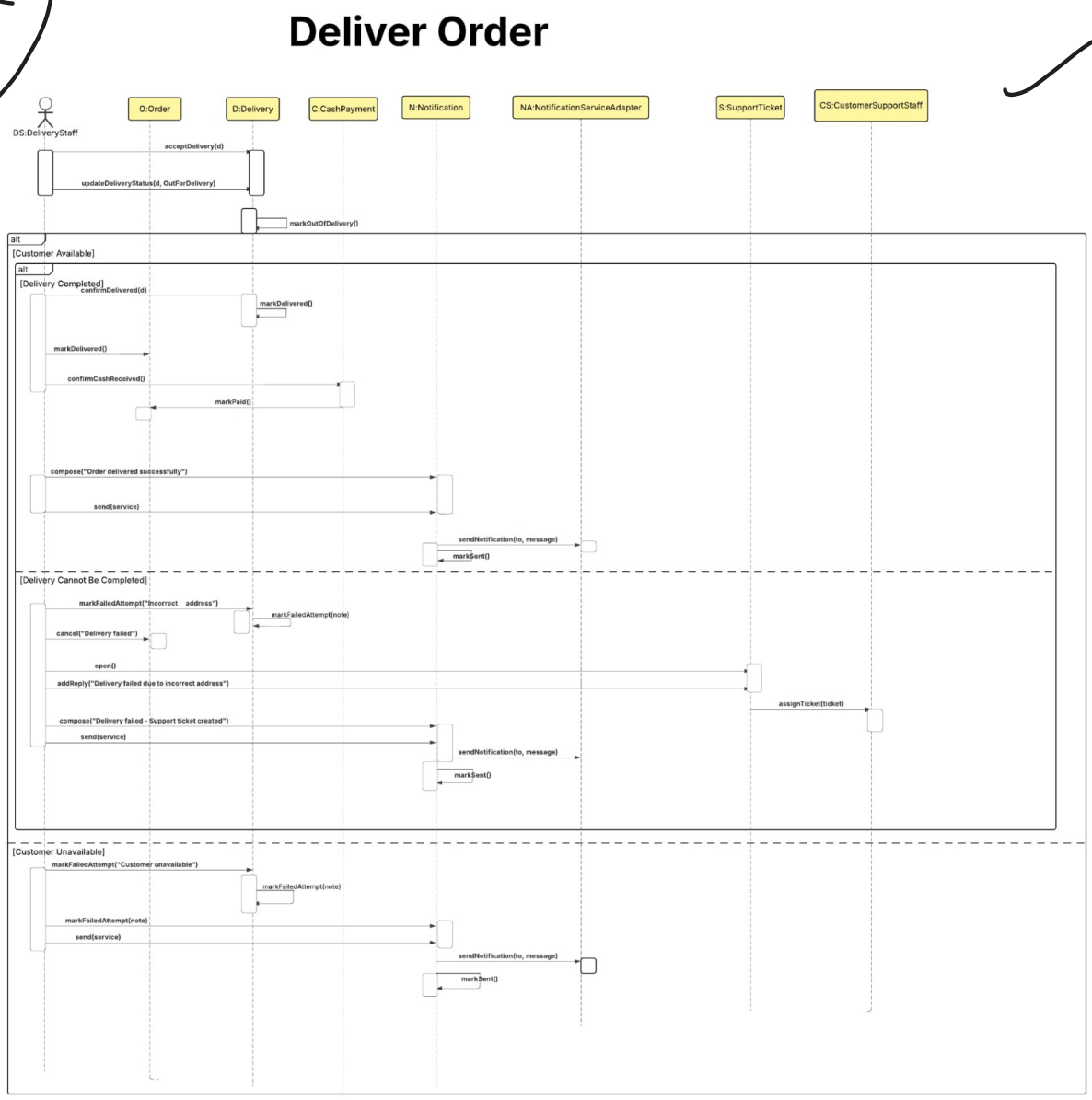


Figure 13: Activity Diagram for Deliver Order Use Case

3.2.5 UC05 - Place Order

Writer: Omar Hussain

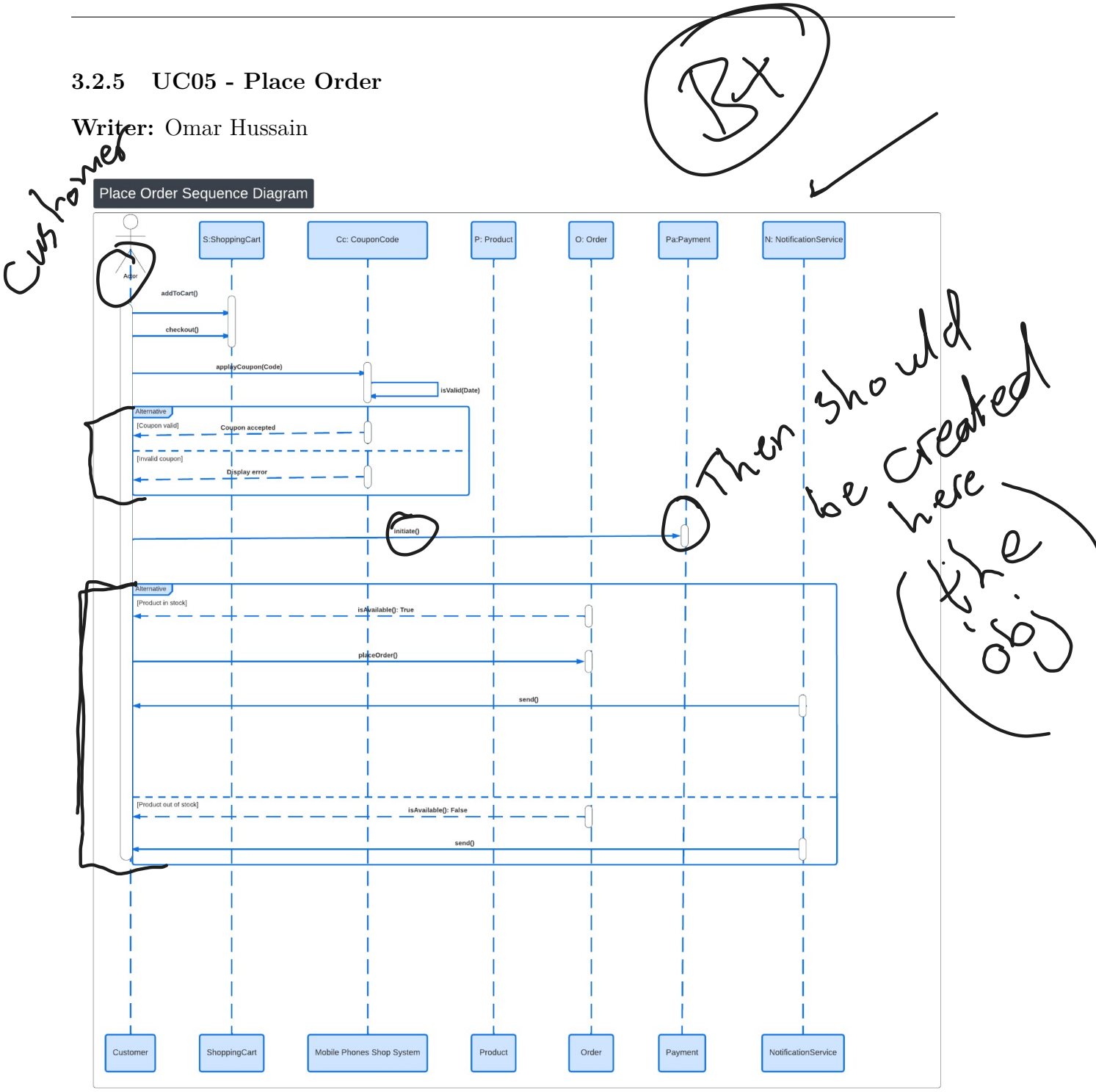


Figure 14: Activity Diagram for Place Order Use Case

4. System Design and Modeling

Lead: Ahmad Hamdan and Mohammad Fareed
(discussion), Omar Husien (review), Ismail Tarteer (writing), Sohaib Badaha (finalization)

4.1 Description of Chosen Design Goals

Goal 1 Low Coupling

Definition Low coupling reduces dependencies between system elements so that changes in one part of the system have minimal impact on others.

How This Goal Is Achieved At the Class Level

1. Core business classes depend on abstract interfaces such as IExternalPaymentGateway and IExternalNotificationService rather than concrete external service implementations.
2. The CreditPayment class interacts only with the IExternalPaymentGateway interface, remaining independent of specific payment providers.
3. External service communication is encapsulated within adapter classes (e.g., PaymentGatewayAdapter) that implement the required interfaces.

At the Component Level

1. The system is divided into functional components such as Order Management, Payment Processing, and Notification Handling.
2. Internal components communicate with each other through well-defined interfaces and do not interact directly with external systems.
3. Access to external services is restricted to the relevant component and occurs only through the corresponding interface.

Goal 2 High Cohesion

Definition High cohesion means that each class or component has a clear, focused responsibility, and its attributes and operations are strongly related to that responsibility.

How This Goal Is Achieved At the Class Level

1. Each class is designed around a single purpose (Order manages order lifecycle, Payment handles payment state, and SupportTicket manages customer support requests).
2. Attributes and operations are directly aligned with the class responsibility (e.g., Order.calculateTotal(), CreditPayment.authorize(), and SupportTicket.updateStatus()), avoiding unrelated functionality in the same class.

At the Component Level

1. Related functionality is grouped into dedicated components (Order Management, Payment Processing, Notification Handling, Delivery Management, and Support).
2. Each component maintains clear internal responsibilities (e.g., Notification Handling is responsible only for composing and sending messages).
3. External integrations are isolated within their relevant component (payment integration remains within the Payment Processing component through PaymentGatewayAdapter), keeping other components focused on business logic only.

Goal 3 Reliability and Robustness

Definition The system is designed to continue operating correctly and to maintain a consistent state even when failures occur, particularly during interactions with external services such as payment systems.

How This Goal Is Achieved At the Class Level

1. Core business classes such as Order, Payment, Delivery, ReturnRequest, and SupportTicket explicitly model their lifecycle using status attributes, allowing both normal and failure states to be clearly represented.

At the Component Level

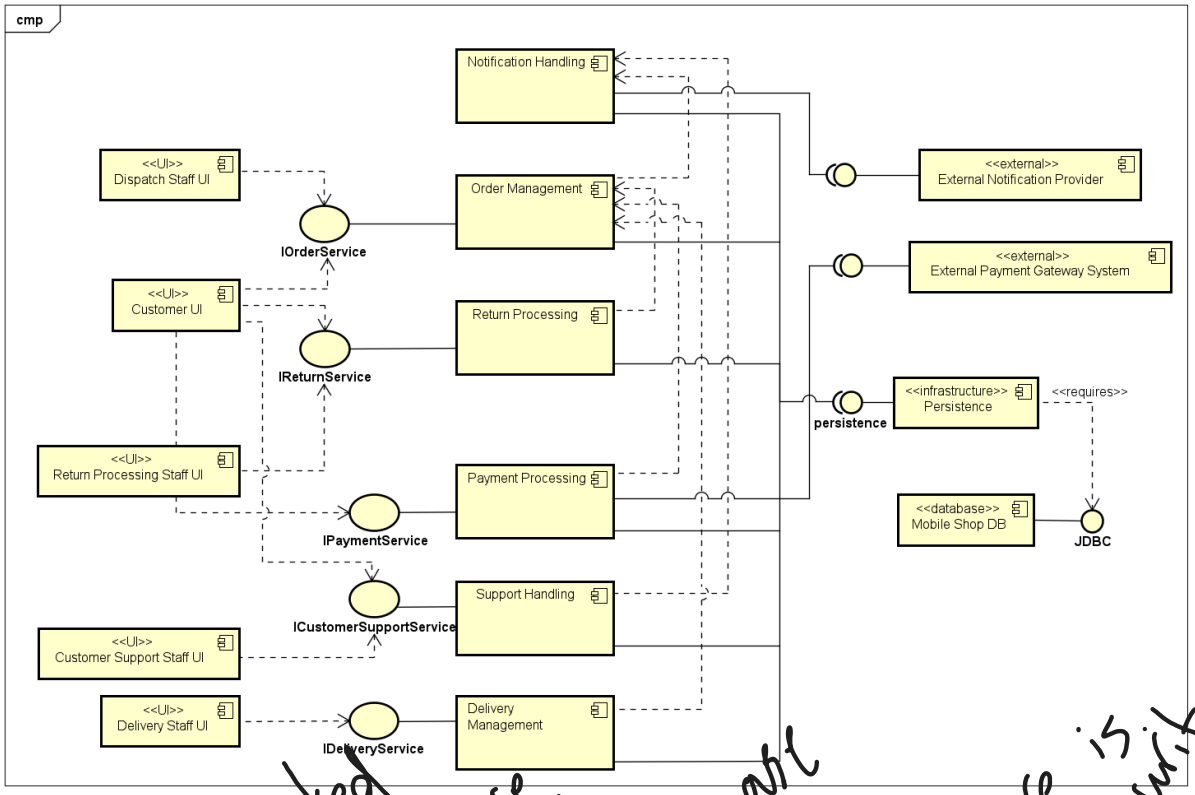
1. External systems are accessed only through well-defined interfaces, isolating failures in external services from the core system components.
2. Each functional component (Order Management, Payment Processing, Delivery Management, and Support Handling) is responsible for handling its own failure conditions without propagating errors to other components.
3. Core business workflows remain operational even when external interactions fail, allowing users to retry operations or select alternative options without system interruption.

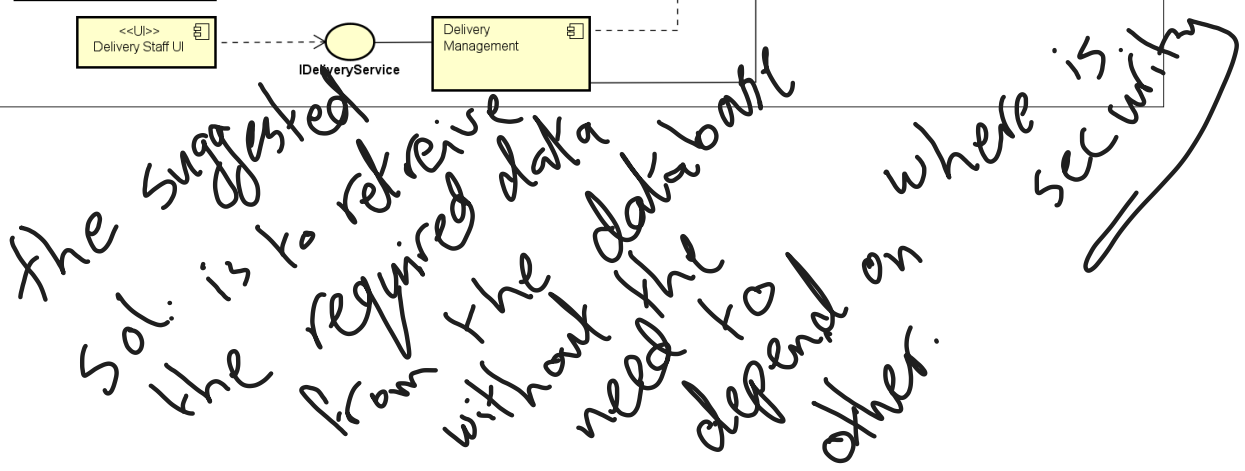
4.2 Component Diagram

This approach was discussed during the presentation, and we explained to Dr. Tasnim that it is the most suitable way to achieve low coupling. She agreed the need of this approach since all components need to interact with the Order Management component and mentioned that she would review it further and suggest a better alternative if one exists.

Lead: Mohammad Fareed

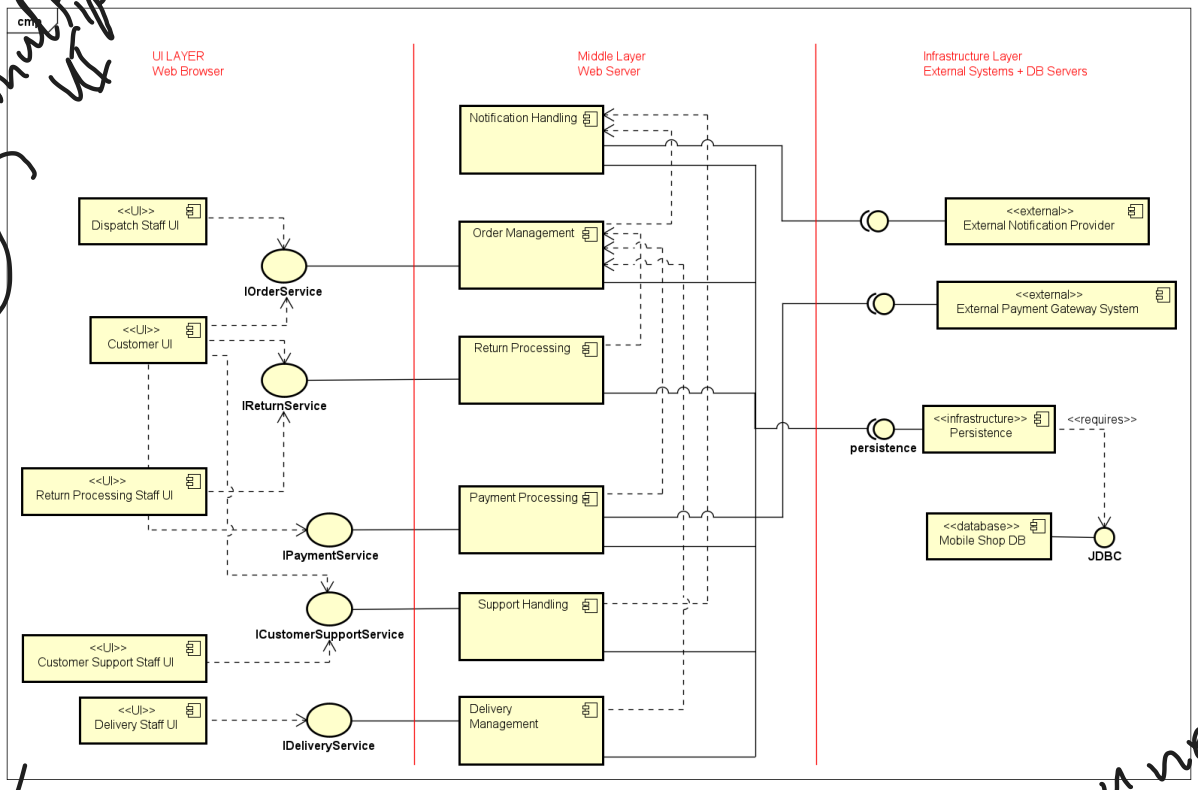
Contributors: Omar Hussain (review), Ahmad Hamdan (modify drawing), Ismail Tarteer (discussion), Sohaib Badaha (discussion)





4.3 Overall Architecture Diagram

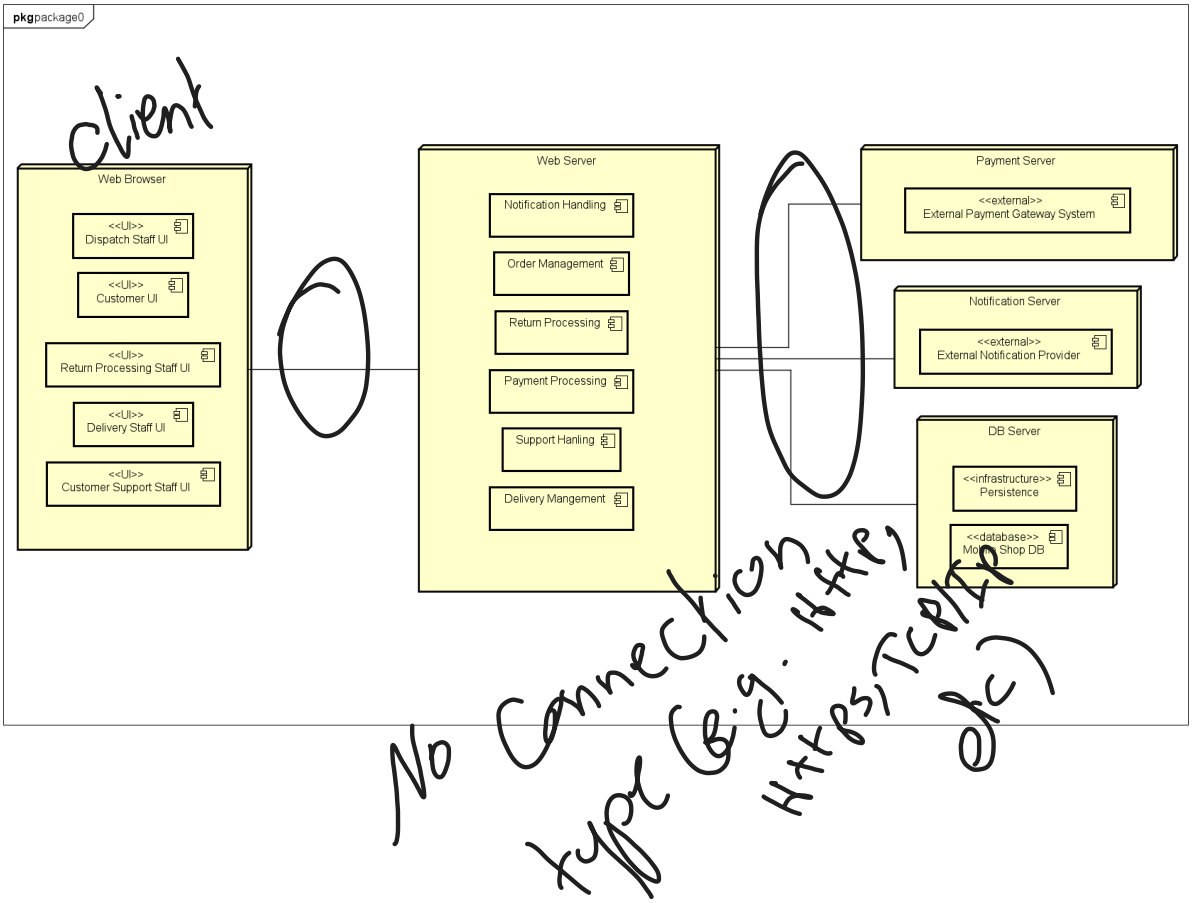
Lead: Ismail Tarteer
Contributors: Mohammad Fareed (modify drawing), Sohaib Badaha (review), Omar Hussain (discussion), Ahmad Hamdan (discussion)



The system follows a layered (three-tier) software architecture composed of a User Interface layer, a Middle (Web Server) layer, and an Infrastructure layer. This architecture promotes low coupling by ensuring that each layer interacts only with the layer directly beneath it through well-defined service interfaces, preventing direct dependencies between the user interface, business logic, and infrastructure services. High cohesion is achieved by grouping related responsibilities within dedicated components such as Order Management, Payment Processing, Return Processing, and Support Handling, each focusing on a single business concern. The separation of concerns inherent in the layered architecture improves maintainability and scalability, allowing changes to external systems such as payment gateways or notification providers to be handled within the infrastructure layer without impacting other parts of the system. Additionally, isolating external services and persistence mechanisms enhances reliability and robustness by limiting the impact of failures, while centralizing sensitive operations in controlled layers strengthens security. Overall, the layered architecture provides a clean, flexible, and extensible system structure that does not mix the business logic with the infrastructure.

4.4 Deployment Diagram

Lead: Omar Hussain
Contributors: Ahmad Hamdan (modify drawing), Sohaib Badaha (review), Ismail Tarteer (discussion), Mohammad Fareed (discussion)



Appendix-I

A.1 Minutes of Meetings with Customers

Two formal meetings were held between the development team and the customer representatives to align expectations, gather requirements, and validate system behavior for the mobile phones and accessories shop. These meetings were essential in defining system features, clarifying workflows, and ensuring that customer needs were accurately reflected in the system design.

Meeting 1

- **Date & Time:** Saturday, 6 December 2025, 11:00 AM
- **Location:** Face-to-face (Birzeit University Campus)
- **Attendees:**
  - **Development Team:** Mohammad Fareed, Ahmad Hamdan, Omar Hussien, Ismail Tarteer, Sohaib Badaha
  - **Customer Representatives:** Tasneem Shelleh, Eman Hamed, Hala Mosafer, Shatha Khdair, Sadeel Assi
- **Discussion Topics:**
  - General overview of the online mobile phones and accessories store.
  - Identification of the main services offered to customers.
  - Customer buying process: browsing products, adding items to cart, checkout, order processing, and delivery.
  - Supported payment methods, including cash on delivery and bank card payment.
  - Delivery workflow, supported delivery areas within the West Bank, and expected delivery time (1–2 days).

Meeting 2

- **Date & Time:** Thursday, 11 December 2025, 5:00 PM
- **Location:** Online meeting
- **Attendees:**
  - **Development Team:** Mohammad Fareed, Ahmad Hamdan, Omar Hussien, Ismail Tarteer, Sohaib Badaha
  - **Customer Representatives:** Tasneem Shelleh, Eman Hamed, Hala Mosafer, Shatha Khdair, Sadeel Assi
- **Discussion Topics:**
  - Review and validation of drafted user requirements.
  - Clarification of missing or ambiguous requirements.
  - Inventory behavior when products are out of stock.

- Implementation of coupon-based discount functionality during checkout.
- Definition of order status stages and customer visibility.
- Handling customer support requests and tracking issues until resolution.

A.2 Minutes of Meetings with the Team

Throughout the project, the development team conducted regular internal meetings to ensure effective coordination, progress tracking, and timely completion of tasks. On average, at least one meeting was held per week, either online via Google Meets or through face-to-face gatherings on the university campus. These meetings focused on reviewing completed tasks, planning upcoming activities, discussing design decisions, and resolving technical or documentation-related issues.

During critical periods, such as phase submissions or major deliverables, the frequency of meetings increased to two or three sessions per week. These intensive meetings allowed the team to adapt plans, redistribute tasks when necessary, and ensure alignment across all project artifacts, including requirements, diagrams, and reports. Team attendance was consistently high, with all members actively participating in discussions and contributing to decision-making.

In addition to scheduled meetings, continuous communication was maintained through the group chat on Facebook Messenger. This channel was used for sharing updates, assigning tasks, clarifying questions, and tracking progress between meetings. The combination of regular meetings and constant written communication played a crucial role in maintaining consistency, resolving issues efficiently, and supporting smooth collaboration throughout the project lifecycle.