



**Faculty of Engineering and Technology**

**Computer Science Department**

COMP433 – Group Assignment phase 1

<<*Reliable Services (RS)*>>



Students Name	Student Number
Ansam Hamayel	1220463
Yara Hamad	1220564
Doaa Assi	1220504
Muna Atta	1202262
Hiba Awwad	1220722

Instructor's Name: Samer Zein

## Table of Contents:

COMP433 – Group Assignment phase 1 .....	1
Phase 1: .....	4
Introduction .....	4
Proposed Solution .....	5
Software development process.....	8
Phase 2: .....	9
User Requirements: .....	9
System Requirements:.....	10
Effort and Cost Estimation: - .....	14
Phase 3: .....	16
Task 3.1: .....	16
Scenario for tracking customer's order status (Ansam Hamayel) .....	16
Scenario for Submitting a Maintenance Request (Mona Atta) .....	18
Scenario for Payment via app (Yara Hamad).....	21
Scenario following up on the inventory status of shortages and overages (Doaa Assi) .....	23
Scenario for feedback (Hiba Awwad) .....	25
Task 3.2: .....	27
Task 3.3: .....	29
Use Case: Track order status (Ansam Hamayel) .....	29
Use Case: Submit a Maintenance Request (Mona Atta).....	31
Use Case: Payment via app (Yara Hamad) .....	33
Use Case: follow up on the inventory status of shortages and overages (Doaa Assi) .....	35
Use-Case: feed back (Hiba Awwad).....	37
Task 3.4: .....	40
Instance Activity Diagrams for tracking customer's order status (Ansam Hamayel).....	40
Instance Activity Diagrams for Submitting a Maintenance Request .....	41
(Mona Atta) .....	41
Instance Activity Diagrams for payment via app (Yara Hamad) .....	42

Instance Activity Diagrams for following up on the inventory status of shortages and overages (Doaa Assi) .....	43
Instance Activity Diagrams for Feedback (Hiba Awwad): .....	44
Phase 4:.....	45
Task 4.1: .....	45
System Class modelling and Analysis: .....	45
2. DETAILED ANALYSIS/ CLASS Diagram: .....	50
Task 4.2: .....	51
System Sequence modelling and Analysis for tracking customer's order status (Ansam Hamayel):.....	51
System Sequence modelling and Analysis for pay for order (Yara Hamad).....	52
System Sequence for following up on the inventory status of shortages and overages (Doaa Assi): .....	53
System Sequence modelling and Analysis for feedback Submission Process (Hiba Awwad) .....	54
System Sequence modelling and Analysis for Submit a Maintenance Request (Mona Atta): .....	55
Task 4.3: .....	56
System Design Goals: .....	56
Task 4.4: .....	57
System Component Design:.....	57
Task 4.5: .....	58
System Deployment Diagram: .....	58

## Phase 1:

### Introduction

With the rapid development of technology until this day, we see that it is important for competitive companies to efficiently and effectively manage information technology. Due to the strong competition in the market, Services-Inc a company that focuses on business, home IT services and provides various IT services in terms of networking, Wi Fi, PC and laptop maintenance, printing machines maintenance, and so on. The company faces various difficulties in handling its IT operations, particularly due to the manual using of Excel and Google sheets. This results in low customer satisfaction, decreased productivity, reduced accuracy, and inefficiency caused by the lack of system integration.

The absence of a unified system creates major problems for our company, which offers essential services such as networking, Wi Fi, PC and laptop maintenance, printing machines maintenance and hardware repairs. For instance, when a client requests a maintenance service for their company, the receptionist records the customer's information, address, and issue description on a physical form before forwarding it to the maintenance department which impacts service efficiency and customer satisfaction. From the other side, Employee information is kept in Excel sheets within the HR department, including tracking leaves, salary payments, causing challenges in managing employees and monitoring their performance.

Recognizing the limitations of the current manual system, which relies heavily on Excel and Google Sheets, it is evident that the lack of integration and centralization is hampering the company's operations. Fragmentation of data across multiple platforms leads to low efficiency, delays, and difficulty in accessing and analyzing information. To overcome these challenges, we propose to implement a comprehensive IT service management system tailored to the needs of Services-Inc. The proposed solution will standardize and streamline the company's operations by automating processes and integrating all departments into a single platform. Key features include a maintenance request portal to improve service efficiency, smart scheduling to optimally allocate resources, and a robust HR management system to enhance employee control. Additionally, the system provides real-time analytics to support decision making, predictive inventory management to avoid over- or under-stocking issues, and an advanced complaint management system to address customer concerns who using the mobile app. With mobile accessibility, offline functionality, and seamless integration with tools like Bisan Accounting, the solution ensures flexibility, scalability, and accuracy. This integrated approach will contribute to enhanced customer satisfaction, increased operational efficiency, and support the company's growth and competitiveness in the market.

## Proposed Solution

We propose an advanced integrated software system designed to unify and simplify all parts of the company's operations. The solution gets rid of paper-based processes and manual interventions, leading to smooth communication between departments and improving overall efficiency. The system is designed to provide management with real-time oversight of various business functions, including sales, customer interactions, employee management, revenue tracking, and complaint resolution.

### Key Features and Functionalities:

#### 1. Portal for customers to submit maintenance requests

Customers can submit maintenance requests via a convenient mobile application designed for easy use. This feature removes the need for face-to-face interactions or telephone conversations, allowing customers to provide service specifications, available appointment times and any extra details required by the maintenance team. Once submitted, customers receive automated notifications confirming their requests and giving updates on the status. When a request is submitted, the system automatically creates a service ticket, which is logged into the company's database and promptly sends it to the maintenance department. This automation significantly decreases response times and enhances customer satisfaction. Where it provides user-centric design, ease of use, and real-time notifications so enhance customer engagement and loyalty.

#### 2. Automated scheduling of maintenance and allocation of resources

The system contains a smart scheduling algorithm that maximizes the distribution of service employees and vehicles considering availability, workload and closeness to the customer location. This ensures resource efficiency and minimizes travel time. When a maintenance request is approved, the system verifies the availability of service employees and company vehicles. After that, it forms a team, estimates service duration, and schedules the appointment based on the earliest possible availability. Customers are informed about the appointment details and have the option to change the time during the company's operating hours. Where it offers High system performance, efficient resource management and minimized operational costs through optimized scheduling.

#### 3. Advanced Human Resources (HR) Management Module

This feature automates HR processes like managing leaves, scheduling trainings, monitoring performance, and processing payroll. The system keeps a comprehensive employee database, allowing HR managers to easily access information and generate

reports automatically. The HR module collaborates with various systems within the company to extract data on employee attendance, performance metrics, and training progress. It automatically computes payroll based on attendance and overtime, creating pay slips for employees and reports for HR management. Where it provides flexibility to handle growing workforce data, data accuracy, and decrease manual HR tasks.

#### 4. Dashboard for reporting and analyzing business data in real time

The system offers management with a dashboard that offers insights into sales performance, revenue trends, customer satisfaction, and other key performance indicators (KPIs). The dashboard includes interactive charts and graphs for in-depth analysis and quick decision-making. Data is constantly collected from various departments and analyzed in real time. Managers have the ability to create personalized reports, filter data by specific metrics, and establish automated alerts for important business thresholds like sales targets and inventory levels. Where it provides real-time data analytics, data visualization, and decision support for enhance strategic planning.

#### 5. Management of maintenance inventory through the use of predictive analysis

The system provides a strong inventory management module specifically designed for the maintenance department. It tracks the availability of spare parts, tools, and equipment, and updates stock levels automatically when parts are used. In addition, it includes predictive analytics for predicting inventory requirements using historical usage data. The system will notify automatically when stock levels drop below a set limit to make sure crucial parts are always available. It uses machine learning algorithms to forecast future inventory requirements, resulting in a reduction of stockouts and overstocking. Where it provides an efficient inventory management, decreased downtime, and enhanced service continuity.

#### 6. Improved System for Handling Complaints

This module allows both customers and employees to submit complaints or issues, which are then categorized and prioritized according to their severity automatically. Management has the ability to monitor the progress of resolving complaints and analyze patterns in order to enhance the quality of service. The system uses natural language processing (NLP) to analyze complaint descriptions and automatically assigns them to the appropriate department. Automated escalation procedures are in place for unresolved complaints, ensuring timely follow-ups and resolution. By enhancing customer satisfaction through quick problem-solving, decreased response times, and thorough complaint monitoring, it achieved an improvement.

## 7. Integration with Current Market Tools (e.g., Bisan Accounting Software)

Our system is designed to integrate seamlessly with well-known tools in the local market, like the Bisan accounting software. This enables more efficient financial reporting and accounting procedures, reducing data entry errors and enhancing financial accuracy. The system uses APIs to integrate with external software facilitating the transfer of data across different platforms. Financial data such as sales, expenses, and payroll can be exported automatically to the accounting system to simplify the process. Where it enhanced interoperability, decreased manual data entry, and enhanced accuracy of financial data.

## 8. Mobile Access and Offline Mode

The system enables customers to easily reach it through mobile apps, allowing them to access it without an internet connection. This enables customers to monitor and manage their services without interruption, even in regions with limited or no network connectivity. While offline, customers can access their previously stored data, such as account information and active services. They are also able to submit new requests or update their information and these modifications are stored on their devices. After internet connection is back, the app automatically synchronizes the data with the platform, making sure all records are updated instantly and accurately.

## 9. Automated process for backing up and recovering data

The system provides advanced automatic backup and data recovery capabilities to guarantee service continuity and data protection for customers using the mobile app and employees using the web interface.

### For web-based employees:

The system automatically creates regular backups of essential business data like customer records, daily transactions, and reports. If technical issues or data loss occur, employees can quickly recover data from saved backups, ensuring business continuity and reducing downtime.

### For app customers:

The application ensures that customer information, such as previous orders, personal details, and transaction records, are regularly backed up. In case of data loss or technical issues, customers can easily recover their data without needing manual help or long waiting times.

## Software development process

For this project, we will implement the Agile development methodology where we will release the project in incremental parts, allowing you to use it after each part is completed. You have the option to provide feedback and ask for any changes during the process. We chose it for its flexibility (adapting quickly to changes in requirements), improving efficiency (teams can increase productivity and focus on delivering value by breaking down work into smaller tasks and focusing on high priority things), improving user satisfaction (we make continuous improvements and adjustments based on user feedback throughout the programming and development process), incremental planning, ... etc.

The approximate time we need to complete the project is **8 Months**

This program Estimated Total Cost: **80000 - 100000 \$**

The project will be delivered in 3 releases:

**Release 1:** Upload and link the company's components data to the server, and send the maintenance request to the maintenance department electronically (inside the company)  
approximate time: **3 months**

**Release 2:** Submit maintenance request electronically from home, and create an effective schedule for providing maintenance service outside the company.  
approximate time: **3 months**

**Release 3:** Follow up the maintenance department with the warehouses and send smart notifications about the contents  
approximate time: **2 months**

Each release will be tested after completion, improved and modified based on user feedback to ensure that the final result achieve Services-Inc's needs, helps in resolving issues it was facing and becomes able to satisfy customers through the services it offers and at the end of each release the management's ability to oversee the overall state of the company increases.



## Phase 2:

### User Requirements:

The system shall allow users to submit maintenance requests via a central platform, providing service details such as the issue type, appointment availability, and extra notes. Customers shall also have the ability to view and track the status of their requests in real-time, promoting transparency and efficiency in the service process.

1: Customers should be able to easily submit maintenance requests via the mobile application by entering the details of the problem, the type of service required and if it is available or not and specifying preferred times.

2: Customers should be able to select one or more services.

3: Customers should confirm the service request details.

4: Customers should be able to track the request's status (under review, appointment being scheduled, on the way, completed).

5: Customers should receive instant notifications once the request is placed (request receipt confirmation).

6: Customers should be able to view details of previous request (service type, date, cost, and final status) and future scheduled appointments via their profile on the App.

7: Customers should be able to evaluate the service provided and provide their feedback.

8: Customers should be able to pay directly through the application using a credit card, bank transfer, or electronic wallet, and a digital invoice will be issued immediately upon completion of payment.

9: Customers shall be able to create an account by entering the required information or log into an existing account.

10: Customers should have a general and fast communication method like an app chat, phone and email ensuring easy access to timely help

## System Requirements:

1. Customers should be able to easily submit maintenance requests via the mobile application by entering the details of the problem, the type of service required and if it is available or not and specifying preferred times.

1.1 The user should be able to submit a new maintenance request easily through the application.

1.2 The times available by the company should be displayed for the user to choose the appropriate time.

1.3 The user can specify the required service, the type of problem, and the preferred times according to the available times.

1.4 The request should reach the technical team to process it immediately after it is submitted.

1.5 The servers should respond to the user's request within quickly.

1.6 The servers should withstand the pressure of requests without affecting their performance.

1.7 The system allows the customer to modify the service request before confirming it, including adding or modifying information or deleting the service.

1.8 The system shall display the price of the service requested by the user if it is not included in the regular bill.

2. Customers should be able to select one or more services.

2.1 The system should allow the customer to select one or more services within the same service request.

2.2 The system shall provide details for each service, including the type of problem, the type of service, and the specific time.

2.3 The system should allow the customer to follow up on the status of the order in full, including all the services he provided in a single order.

2.4 The system shall request confirmation from the customer after selecting the services and entering the information, in order to confirm the request and send it to the competent authority.

3. Customers should confirm the service request details.

3.1 After the service is provided, the order information shall be fully displayed to the customer, including all details related to the services, times, and the problem.

3.2 The system should provide an option for the customer to confirm or cancel the service request.

3.3 Customers can amend incorrect details before confirming the service order.

3.4 The system shall record the date and time of order confirmation as part of the order history in the system.

3.5 The system should display a notification confirming the success of the order confirmation process after it is completed.

4. Customers should be able to track the request's status (under review, appointment being scheduled, on the way, completed).

4.1 The system should ensure that the order status is updated instantly when any changes occur, so that the customer can follow the updates moment by moment.

4.2 The system should send instant notifications to customers when there are any updates in the order status.

4.3 The system should clearly display the current status of the service request to the customer: under review, in transit, completed.

4.4 The maintenance team should be able to update the order status in the system based on the actual order submission.

5. Customers should receive instant notifications once the request is placed (request receipt confirmation).

5.1 The system should send instant notifications to the customer upon successful order placement, displayed in the app.

5.2 Notifications should contain details of the request, such as: successful submission of the request, tracking of the request status, expected response time, type of service, etc.

5.3 Through the application, customers should be able to customize the types of notifications they wish to receive, such as estimated response time, expected arrival time, and updates about additional services.

5.4 Through the application, customers should be able to select their preferred method for receiving important notifications, including email or SMS.

5.5 Messages should be in clear language that is understandable to all users.

6. Customers should be able to view details of previous request (service type, date, cost, and final status) and future scheduled appointments via their profile on the App.

6.1: The system shall provide customers with access to a dedicated section within their profile to view details of all previous maintenance requests, including service type, date, cost, and final status

6.2: The system shall allow customers to view all upcoming appointments and their associated details within their profile.

6.3: The system shall ensure that any changes to the status of a service request are reflected in real-time within the customer's request history and upcoming appointments sections.

#### 7. Customers should be able to evaluate the service provided and provide their feedback.

7.1 Upon completion of the maintenance process, the system should allow the customer to provide his written feedback and suggestions to improve the service via the application.

7.2 The system should allow the customer to rate the service and response speed using the numerical rating scale.

7.3 The system should allow management to analyze evaluations and feedback with the aim of improving the quality of service, and monitoring the overall situation.

#### 8. Customers should be able to pay directly through the application using a credit card, bank transfer, or electronic wallet, and a digital invoice will be issued immediately upon completion of payment within less than 1 minute.

8.1 The system should support three payment methods: cash on delivery, credit card, or checks, and you can choose one payment method with each maintenance request.

8.2 Integrate a secure payment system that supports credit cards, bank transfers, and e-wallets.

8.3 The system should allow the creation of a digital invoice and send it to the user of successfully completing the payment process.

8.4 The system should allow invoice records to be securely stored in user accounts to access the previous invoice at any time.

8.5 The system shall allow users to view a list of their bills, select a specific bill, and complete the payment process through the application.

#### 9. Customers shall be able to create an account by entering the required information or log into an existing account.

9.1 The system shall allow users to log in using their username and password.

9.2 The system shall provide the ability to create a new account by entering the required basic information such as name, phone number, email, password, and detailed address (GPS must be provided to display the address accurately).

9.3 The system shall verify the strength of the password and the uniqueness of the username when creating a new account.

9.4 The system shall send a confirmation code to the user via phone number or email to complete the registration process.

9.5 The system shall give the user the option to reset the password when needed, by sending a verification code to the phone number or email.

9.6 The system shall ensure that there is a waiting period of at least one minute before requesting a new verification code.

10. Customers should have a general and fast communication method like an app chat, phone and email ensuring easy access to timely help

10.1 Providing a chat feature on the application to communicate with the support team while keeping messages for reference at any time.

10.2 Providing a phone number on the application for urgent cases.

## Effort and Cost Estimation: -

### ❖ Assign Function Points

Function points are typically assigned on a scale from 0 to 5, with 0 being no complexity and 5 being very high complexity.

- **UR.1:** 5 function points.
- **UR.2:** 4 function points.
- **UR.3:** 3 function points.
- **UR.4:** 4 function points.
- **UR.5:** 2 function points.
- **UR.6:** 2 function point.
- **UR.7:** 2 function points.
- **UR.8:** 4 function points.
- **UR.9:** 3 function points.
- **UR.10:** 3 function points.

### ❖ Calculate Total Function Points

- **Total function points = 5+4+3+4+2+2+2+4+3+3 = 32 function points**

### ❖ Convert Function Points to Effort

The conversion factor is **20 hours** per function points:

**Total effort = Total function points \* Conversion factor = 32 function points \* 20 hours/function point = 640 hours**

**So, the estimated effort required to develop the web application is 680 hours (Add 40 hours for preparation and meetings)**

❖ **Adjust for Complexity Factors**

User Requirement	Function point	Total Function points (*20)	Number of Days
UR1	5	100	17
UR2	4	80	13
UR3	3	60	10
UR4	4	80	14
UR5	2	40	7
UR6	2	40	7
UR7	2	40	7
UR8	4	80	13
UR9	3	60	10
UR10	3	60	10
Total	33	640 +40=680	108 +7=115

- ✓ We will need **4 months** for one developer, **2 months** for two developers.
- ✓ The two developers will work 100%:
  - **Developer one salary: 7000 ILS.**
  - **Developer two salary: 8000 ILS.**
- 7000 +8000= 15000 is the cost of 2 developers/month.**
- ✓ Which means **15000 \* 2 = 30000 for two months**
- ✓ The **min offer is 37800 ILS** and **max offer is 43000.**

## Phase 3:

### Task 3.1:

#### Scenario for tracking customer's order status (Ansam Hamayel)

##### ➤ **Initial Assumption:**

Ansam is a regular customer in ( Reliable Services (RS) Company) and she has previously submitted an order via the mobile application. This order has been successfully logged into the system with a unique order ID and contains details like the customer's name, contact details, service description, and submission date. The system is intended to offer real-time updates on the order's status, displaying statuses like "Under Review," "In Progress," or "Completed". Ansam has accessed her account and intends to use the "Track Order Status" feature to check on the progress of her order.

##### ➤ **Normal Flow:**

After logging in, Ansam goes to the Orders section and selects "Track Order Status". Then she gets a list of all previously submitted orders, organized into "Active" and "Completed". After that, she chooses the order that she wants to track to see its details and the system obtains and presents the relevant information, such as the order's current status (for instance, "Under Review," "In Progress," or "Completed"), the name of the assigned technician (if necessary), the estimated completion time, and any recent updates. Then the customer reviews the information and closes the app or moves to another section.

##### ➤ **Alternative Flow:**

The customer Ansam remembers that she wants to modify her order at some point, so she selects the order and chooses the "Edit Order" option. Then a notification appears to ask her to confirm the Edit Order choice:

“Are you sure that you want to confirm the Edit Order choice?”

Ansam confirms the completion of the process, and the order relevant details are updated such as the appointment time or service description. After that, the changes are validated by the system, which then shows a confirmation message after saving the updates.

Ansam decides to cancel her order so she selects the "Cancel Order" option. Then a notification appears to ask her to confirm the Edit Order choice:



“Are you sure from deleting this order?”

Ansam confirms the completion of the process, and the system updates the status to "Canceled" and archives the order in the system.

### ➤ **Error Flow:**

The customer Ansam has no active orders and for this the system shows a message: "You currently have no active orders to track". After that, the system redirects Ansam to the "Submit Request" section to create a new order if desired.

Another aspect to consider is that occasionally the system cannot access order status because of server downtime or database errors, the customer Ansam is shown an error message: “Unable to fetch order details at this time.

Kindly attempt once more at a later time”. At the same time, the system records the problem and notifies the administrator for resolution, ensuring prompt resolution.

### **Other Activities:**

Email notifications are sent to Ansam for significant order updates (For instance "Order Completed").

### **System State on Completion:**

Ansam is logged on. The system effectively shows the order status and related information. When Ansam chooses a particular order, the system confirms that the details are accessed, and any updates are recorded into the database, an order is added to the system log documenting the start and end times of the session, the customer's activity (such as viewing or changing an order) and the steps taken to maintain traceability.

## Scenario for Submitting a Maintenance Request (Mona Atta)

### ➤ Initial Assumption:

Mona is a regular user of Reliable Services (RS) Company, who has already registered an account, with her personal details (such as name, contact information, and address) safely stored in the system. One day, she noticed an issue with her internet connection. She decided to use the RS app on her smartphone to submit a service request.

### ➤ Normal Flow:

After logging in, Mona navigated to the "IT Support Requests" section from the main menu. The system displayed a categorized list of common internet and IT issues, such as "No Internet Connection," "Slow Internet," "Router Not Working," and "Frequent Disconnections.". Each category was accompanied by clear icons and images to make the selection process easier. Mona selected "Slow Internet" since her connection was performing below normal speeds. If her issue hadn't been listed, she could have selected "Other" and manually described the problem.

Next, Mona was prompted to provide additional details. She selected "Home Office" as the location of the issue and described the problem: "The internet speed is much slower than usual, making it difficult to stream videos or attend online meetings.". She was also able to attach photos or videos to better illustrate the issue, so Mona uploaded an image of her speed test results to illustrate the issue.

After entering the details, the system presented a calendar of available maintenance times. Mona chose "Tomorrow, December 24th, 2:00 PM," but the system informed her that the time was unavailable. The system then suggested alternative times, and Mona selected 3:00 PM, which was more convenient for her.

The system displayed a summary of Mona's request, including the selected category ("IT Support Requests"), the issue ("Slow Internet"), the location ("Home Office"), the description("The internet speed is much slower than usual, making it difficult to stream videos or attend online meetings."), and the selected maintenance time("December 24th, 3:00 PM"). Mona reviewed the details and clicked "Submit Request."

Upon submission, the request was saved in the database with a unique tracking number. Mona received a confirmation message on her screen and by email, which included the tracking number, full request details, and a link to track the request status. The system then forwarded her request to the appropriate maintenance team for follow-up.

### ➤ Alternative Flow:

After submitting her request, Mona was confident that everything was in order. However, in case her issue was not listed, she knew the process for handling such situations.

In the future, if Mona encounters a problem that is not listed under the predefined options, she can easily select the "Other" option. The system will prompt her to manually describe the issue in a text field. Additionally, Mona can provide any further details in optional fields to ensure that the maintenance team understands the specific problem.

While scheduling, Mona had initially selected 2:00 PM on December 24th, but when the system informed her that this time was unavailable, she was able to choose from several alternative options. The system notified her about the unavailability and provided times such as 3:00 PM, 4:00 PM, and 5:00 PM. Mona chose 3:00 PM, which suited her schedule better.

In case Mona needed to make changes after submitting her request, she could go to the "My Requests" section of the app. Here, she could update the issue description, change the location, or modify the time if necessary. For example, if she realized she needed to reschedule, she could easily select a new time that worked better for her. If she changed her mind entirely and decided not to go forward with the request, she could select the "Cancel Request" option. The system would redirect her back to the homepage, and the request would not be saved in the system.

#### ➤ **Error Flow:**

During the submission process, Mona experienced an unexpected internet connection loss. The system immediately displayed a message indicating the failure and reassured her that the entered data had been saved temporarily. As soon as the connection was restored, the system automatically attempted to submit the request again. Mona didn't have to redo the entire process, and the system managed the retry smoothly, completing the submission once the connection was stable.

#### ➤ **Error Flow:**

When Mona attempted to submit her request, the system detected that she had missed filling in one of the mandatory fields. A prompt appeared on the screen, clearly highlighting the missing field and requesting Mona to complete it before proceeding. Mona quickly filled in the necessary information and was able to continue with her request submission without further issues.

#### ➤ **Error Flow:**

While uploading a photo of her speed test results to illustrate the issue, Mona encountered an error with the file upload. The system displayed a clear error message indicating the failure. Mona was given the option to retry uploading the image. If the issue persisted, she could also choose to proceed with the request without any attachments, ensuring that her request was still submitted on time.

### ➤ **System State on Completion:**

After Mona submitted her maintenance request, the system recorded it in the database with a unique tracking number. She received a confirmation message with the request details and tracking number. The request was assigned to the appropriate maintenance team, and the status was updated in real-time as the team worked on it. Mona could track the progress and receive notifications for any updates, ensuring she was informed throughout the process.

## Scenario for Payment via app (Yara Hamad)

### ➤ Initial Assumption:

The customer (Yara) has created an account in the Reliable Services (RS) application and successfully logged in. She has submitted a maintenance request that requires payment. The system has generated a list of pending orders associated with her account, and Yara is accessing the payment section to proceed with completing the transaction.

### ➤ Normal Flow:

Yara goes to the Orders section and selects "Pending Payments". Yara checks the list of pending requests and selects the request she wants to pay for. She then selects the credit card as the payment option and enters the necessary card information such as the card number, expiration date and CVV code.

The system encrypts and securely encrypts and transmits Yara's payment information to the bank for confirmation. After the payment is processed successfully, Yara gets a confirmation text sent to her registered phone number, notifying her of the completed payment. A digital invoice is automatically created and sent via email to Yara, including information about the service, the payment amount, and the payment method. The status of the maintenance request is updated by the system to "Paid."

### ➤ Alternative Flow:

Yara selects the "Pay Cash" option. A confirmation message appears:

"Do you want to confirm cash on delivery?"

Upon confirmation, the order updates to "Pending - Cash on Delivery," and the maintenance team is notified. Yara receives an email confirmation. Payment is collected upon service completion, and the status updates to "Paid." A digital invoice is sent.

### ➤ Alternative Flow:

Yara selects the "Pay via e-wallet" option. She enters the wallet number or phone. A verification code is sent to her email. Upon entering the code, the system deducts the payment and updates the status to "Paid." A confirmation invoice is sent.

### ➤ Error Flow:

Yara enters incorrect card details (number or expiration). The system notifies her and redirects to the card entry page. After correction, if the bank rejects the payment due to insufficient balance, Yara is notified and given the option to retry or cancel. If the card has sufficient funds but a processing error occurs, Yara is asked to choose another payment method.

➤ **Error Flow:**

Yara's e-wallet has insufficient balance. She is notified and directed to recharge or select another method. If wallet information is incorrect (wallet number or phone), the system notifies her and redirects to correct the data.

If a technical error occurs during payment, Yara is alerted and asked to try again or select another payment method.

➤ **System State on Completion:**

System status upon completion: Payment is processed successfully, and the maintenance request is marked as "Paid". A digital invoice is generated and added to the system. All payment details are recorded, and the customer can access the invoice at any time through the app. The system automatically updates the financial records and notifies the accounting department.

## Scenario following up on the inventory status of shortages and overages (Doaa Assi)

### ➤ **Initial Assumption:**

The inventory management system continuously monitors the inventory contents as they change, detecting both shortages and excesses. The staff in the maintenance department rely on this system to monitor the status of inventory and respond according to the information provided.

### ➤ **Normal Flow:**

The maintenance team was carefully tracking their inventory, keeping track of the names of parts, quantities and storage locations. Each time there are changes in content, whether by adding new quantities or consuming items, the system automatically updates the database after analyzing incoming requests, such as maintenance requests or purchase orders from suppliers. The team verifies the validity of these updates before they are approved, ensuring data accuracy and seamless business continuity. With this up-to-date information, the maintenance team can better plan and make accurate decisions on future orders to ensure inventory is always ready.

### ➤ **Alternative Flow:**

While the system was monitoring inventory, a shortage of the item was detected, as the quantity was below the predetermined minimum. An alert was sent via the system to the maintenance department showing the item, the current quantity, and the estimated quantity required based on the inventory consumption of that item in a given period. The system automatically suggests placing a request to return the stock to the required limit.

### ➤ **Alternative Flow:**

The maintenance department recently ordered a set of parts to add to inventory. The system thoroughly reviewed and analyzed the order, comparing the required quantities with the predetermined maximum. When the system notices that the required quantities exceed this limit, send an alert to the manager to make the appropriate decision.

### ➤ **Alternative Flow:**

The maintenance department received two simultaneous requests for the same maintenance service at the same time. When reviewing the available quantities, the system discovered that inventory was insufficient to meet all orders. Immediately, the system issued an alert warning the maintenance department of the shortage and displaying the current quantity available in stock.

➤ **Error Flow:**

During the inventory check in the warehouse, a worker in the maintenance department noticed a discrepancy between the actual quantities of inventory and the records in the system. This may be the result of manual additions or removal of items without modifying the data in the system. An alert has been sent to the maintenance department in order to review and update the inventory information to ensure its accuracy.

➤ **Error Flow:**

Two people try to change the inventory quantity for the same item and on the same order at one time, which can lead to a possible information collision. The system was able to detect this discrepancy by reviewing the last record of data, where it prevented duplicate modifications to the same item for the same request. This ensures inventory accuracy and efficient operations continuity.

➤ **Other Activities:**

Detailed inventory status reports, including shortages and surpluses, can be generated and regularly sent to the maintenance department for review, follow-up, and appropriate action.

➤ **System State on Completion:**

Once the adjustments and inventory verification have been completed, the system updates the inventory data to accurately reflect the current stock contents. In addition, comprehensive reports are generated summarizing the inventory status in terms of shortages and surpluses, and these reports are sent to the maintenance department for follow-up and decision-making.



## Scenario for feedback (Hiba Awwad)

### ➤ **Initial assumption:**

Hiba received the service and opened the company's application to evaluate his experience with the service provided, which reflects his satisfaction, and the system is ready to receive feedback and includes a dedicated form to collect customer opinions.

### ➤ **Normal:**

Hiba provides a stars rating. The system processes her feedback and displays a thank you message.

### ➤ **Alternative:**

The star rating system doesn't work (due to a technical issue), Shows to Hiba a text box to manually write a comment or voice feedback. The system processes her feedback and displays a thank you message.

### ➤ **Error:**

When submitting feedback, there is a server outage.

The system automatically takes action when it detects that the server isn't functioning by informing.

Hiba and the technical support team regarding the problem. Hiba received this message: "Sorry, there is a problem. We will attempt to resolve it as quickly as possible."

Once the server was operational again, the system subsequently sent out another email containing the feedback link.so that when the system is stable, Hiba can provide her feedback.

### ➤ **Other activities:**

1. Feedback Modification Options: Hiba can use different options to update or change her feedback. Hiba may use the system to modify their ratings, add further remarks, or offer clarifications even after first submission, guaranteeing a more flexible and iterative process for feedback

2. Feedback Prioritization and Categorization: By putting in place an automated system, feedback is ranked according to its type. The system can detect important

problems using this alternate technique and produce support requests that are prioritized.

➤ **System state on completion:**

1) If applicable, the customer support team is notified to resolve certain concerns brought up by the feedback, and the feedback is safely stored in the customer feedback database.

2)Hiba got a thank-you note, which improved their rapport with the business.

### Task 3.2:

- Actors:

- 1) Customer: The customer interacts with the system by sending maintenance requests through the app, tracks the status of their requests, provides feedback or ratings for finished services, and processes payments using the e-payment system for requested services.
- 2) Maintenance Team: The maintenance team interacts with the system by reviewing and carrying out assigned maintenance requests, accessing inventory details, updating the system on the progress and status of each task, and communicating with the customer if needed regarding the service.
- 3) HR Management: Manages the recruitment process of maintenance team members, ensures that employees have the necessary skills and certifications for assigned responsibilities, and follows up on financial reports.
- 4) Email System: to sends automated notifications to customers and the maintenance team regarding service updates, request confirmations, and payment receipts. Furthermore, notifies the manager regarding escalated problems or system downtime.
- 5) E-payment System: Processing and confirming payment methods from customers for services.
- 6) Accounting System: Tracks all financial transactions, including service charges, costs, and employee salaries, produces financial reports for management, assists in preparing invoices and maintaining documentation for compliance and audit reasons.
- 7) GPS: to allocate the customer's address.
- 8) Technical Support Team: Provides support to customers and maintenance team members regarding technical problems with the application and addresses complaints or bugs identified in the system.
- 9) Manager: Manage overall operations and ensure quality of service, allow escalation or exceptions in requests, and follow up on financial reports and company status.

## ○ Use Case Diagram

*Reliable Services Diagram*



## Task 3.3:

### Use Case: Track order status (Ansam Hamayel)

#### 1.1 Brief Description:

This use case allows the customer to track the status of their order in the system. The system provides updates on the service's status and estimated delivery details.

#### 1.2 Actors:

1.2.1 Customer.

#### 2. Preconditions (Entry condition):

2.1 Log In: Before this use case begins, the customer needs to possess an active account and be signed into the system.

2.2 Placing order: The service must already exist in the system and associated with the customer's account.

#### 3. Flow of Events:

3.1 Basic Flow - Track Order Status:

3.1.1 The customer logs into the system by entering their credentials.

3.1.2 The customer goes to the "Track Order Status" section within the app.

3.1.3 The customer enters the order id and selects "submit".

3.1.4 The customer can refresh the status view to get the latest update manually.

3.1.5 When the customer is finished tracking the order, the use case ends.

#### 4. Alternative Flows:

4.1 Order Not Found:

If the system is unable to locate the order with the specified id, it shows an error message, "Order Not Found". The User may either input the order id again or terminate the operation, at which point the use case ends.

4.2 Order Postponed:

If the system identifies that the order is late beyond the expected delivery time, it alerts the User with a notification message. The User has the option to either wait for additional updates or reach out to the Laundry worker.

#### 4.3 System Downtime:

If the system is temporarily unavailable, the customer gets a message saying, "Order status tracking is currently unavailable, Kindly attempt again at a later time."

### **5. Post-conditions (Exit condition):**

5.1 The customer is able to see the current status of their order on the review page.

5.2 The System confirms the order and sends a notification to the delivery system to pick up the order from the customer.

### **6. Special Requirements:**

6.1 Multi-Language Support: The system should provide support in various languages to cater to diverse users.

6.2 The system should Verifying all customer financial data

### **7. Extension Points:**

7.1 Integration with Delivery Personnel System: The system obtains live updates and contact information from the tracking system of the delivery personnel.

7.2 Integration with Notification System:

The system sends notifications through email or app alerts to update the customer regarding changes in order status.

## Use Case: Submit a Maintenance Request (Mona Atta)

### 1.1 Brief Description:

This scenario allows the user to submit a maintenance request easily by selecting the required service, describing the issue, choosing a suitable maintenance date, and submitting the request to the technical team for follow-up.

### 1.2 Actors:

- 1.2.1 User(customer)
- 1.2.2 System
- 1.2.3 Maintenance Team

### 2. Preconditions (Entry condition):

- 2.1 The user has a registered account and has logged in successfully.
- 2.2 The required service is available in the service list.
- 2.3 The problem description is entered accurately.

### 3. Flow of Events:

- 3.1 Basic Flow -Request Submission
  - 3.1.1 The user logs into the system via the website or mobile app using their registered credentials (username and password).
  - 3.1.2 The user navigates to the "IT Support Requests" section and selects the required service category from the list (e.g., No Internet Connection, Slow Internet, Router Not Working, and Frequent Disconnections.).
  - 3.1.3 Specifies the location (such as home office or company ) and access details if necessary.
  - 3.1.4 Selects a suitable date for maintenance.
  - 3.1.5 Reviews and confirms the request.
  - 3.1.6 The submitted request is saved with a tracking number for the user.

### 4. Alternative Flows:

- 4.1 Problem Not Found in Options:
  - User selects "Other" option and enters problem description, they can manually enter a description.
- 4.2 Appointment not available:
  - The system displays alternative available time slots for the user to choose from.
  - The user can then select an alternative appointment time that fits their schedule.

#### 4.3 Cancel order:

User can cancel order before confirming and the system will redirect them back to the main page without saving the request, effectively discarding the order.

### **5. Post-conditions (Exit condition):**

5.1 The order is registered in the system with a unique tracking number.

5.2 A confirmation message is sent to the user containing the order details.

5.3 The request is transferred to the technical team for processing.

### **6. Special Requirements:**

6.1 The system interface should be easy to use.

6.2 The system should be always online to ensure that requests are sent and responses are received.

6.3 The mobile app should be available for both Android and iOS users.

6.4 The system should support at least two languages (such as English and Arabic).

6.5 The system should be accessible for users with disabilities, with support for screen readers and large text options.

### **7. Extension Points:**

7.1 The user can track the status of the order later through the control panel.

7.2 Adding a feature that allows users to communicate with technical support staff via video calls or text conversations to obtain immediate assistance without the need to visit the site.



## Use Case: Payment via app (Yara Hamad)

### 1.1 Brief Description:

This use case allows the customer to pay the order or invoice amount through several payment methods. It allows the user to choose immediate or deferred payment.

### 1.2 Actors:

- 1.2.2 Customer.
- 1.2.3 E-Payment.
- 1.2.4 System
- 1.2.5 Maintenance team

### 2. Preconditions (Entry condition):

- 2.1 Log In: Before this use case begins, the customer needs to possess an active account and be signed into the system.
- 2.2 Submit Request: The customer must have submitted a maintenance request before starting the payment process.

### 3. Event Flow:

#### 3.1 Basic Flow - Payment via the App:

- 3.1.1 The customer logs in to the system by entering his credentials.
- 3.1.2 The customer navigates to the "Requests" section within the app.
- 3.1.3 The customer navigates to the "Pending Payment Requests" section within the Orders section.
- 3.1.4 The customer selects the order he wants to pay.
- 3.1.5 The user selects the payment method from the options available in the application.
- 3.1.6 When the user completes the payment process, the use case ends.

### 4. Alternative flows:

#### 4.1 Payment by e-wallet:

The user chooses to pay via the e-wallet and enters all the required information to pay the due amount and issue a digital invoice and change the order status to "Paid"

#### 4.2 Payment by cash on delivery:

The user chooses to pay by cash on delivery "cash" and the order remains in the "pending payment" status until the amount is received from the employee and a digital invoice is issued and the order status is changed to "Paid"

## **5. Post-order conditions (exit condition):**

5.1 The customer can see the order status as paid.

5.2 The customer ensures that a digital invoice is issued with the correct details.

## **6. Special Requirements:**

6.1 Support for encryption and confidentiality of information and maintaining its security when using a credit card or electronic wallet for payment

6.2 Provide an easy-to-use payment interface with clear confirmation messages during the process and upon completion

6.3 Guide the customer to deal appropriately with problems he faces during the payment process

## **7. Extension points:**

7.1 Integration with the notification system:

The system sends automatic notifications via email or application notifications to confirm the success of the payment process or in the event of its failure.

7.2 Integration with the bank account management system:

The system is automatically updated with the payment status from the bank or the entity responsible for the transaction, with immediate verification of the validity of the banking information or the customer's balance.

7.3 Integration with the billing system:

Once the payment process is completed, the invoice details are sent electronically to the billing system and the account records are automatically updated.

7.4 Integration with the technical support system:

In the event of any problem during payment, the customer can send an immediate report to the technical support team, and the report is automatically recorded and followed up.

7.5 Integration with Loyalty and Rewards System:

If there are rewards or loyalty points based on payments, the points are automatically added to the customer's account after successful payment.

Use Case: follow up on the inventory status of shortages and overages  
(Doaa Assi)

### **1.1 Brief Description:**

The system allows maintenance teams to accurately track inventory quantities, and send notifications if quantities fall below the agreed minimum or exceed the maximum. This system contributes to maintaining stock balance, which reduces the likelihood of unwanted shortages or increases.

### **1.2 Actors:**

1.2.1 Maintenance staff.

### **2. Preconditions (Entry condition):**

- 2.1 The presence of accurate and up-to-date information on the contents of the inventory in the system.
- 2.2 Determine the lowest and highest level of inventory.

### **3. Event Flow:**

- 3.1 The system monitors the inventory contents automatically, in addition to the inventory follow-up by the maintenance department staff.
- 3.2 The system analyzes maintenance requests automatically.
- 3.3 The system checks the availability of the required parts, and if the quantity is sufficient, the required parts are reserved and linked to the maintenance request without modifying the quantities in the database.
- 3.4 The maintenance employee confirms the consumption of the reserved parts after the maintenance work is completed.
- 3.5 The inventory database is automatically modified based on the quantity consumed.
- 3.6 The system reviews the available quantities after the update.

### **4. Alternative flows:**

- 4.1 If the system detects a shortage of an item below the specified minimum, it sends an alert to employees about the problem, suggesting the required order quantity based on inventory consumption.
- 4.2 When a request is made to replenish stock, the system analyzes the request. If the requested quantity exceeds the pre-defined maximum, the system issues an alert to the manager to re-evaluate the request and avoid unnecessary increase.
- 4.3 If the system receives two simultaneous maintenance requests for the same item and the inventory is insufficient, the system issues an alert and displays the available quantity for appropriate action.

### **5. Post-order conditions (exit condition):**

- 5.1 Record the current inventory accurately and reflect the actual situation in the database.
- 5.2 Send appropriate alerts based on the situation.
- 5.3 Generate comprehensive reports on the inventory status and send them to the maintenance department for necessary action.

## **6. Special Requirements:**

- 6.1 The system should support sending alerts via approved email and immediate alerts for shortages or surpluses within 3 seconds of detecting the condition through the system.
- 6.2 The system should support receiving simultaneous requests.
- 6.3 Unauthorized modifications to inventory data should be prevented.
- 6.4 All inventory transactions should be recorded and periodic and comprehensive reports should be generated.
- 6.5 There should be a mechanism for backup and recovery in case of system failure.
- 6.6 Analysis tools should be supported to make future decisions.

## **7. Extension points:**

- 7.1 Integration with a notification system: The system sends notifications to the maintenance department to alert them of stock shortages or overages.
- 7.2 Track quantities: To ensure that actual quantities match those in the system. Technologies such as barcodes can be used to reduce the need for human intervention, with manual intervention possible when necessary.

## Use-Case: feed back (Hiba Awwad)

### 1.1 Brief Description:

The system allows the customer to submit and send his comments on the service he received, process the inputs, and confirm them with a thank you message.

### 1.2 Actors:

1.2.1 Customer

1.2.2 System

### 2. Preconditions (Entry condition)

2.1 The customer has successfully received the service.

2.2 The system is up and running and prepared to take comments, and securely storing it

2.3 Within the system, the customer can access the feedback form.

### 3. Flow of Events

3.1 Basic Flow - Providing Feedback:

3.1.1. The customer receives the service, and its status changes to "Complete."

3.1.2. The customer enters the application, and a notification appears: "The service has been completed. Do you want to rate it?" When the customer clicks on the notification, they are transferred to the comments section in the system.

3.1.3. The system displays the comments form with a star rating system and an optional text box for comments.

3.1.4. The customer sends the comment by clicking on the "Submit" button.

3.1.5. The system processes the comment and stores it in the system for future reference.

3.1.6. The system displays ("Thank you") to the customer, and the use case ends.

### 4. Alternative Flows:

4.1 Alternative Flow\_1 :

The customer writes a comment and clicks on the "Submit" button without selecting a star rating. The system detects that the rating is missing and prompts the customer to select a rating. The system displays a message: "Please select a star rating before submitting your feedback." The customer selects a star rating and clicks "Submit" again. The system processes the feedback and stores it, then displays a thank-you message.

#### 4.2 Alternative Flow\_2 :

The customer selects the star rating and clicks "Submit" without writing a comment. The system processes the feedback without the comment. The system stores the rating and displays a thank-you message to the customer.

#### 4.3 Alternative Flow\_3:

The customer clicks "Submit" to send the feedback. The system encounters an error while processing the data . The system displays an error message: "An error occurred while submitting your feedback. Please try again later." The customer is prompted to retry the submission. The customer retries or exits the feedback section.

#### 4.4 Alternative Flow\_4:

The customer decides to modify their rating or comment after submitting the feedback. The customer submits the feedback successfully. The customer realizes they want to change their rating or modify their comment. The system provides an option to "Edit Feedback." The customer modifies the feedback and clicks "Submit" again. The system updates the feedback and displays a thank-you message.

### **5. Post-conditions (Exit condition)**

- 5.1 The system has successfully recorded the feedback, for future reference.
- 5.2 The system has displayed a thank-you message to the customer.
- 5.3 The service status remains marked as "Complete" after the customer submits feedback, confirming that the process has finished.
- 5.4 The notification prompting the customer to rate the service is no longer displayed, as the feedback process is completed.

### **6. Special Requirements**

- 6.1 The feedback form should be designed clearly, intuitively, visually and easily to use.
- 6.2 The text displayed in the feedback form is translatable according to the user's language settings.
- 6.3 The system should automatically retry submission in case of a temporary failure, and notify the customer once the feedback has been successfully submitted.
- 6.4 The feedback process is usable by people with special needs
- 6.5 Encrypt all feedback data, to ensure data privacy.
- 6.6 The feedback form supports multiple languages.
- 6.7 The feedback system should be able to handle a large number of simultaneous submissions without a decrease in system performance.

6.8 Load the feedback form in a very short period of time, even during peak times. The system should also process and store the feedback data in a very short period of time after submission.

6.9 The feedback submission system should integrate with customer service platforms to alert support teams if the rating indicates a negative experience.

6.10 The optional comment box should allow customers to enter a very large number of characters.

## **7. Extension Points**

7.1 Handles errors during feedback submission.

7.2 Switch Language Based on User's Locale.

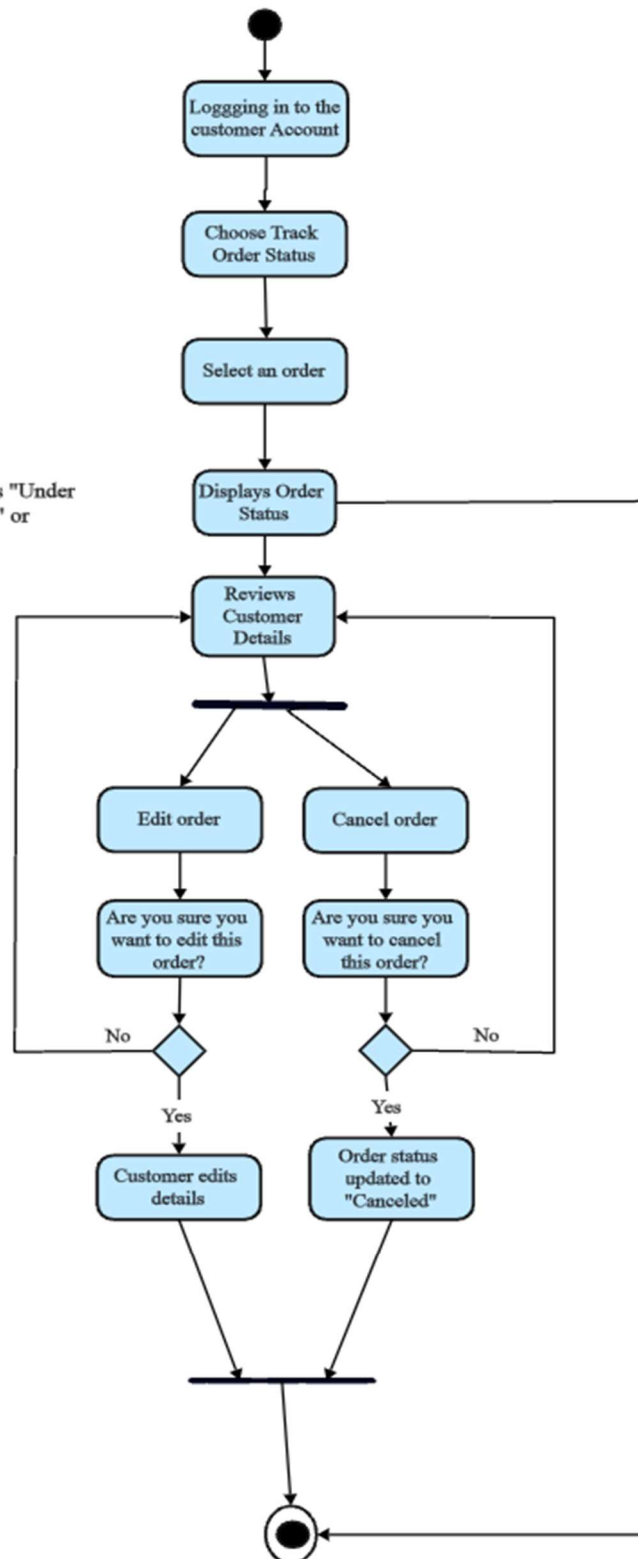
7.3 Follow-up for Negative Feedback.

7.4 Linking up with the Notification System

### Task 3.4:

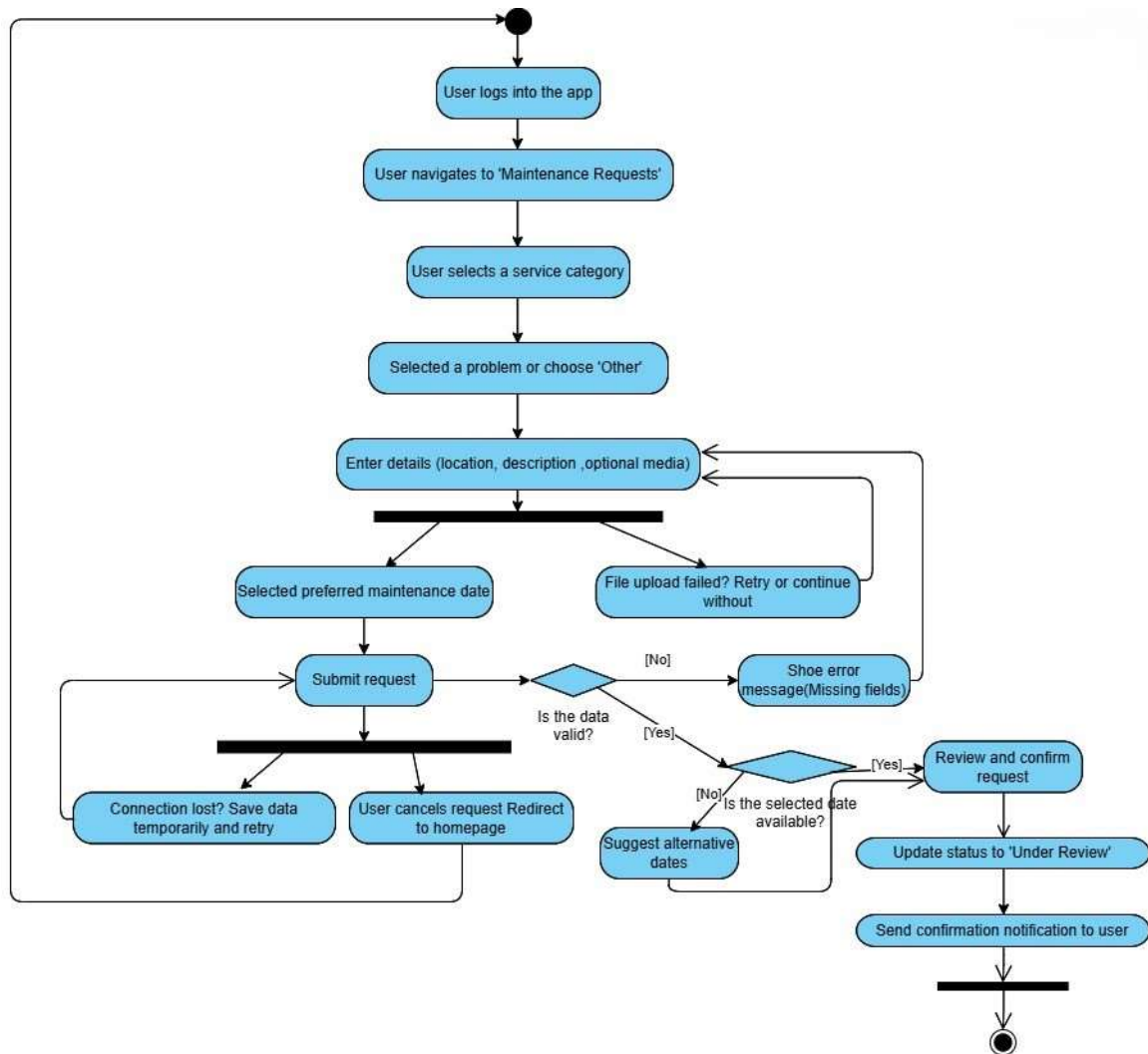
Instance Activity Diagrams for tracking customer's order status (Ansam Hamayel)

- Note: the order status is "Under Review," "In Progress," or "Completed"

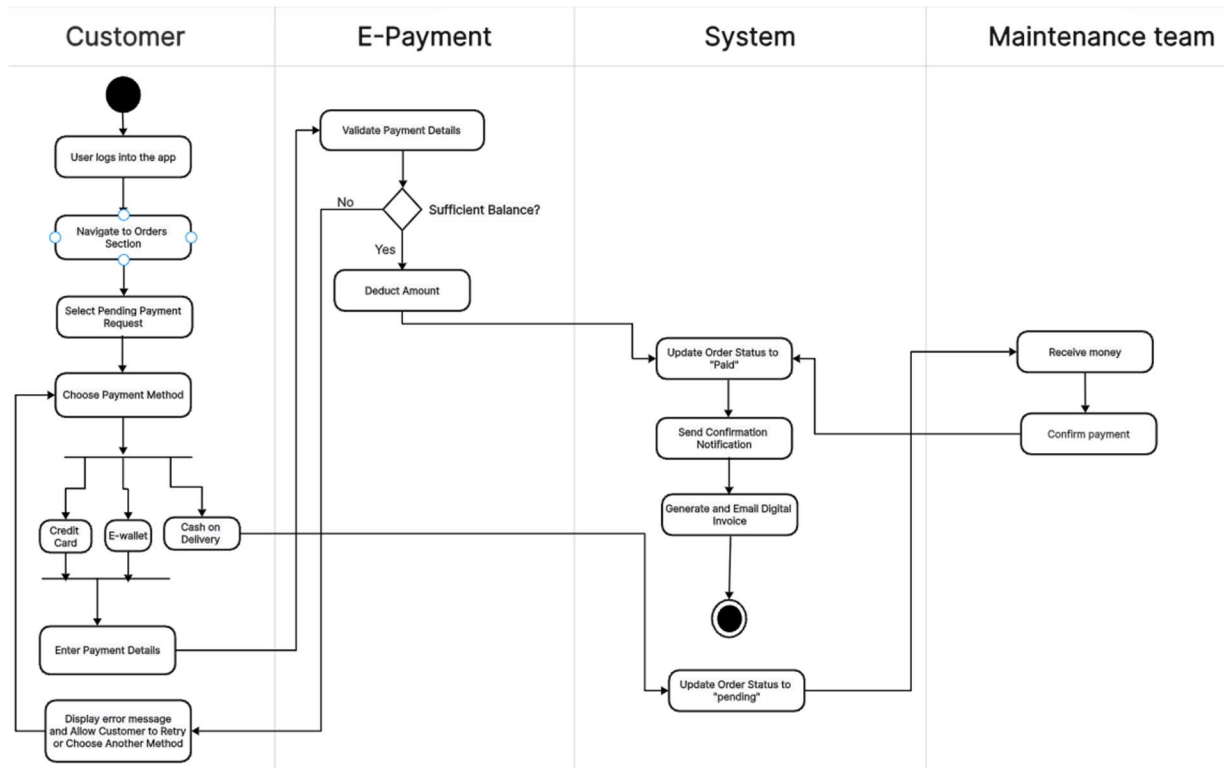




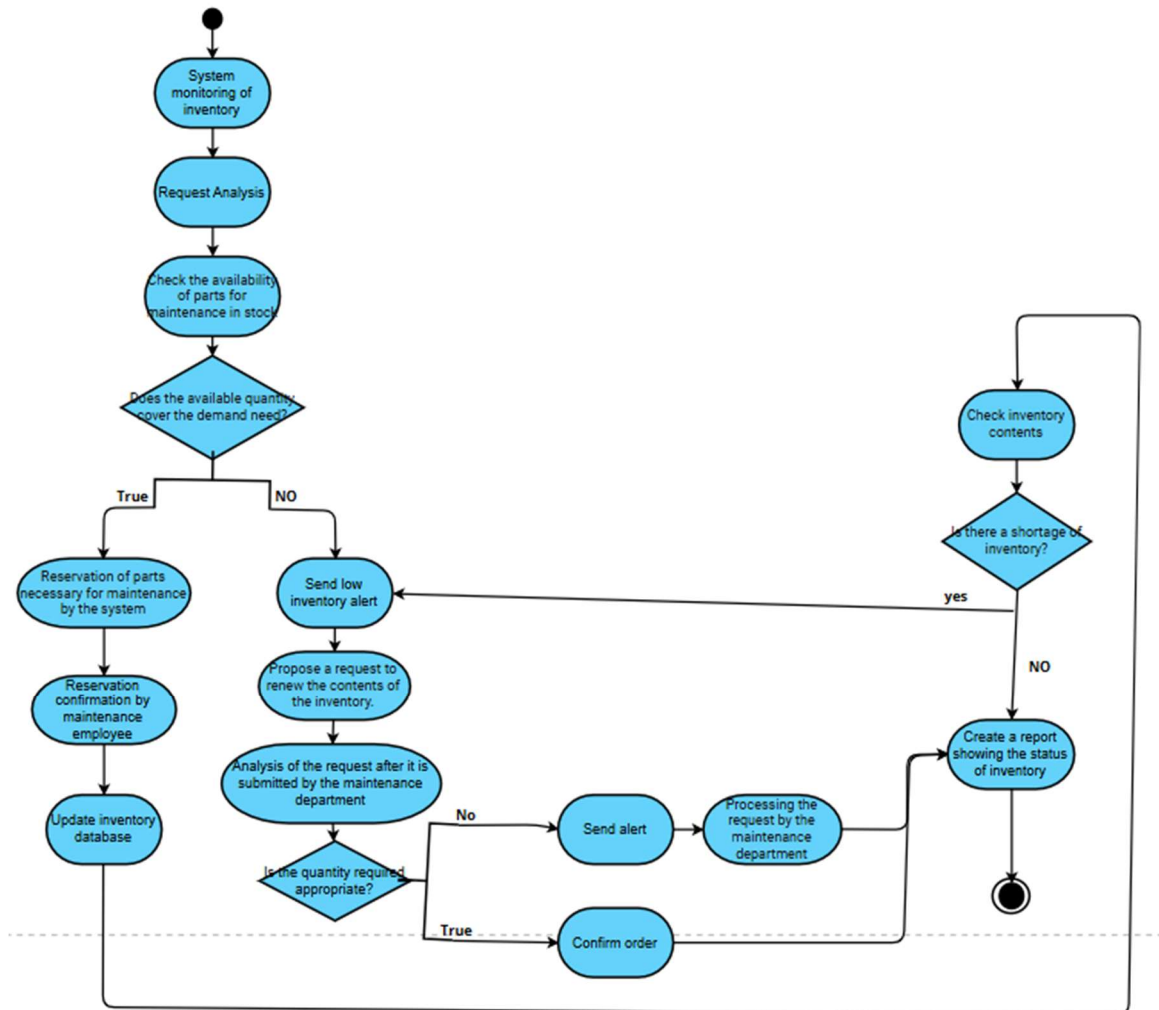
## Instance Activity Diagrams for Submitting a Maintenance Request (Mona Atta)



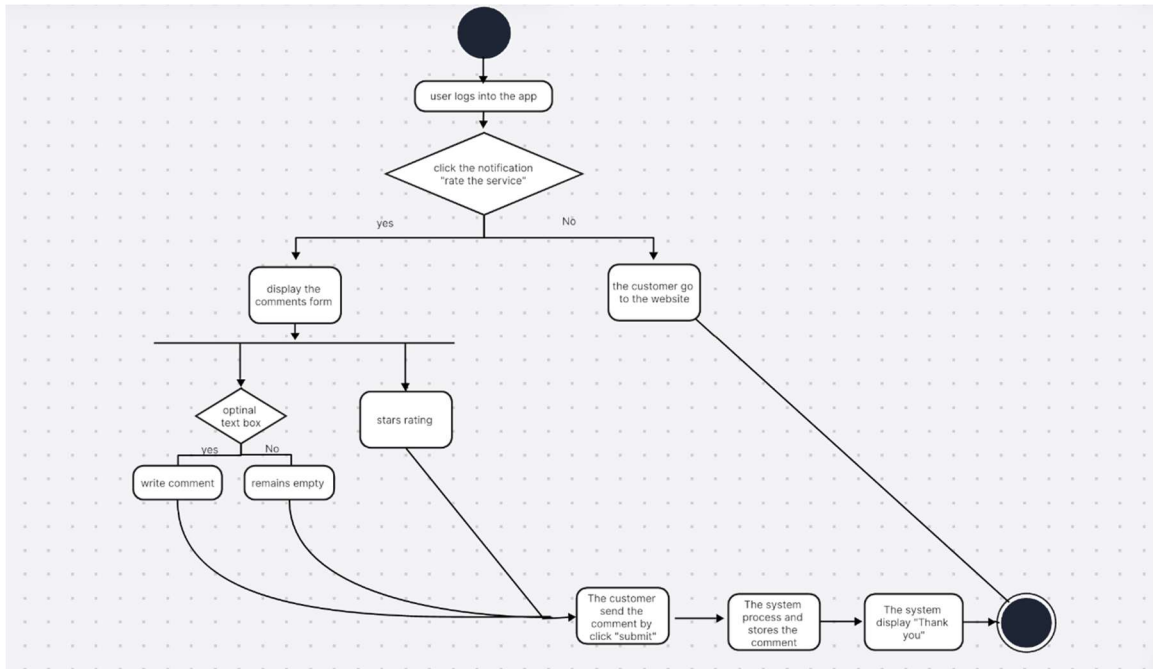
## Instance Activity Diagrams for payment via app (Yara Hamad)



## Instance Activity Diagrams for following up on the inventory status of shortages and overages (Doaa Assi)



## Instance Activity Diagrams for Feedback (Hiba Awwad):



## Phase 4:

### Task 4.1:

## System Class modelling and Analysis:

### 1. Identifying System Analysis Classes

Below is the list of identified **analysis classes** for the system, with descriptions for each:

#### 1. Person (Abstract Class)

- **Attributes:**
  - `name: string` - Represents the name of the person.
  - `phone_number: int` - Represents the contact number.
  - `email: string` - Represents the email address.
  - `username: string` - Unique identifier for the customer.
  - `password: string` - Used for authentication.
- **Methods:**
  - `login(): void` - Allows the user to log in.
- **Purpose:**
  - Parent class for `Employee` and `Customer`.

#### 2. Customer

- **Attributes:**
  - `address: string` - Represents the customer's address.
- **Methods:**
  - `trackOrderStatus(): void` - Allows the customer to track the status of their orders.
  - `submitMaintenanceRequest(): void` - Allow the customer to submit a new maintenance request.

#### 3. Employee (Abstract Class)

- **Attributes:**
  - `salary: double` - Represents the employee's salary.
  - `role: string` - Represents the role of the employee.
  - `id: int` - Unique identifier for the employee.

- **Methods:**

- `provideFeedback(): void` - Allows the employee to submit feedback.
- `makePayment(): void` - Allows the employee to process payments.

- **Purpose:**

- Parent class for various employee roles (`MaintenanceEmployee`, `Manager`, etc.).

#### 4. **MaintenanceEmployee**

- **Methods:**

- `trackOrderStatus(): void` - Tracks the status of maintenance tasks.
- `updateInventory(): void` - Updates the inventory after maintenance.

#### 5. **Manager**

- **Methods:**

- `viewReports(): void` - Allows the manager to view generated reports.

#### 6. **HR Management**

- **Methods:**

- `manageRecruitment(): void`
- `processPayroll(): void`
- `monitorPerformance() : void`

#### 7. **TechnicalSupportTeam**

- **Attributes:**

- `technicianId: int` - Unique identifier for technicians.
- `name: string` - Name of the technician.
- `availability: Boolean` - Availability status.

- **Methods:**

- `analyzeFeedback(): void` - Analyzes customer feedback.
- `resolveComplaint(): void` - Resolves system-related complaints.

#### 8. **MaintenanceRequest**

- **Attributes:**

- `requestId: int` - Unique identifier for the request.
- `status: string` - Status of the request.
- `description: string` - Detailed description of the request.

- appointment\_time: string - Time scheduled for maintenance.
- requestDate: date - Date the request was made.
- pickUpDate: date - Date for picking up completed work.
- price: double - Total cost of the maintenance.

- **Methods:**

- createRequest(): void - Creates a new maintenance request.
- updateRequest(): void - Updates an existing request.

## 9. Service

- **Attributes:**

- serviceId: int - Unique identifier for the service.
- serviceType: string - Type of service offered.
- extraCost: double - Additional cost for the service.
- estimatedDuration: double - Estimated time for completing the service.

## 10. Section

- **Attributes:**

- sectionId: int - Unique section identifier.
- sectionName: string - Name of the section.
- numOfEmployee: int - Number of employees.

## 11. Inventory

- **Attributes:**

- itemId: int - Unique item identifier.
- itemName: string - Name of the item.
- quantity: double - Available quantity.

## 12. Inventory Employee

- **Methods:**

- checkStock(): void - Verify stock levels.
- updateStock(): void - Update stock records.
- reOrderItems(): void - Reorder items.
- generateReport(): void - Generate inventory reports.

## 13. Payment (Abstract Class)

- **Attributes:**

- name: string - Name associated with the payment.
- paymentId: int - Unique identifier for the payment.

- `totalAmount: double` - Total payment amount.
- **Methods:**
  - `processPayment(): void` - Processes the payment.
  - `generateInvoice(): void` - Generates an invoice for the payment.
  - `refundPayment(): void` - Handles payment refunds.
- **Purpose:**
  - Parent class for payment methods like `Cash`, `CreditCard`, and `EWallet`.

#### 14. Cash

- **Attributes:**
  - `currencyType: string` - Type of currency.

#### 15. CreditCard

- **Attributes:**
  - `cardNumber: int` - Card number.
  - `bankName: string` - Bank name.
  - `cardHolderName: string` - Cardholder's name.
  - `endDate: date` - Expiration date.

#### 16. EWallet

- **Attributes:**
  - `walletNumber: int` - Wallet number.
  - `linkedPhoneNumber: int` - Linked phone number.
  - `walletProvider: string` - Wallet provider.

#### 17. Feedback

- **Attributes:**
  - `rating: int` - Customer rating for the service.
  - `comments: string` - Customer comments or feedback.
- **Methods:**
  - `submitFeedback(): void` - Submits customer feedback.
  - `viewFeedback(): void` - Displays submitted feedback.

#### 18. UserAccount

- **Attributes:**
  - `username: string` - Account username.
  - `password: string` - Account password.



- lastLogInTime: time - Last login time.

- **Methods:**

- login(): void - Login action.
- logout(): void - Logout action.
- resetPassword(): void - Reset password.

## 19. Notification

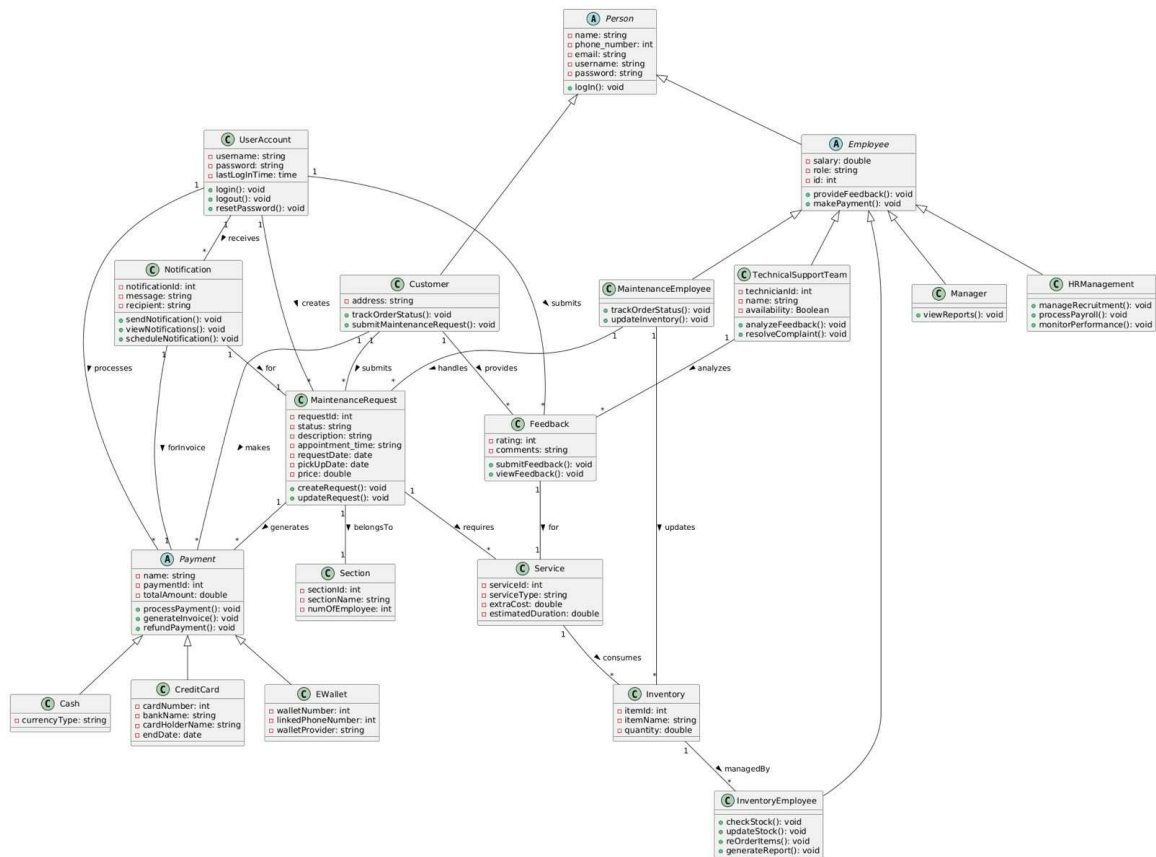
- **Attributes:**

- notificationId: int - Unique identifier for the notification.
- message: string - Notification message content.
- recipient: string - Recipient of the notification.

- **Methods:**

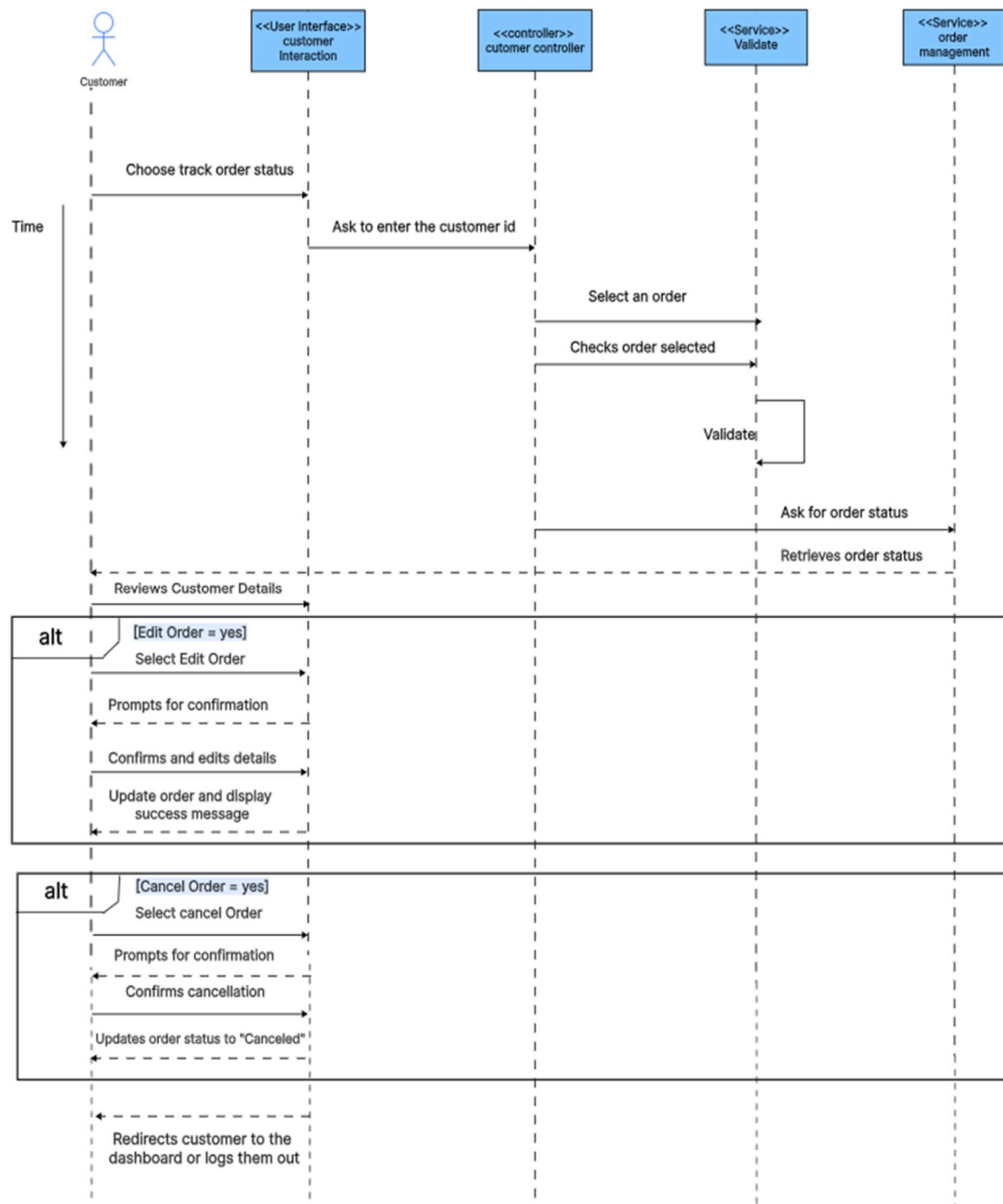
- sendNotification(): void - Sends a notification to a user.
- viewNotifications(): void - Displays all notifications for the user.
- scheduleNotification(): void - Schedules a notification for a specific time.

## 50 | Page

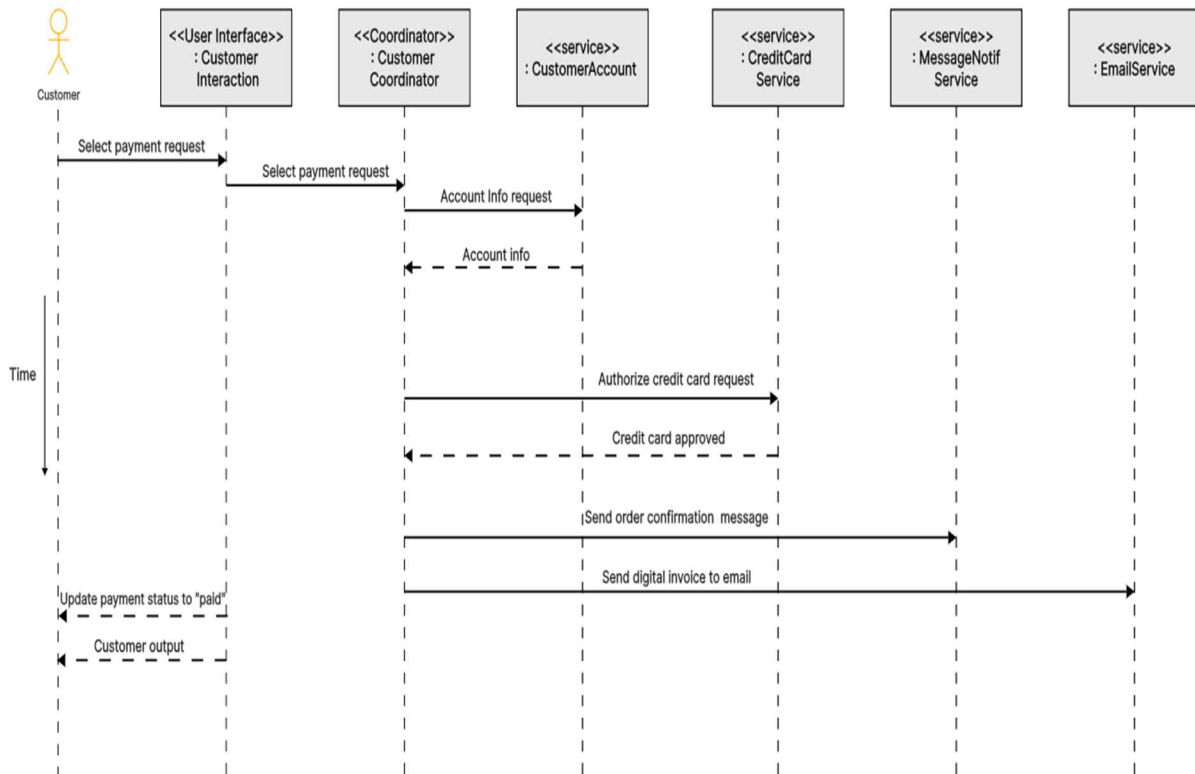


## Task 4.2:

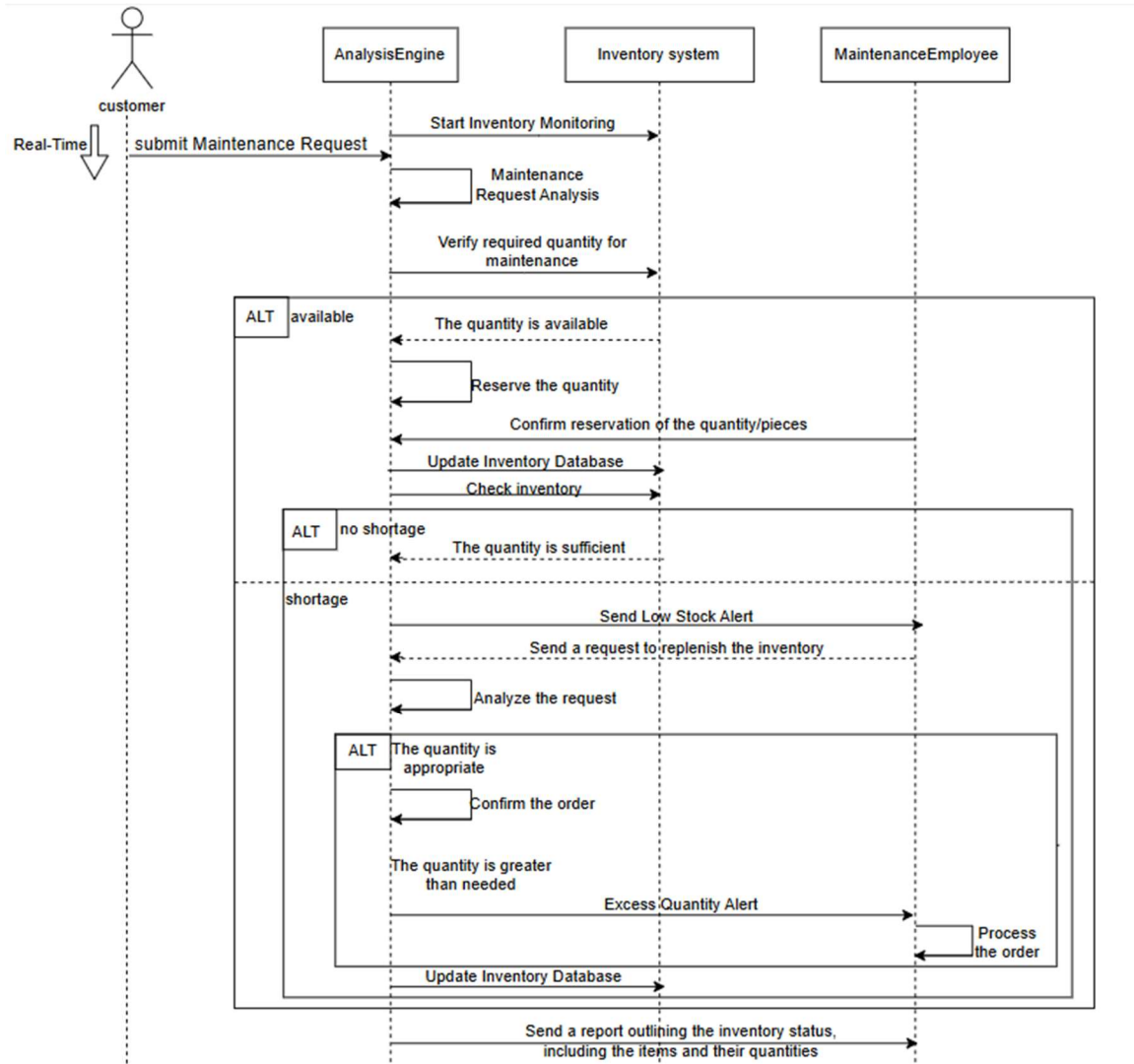
System Sequence modelling and Analysis for tracking customer's order status  
(Ansam Hamayel):



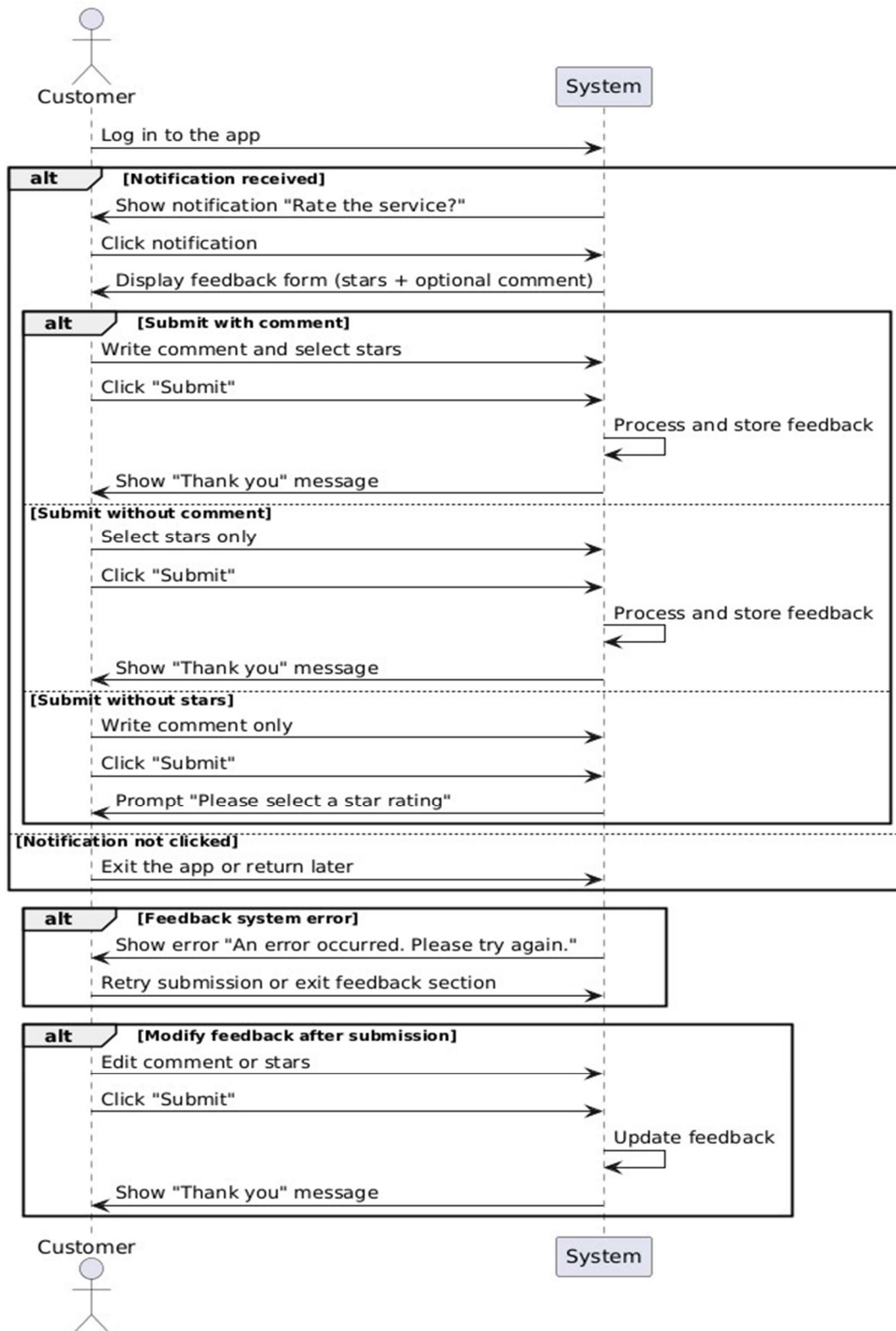
## System Sequence modelling and Analysis for pay for order (Yara Hamad)



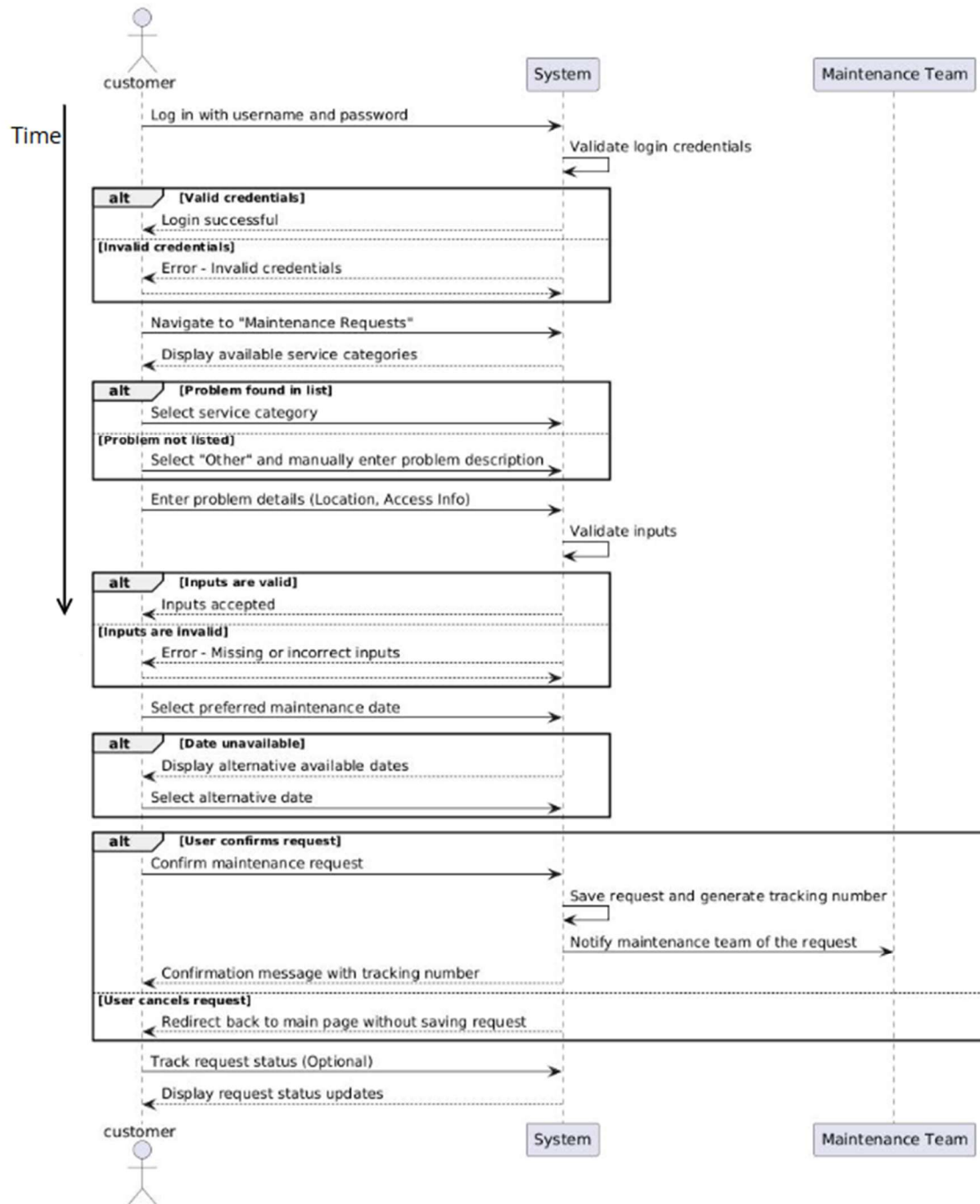
System Sequence for following up on the inventory status of shortages and overages (Doaa Assi):



## System Sequence modelling and Analysis for feedback Submission Process (Hiba Awwad)



## System Sequence modelling and Analysis for Submit a Maintenance Request (Mona Atta):



## Task 4.3:

### System Design Goals:

A set of design goals were established that focus on meeting user needs, while enhancing the system's flexibility and ensuring its long-term scalability.

➤ **Low Coupling:**

If any modification is made to one component of the system, it should not affect the other components of the system. The degree of dependency between components should be as low as possible to ensure their independence.

Example: If the inventory tracking system is modified, it should not affect the payment system. The payment system should remain working properly without any glitches, which enhances the flexibility and ease of maintenance of the system.

➤ **High Cohesion:**

Classes within a component should be interconnected and work together to achieve a specific and clear goal.

For example: In an inventory tracking component, the functionality should be limited to providing inventory tracking services only. This component should not provide other functionality such as payment processing, as this is a separate functionality that should be implemented by another independent component.

➤ **Performance:**

The system should have a high capacity to analyze maintenance requests as soon as they are received, identifying the problem accurately, efficiently and quickly. It should also ensure that the results are sent directly to the maintenance department to start implementing the request immediately, which contributes to improving the customer experience and speeding up the response process to requests.

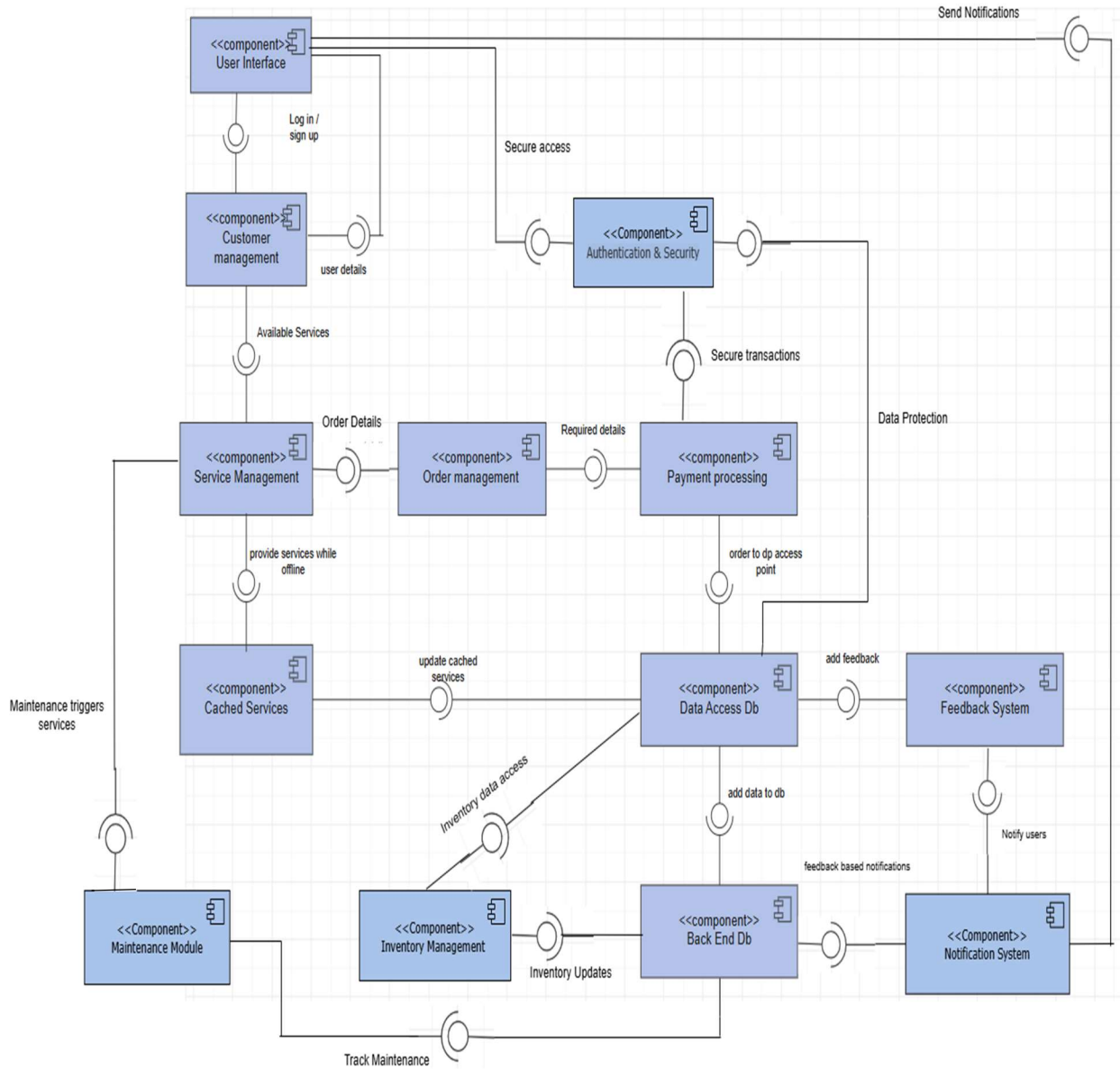
➤ **Security:**

The system must ensure comprehensive protection of data from unauthorized access through encryption and authentication techniques, with immediate response to any threats to ensure data integrity and service continuity.



## Task 4.4:

### System Component Design:



## Task 4.5:

### System Deployment Diagram:

