



Faculty of Engineering and Technology

Computer Science Department

COMP433 – Group Assignment phase 1

<<Sahab>>

Students Name	Student Number
Deema Abu Nimeh	1220002
Aseel Abd Elhaq	1221024
Amro Deek	1221642
Fadi Abu Aletham	1220316
Mustafa Alayasa	1221015

Phase 1 :

Introduction

Services-Inc is a leading company in the field of providing business and home IT services. The company provides various IT services in terms of networking, Wi-Fi, PC and laptop maintenance, printing machine maintenance, and so forth. The company's current system relies on traditional techniques such as using hard copies, online storage forms, or standalone systems. The company faces a dilemma in terms of organizing data since data is currently stored and processed in Excel sheets and Google Sheets, thus causing a great deal of data scattering and, in turn, causing duplicates in data and an ineffective way of dealing with data. The company also faces an issue with their manual processes in terms of dealing with hard copies of data, which leads to loss of data as well as making the task of accessing data incredibly time-consuming and causing errors in calculations, especially when creating statistical reports of any sort. Another concern would be managing private client information in a way that compromises the clients' privacy.

One more issue would be the lack of a proper inventory management system, which leads to wasting the customer's time as well as the technician's time. This causes a great decrease in customer satisfaction. Another issue would be difficulty in terms of contacting the company since their only way of communication is through the phone, which is inefficient and time-wasting as well as misunderstanding the customer's needs. The company lacks a strict scheduling system in terms of company vehicles as well as the number of technicians and employees on duty, which leads to time consumption, overworking employees, and underutilized resources. All of these problems make it evident why Services-Inc is in great need of an integrated software solution since the current system lowers operational efficiency, leading to customer dissatisfaction, and lengthens service delivery delays.

Our software system will offer solutions to all the issues above in order to make the company thrive and accommodate more clients efficiently. We are aware that purchasing software is costly. However, services offered by our company give you access to years of industry knowledge and to high-quality work that will pay off greatly both now and in the future, all at a reasonable price that you can choose depending on your needs from a table that will be attached. We can ensure a system without delivery delays that allows you to start integrating the new software into the current working system in no time, all while keeping our client in touch and welcoming change through the process. The proposed system is a hybrid platform, seamlessly integrating a responsive web interface

and a mobile application. This ensures accessibility for both desktop and mobile users, catering to diverse user preferences and operational requirements. By offering a unified user experience across platforms, the system aims to enhance user engagement, operational efficiency, and accessibility.

Some of our suggested solutions would be solving the issue that deals with data scattering and duplication using a database management system, In addition to manual process issues, in a way that the company will have a smooth transition from their old system to our new efficient management system. Another concern that was tackled was allowing for an innovative form of interaction between the company and employees, in addition to customers. Our system considers client privacy as a main concern, which is dealt with through our experts working in the security department. In our program, obliviousness is not taken lightly in terms of not knowing every aspect that deals with the current company, so we will focus on creating an inventory management system that is simple and keeps track of stock and notifies the company about the stock's status. Another top priority is customer communication, which will be made flexible as well as smooth for both the customer and employee.

Proposed Solution

1. DBMS and GUI :

Our software will use a powerful Database Management System (DBMS) to ensure Services-Inc's important data is stored, accessed, and managed safely and efficiently. As the company grows, the system can handle more customer and operational data. It will include top-notch security features like encryption and access controls to keep sensitive information safe, and it will integrate seamlessly with backup and recovery processes to maintain data integrity and allow quick recovery if something goes wrong. The new system will include a data migration module designed to import and validate existing data from Excel sheets and other manual records into the centralized database.

This module will ensure data integrity, eliminate duplicates, and provide tools to map old formats into the new system structure. Historical data from Google Sheets, Excel, and paper records will be digitized and integrated using automated workflows and manual reviews. The software will also feature an easy-to-use Graphical User Interface (GUI) that simplifies everyday tasks such as scheduling, managing employees, and generating reports. The GUI will be implemented as a hybrid platform, available on both web and mobile platforms. This design ensures a consistent user experience across devices, allowing users to access the system functionalities. The GUI will have customizable settings for different user roles, including management, HR, and service staff, providing a personalized experience for each type of user. It will also be designed for accessibility, following standards like WCAG to help users with disabilities. Overall, the combination of the DBMS and GUI will create a secure, effective, and user-friendly system that boosts productivity and supports the company's ongoing growth.

2. AI Prediction System:

Our software will feature a powerful artificial intelligence-based prediction system that will be integrated throughout the platform, offering diverse capabilities across several areas of the company. This system will generate comprehensive reports, charts, and statistical data with high accuracy and real-time performance, allowing the management team to analyze the company's current status and make data-driven decisions efficiently. These insights will help the company define its future path based on predictions from real-time data stored securely in the database management system, ensuring that all critical data is protected and accessible at all times. Additionally, the system will evaluate customer ratings, employee cooperation, and overall company performance to suggest valuable solutions for any potential issues, increasing customer satisfaction and ensuring higher customer retention and profits. The AI prediction system is designed to support the company's growth and will benefit various departments. In the Human Resources (HR) department, the system will predict employee productivity, potential turnover, and staffing needs, and the training requirements for employees, as well as integrate with payroll systems such as ADP to align payroll management with employee performance data helping to optimize workforce management quickly and accurately. For the financial department, the system will forecast revenue, expenses, profits, and losses, offering valuable insights into cost-saving opportunities while maintaining reliable and secure financial data. Furthermore, the system will continuously monitor and analyze inventory levels in real-time, providing accurate, smart recommendations for restocking based on demand trends, thus preventing overstocking or shortages. These inventory suggestions will be especially beneficial for the maintenance department, ensuring service efficiency by minimizing delays due to inventory shortages. The system will also incorporate smart scheduling, optimizing the assignment of employees to tasks and routes based on real-time data, improving resource allocation and service delivery. Lastly, by analyzing customer and operational data, the AI system will help the company maintain and increase profitability while improving overall business performance and strategic decision-making. All components of the system are designed for high reliability.

3. AI ChatBot :

The WHO estimates that by 2030, more than 2.5 billion people with disabilities will require one or more assistive technologies (2 January 2024, WHO). When designing software systems, it's crucial to consider the diverse needs of all users, regardless of their background. In the tech industry, people with disabilities are often underserved. Our program aims to address this by first focusing on individuals with hearing impairments who cannot communicate via phone. Our AI chatbot can convert their distorted speech into fluent text or use a live camera to interpret sign language or video input. The system is optimized to respond instantly, ensuring smooth interaction and minimal delays for the user. Additionally, we provide support for individuals with visual impairments by translating Braille or offering an audio-to-text option that the bot can understand. High availability will guarantee that users can access the chatbot without interruptions, and the system can accommodate growing user numbers without sacrificing performance. Another key feature is the chatbot's ability to personalize the user experience by remembering common requests and providing reminders for billing dates, recent checkups, or past purchases. The chatbot can manage personalized requests across a wide user base, delivering consistently reliable service. All these additions allow for a more inclusive program that increases user loyalty.

4. Security department :

IT services often involve sensitive data, and the Security Department ensures strict protocols for handling client information. Our software will manage and enhance data safety and integrity, adhering to high security standards like end-to-end encryption. Only authorized staff can access protected areas using biometric data, ensuring customer confidence and data protection. The system also includes a robust backup and recovery system, storing essential information securely using cloud-based solutions and local storage. Backups are streamlined to minimize time and storage use, and incremental backups reduce unnecessary usage. Advanced encryption methods are used in transit and at rest to protect sensitive information. The recovery process is regularly tested for dependable restoration during emergencies, with a disaster recovery plan in place. Real-time data replication is used for mission-critical services, minimizing data loss risk. This system is designed for high reliability and maintains operational continuity even in challenging scenarios.

5. GPS Feature :

The GPS feature of our software tracks the current locations of service personnel and vehicles, enhancing operational efficiency by optimizing schedules and reducing travel times and costs with suggested routes. It improves appointment management through live location data, allowing dispatchers to efficiently assign tasks to the nearest employee. As the company grows, the system can scale to support an expanding community and workforce. This feature also reduces customer wait times and increases the number of daily service calls. Additionally, our software integrates seamlessly with third-party GPS apps like Google Maps API, Waze, and TomTom for enhanced tracking and route optimization.

6. Employee performance and QA :

Our software enhances employee productivity and ensures quality assurance by integrating comprehensive tracking and testing features. It includes a progress tracker that monitors working hours and evaluates productivity through real-time updates, helping produce accurate performance reports. This enables managers to generate detailed reports and charts for recognizing and rating employee performance, ensuring that standards are consistently met and time is utilized efficiently. The software also supports employee self-evaluations to promote accountability and focus. For quality assurance, the software incorporates expert advice from strategic consultants to adhere strictly to company regulations. It conducts thorough usability tests that are closely compared against these standards to ensure that all services meet the required quality requirements. This careful process helps minimize errors and enhances customer trust by ensuring that every service is perfected. By implementing these processes, the software not only increases productivity but also guarantees high-quality outcomes in all operations.

7. Clocking in system:

Our software introduces a location-based clock-in system using geofencing technology, in combination with a biometric clock-in system, to enhance attendance management and ensure greater accuracy and security. By setting up a virtual perimeter around the company premises using Geofencing APIs from platforms like Google Maps API and Apple's Core Location Framework, the system automatically detects when an employee enters or exits the designated area. As soon as the employee is within the geofence, the system automatically registers their presence, while the biometric verification ensures further validation when required. This hybrid solution guarantees accurate time tracking, minimizes errors, and reduces the possibility of manipulation. It also eliminates the need for manual attendance logs, providing a seamless and secure method of recording work hours. The user-friendly interface ensures ease of use, while real-time location detection reduces clock-in delays during busy periods. By combining geofencing and biometric authentication, the system offers a reliable, efficient, and secure way to manage employee attendance, boosting payroll accuracy and minimizing administrative costs.

Software development process

We've chosen the Scrum framework for developing our software because it's critical to projects where requirements may shift as work progresses. Scrum allows us to regularly update the project based on feedback, enabling quick adaptation to new requirements from Services Inc. This ensures we remain focused on the most important features and can address any issues immediately .

Scrum is ideal for this project as it holds close collaboration with all clients , from company leaders to end users. Through continuous feedback, we ensure that the project stays aligned with business needs, allowing for quick adjustments when necessary. The software will be delivered in small, manageable increments, with each part thoroughly tested before moving to the next. This strategy not only minimizes risks but also keeps the project on schedule by addressing technical challenges in smaller, controlled steps.

The software is planned in stages, each designed to deliver tangible business value. Initially, we'll focus on building the basic framework and implementing essential user features. Following that, we will introduce more complex functionalities, such as the AI system for HR and tools for managing inventory and finances. Afterward, we will integrate third-party services like Google Maps for tracking and ADP for payroll, ensuring secure and seamless connections with these platforms. Our release plan prioritizes business value while also managing technical risks, ensuring that each sprint directly supports the operational needs of Services Inc.

Each sprint, lasting a couple of weeks, focuses on a set of features. Daily stand-up meetings allow the team to share progress and address any obstacles immediately, helping maintain momentum. At the end of each sprint, we hold a sprint review, where the client can review the progress and provide feedback. This keeps the development on track and aligned with expectations. Additionally, we hold a Sprint retrospective to reflect on the process, identifying areas for improvement and enhancing our workflow.

Prototyping plays a key role in our development approach, particularly for complex features. Early prototypes give developers and the client a clear visual reference, organizing discussions and providing early feedback, which helps guide further development.

Even after the software is launched, we don't consider the process finished. Continuous adjustments based on user feedback and evolving business conditions is a critical part of our approach. Scrum's flexibility allows us to welcome changes, even late in development and after, ensuring the final product remains up to date.

By using Scrum, we ensure that our development process is both transparent and adaptable, aligning closely with the evolving needs of Services Inc. This method supports a functional environment, delivering high-quality software that grows and improves alongside the business.

Phase 2:

User Requirements:

1. The system shall be able to notify the maintenance team about the stocks current status in order to prevent shortages as well as waste reduction in terms of overordered parts.
2. The system should support intelligent scheduling mechanisms in order to improve task allocation as well as routing to aid the maintenance team, GPS tracking, and predictions to ensure any form of service is done in a timely manner with the highest resource utilization.
3. The system shall allow users to submit, modify, or cancel detailed service requests with real-time validation, tracking, and notifications.
4. The system shall replace scattered Excel and Google Sheets with a centralized database that supports high-volume data storage.
5. The system shall provide a reliable reporting system that provides enlightenment into operational performance as well as inventory and customer satisfaction using the feedback that is returned for more insightful decision-making.
6. The system shall provide an advanced customer portal with comprehensive feedback management and communication capabilities to enhance service delivery, accessibility, and customer satisfaction.

7. The system shall provide a mobile application to cover all aspects across the board for all different roles in the company, ranging from the technicians to managers to supervisors to normal employees, for better communication and updates in real time.
8. The system shall provide a comprehensive logging system that tracks user activity and assigns unique identifiers for organizing and managing service requests efficiently, as well as secure user authentication and role-based access control for employees and customers.
9. The system shall protect customer data, employee information, and sensitive business records from unauthorized access, breaches, and data loss while keeping up compliance with industry standards and regulations.

System Requirements :

1.The system shall be able to notify the maintenance team about the stocks current status in order to prevent shortages as well as waste reduction in terms of overordered parts.

1.1: The system shall have seamless integration with a predefined inventory management system such that the maintenance team is able to receive real-time data feeds.

1.2: The system shall inform staff about stock through notifications based on the highest demand items or seasonal items in order to keep stock supplied at all times.

1.3: The system shall supply the maintenance team with automatic alerts when stock levels are running low as well as an optimal reorder operation in terms of having everything necessary tracked and ordered at once.

1.4: The system shall employ machine learning for predictions from previously collected data to take note of current trends and seasonal patterns for possible future inventories.

1.5: The system should send notifications that are tailored based on the recipient's role; for example, managers receive summary updates, whereas staff receives alerts that are actionable (in case of any restocking or updates on stock).

1.6: The system should provide an option for batch/lot tracking of parts to enable any maintenance group to track some items back to their source in case of any problems arising.

2. The system should support intelligent scheduling mechanisms in order to improve task allocation as well as routing to aid the maintenance team, GPS tracking, and predictions to ensure any form of service is done in a timely manner with the upmost resource utilization.

2.1: The system shall implement scheduling algorithms that use AI to take full advantage of task allocation and routing.

2.2: The system shall integrate with GPS systems for real-time location tracking.

2.3: The system should use predictive analytics to prepare for delays and easily reschedule tasks.

2.4: The system should provide users with real-time updates and notifications about schedule changes.

2.5: The system shall monitor the availability of vehicles and equipment, ensuring they are taken into consideration when assigning tasks and routes.

2.6: The system should ensure balanced task distribution among technicians, avoiding overloading any individual or team or in case any technician is off duty.

2.7: The system shall optimize routes to minimize fuel consumption and increase the time-saving aspect.

2.8: The system shall allow the management team to track the status of tasks in real-time, providing transparency into ongoing operations.

2.9: The system shall send automated notifications to customers about estimated arrival times and any delays in service.

3. The system shall allow users to submit, modify, or cancel detailed service requests with real-time validation, tracking, and notifications

3.1: The system shall display a form for users to submit service requests with fields for service type (networking, Wi-Fi, and laptop/PC maintenance), description, and urgency level (high, mid, low), and location (given when necessary, like operations done in households).

3.2: The system shall validate all service request fields before allowing submission (required fields cannot be empty).

3.3: The system shall allow users to edit or cancel their submitted service requests before company approval.

3.4: The system shall notify users via email or SMS upon successful submission of their service request, including their request details.

3.5: The system shall enable users to track the status of their service requests (the status should be one of the following: pending, approved, in progress, or completed).

3.6: The system shall ensure that service request submission can be completed seamlessly on both web and mobile platforms.

3.7: The system shall allow users to upload attachments (photos or documents) relevant to their service requests.

3.8: The system shall detect and prevent duplicate service requests for the same issue submitted by the same user by providing the user with a friendly alert (you have submitted this request before our team will respond shortly).

3.9: The system should validate entered locations for accuracy and completeness when required.

4. The system shall replace scattered Excel and Google Sheets with a centralized database that supports high-volume data storage.

4.1: The system shall implement a robust database management system (DBMS) that is able to handle high-volume data storage, retrieval, and management.

4.2: The system shall support seamless importing and migrating of existing data from Excel, Google Sheets, and other formats, ensuring data integrity and consistency during migration.

4.3: The system shall validate data during migration and entry, ensuring only proper and consistent data is stored in the database.

4.4: The system shall implement role-based access controls to ensure data is accessed, modified, or viewed only by authorized staff.

4.5: The system shall include automated backup and recovery mechanisms to prevent data loss during unexpected failures.

4.6: The system shall allow for real-time updates to the database, ensuring all users see the latest data updates across the database in accordance to their privilege when accessing data.

4.7: The system shall support indexing and advanced search functionalities to allow for quick retrieval of specific data.

4.8: The system shall maintain an audit log to track all changes made to the database, including additions, updates, and deletions, for security and accountability (certain information such as the name, IP address, location, date, and time of whomever was in charge of the last operation for transparency).

4.9: The system shall support archiving of historical data to uphold database efficiency while preserving access to older records.

5.The system shall provide a reliable reporting system that provides enlightenment into operational performance as well as inventory and customer satisfaction using the feedback that is returned for more insightful decision-making.

5.1: The system shall provide an option for detailed reports that are tailored to the needs of the employee and according to their privilege to increase efficiency.

5.2: The system shall use real-time data to produce up-to-date reports on important aspects, such as service delivery and customer feedback, as well as provide alert reports in case of violations and low customer ratings to improve weakness points.

5.3: The system shall provide tools for observing previous data to identify recurrences and to better the decision-making process.

5.4: The system should report metrics such as average response time, task completion rates, and service downtime to keep track of system quality.

5.5: The system shall include reports that assess employee performance based on the number of tasks completed and customer feedback on the employee.

5.6: The system should allow reports to be exported in multiple formats (PDF, Excel, Docx, and CSV) for trouble-free sharing of reports.

5.7: The system shall ensure reports can be accessed and viewed on desktops, tablets, and mobile devices.

5.8: The system should allow users to filter reports by criteria (date range, location, or service type) for better analysis.

6.The system shall provide an advanced customer portal with comprehensive feedback management and communication capabilities to enhance service delivery, accessibility, and customer satisfaction.

6.1: The system shall provide an intuitive and easy-to-navigate interface for customers to submit, modify, and track service requests seamlessly.

6.2: The system shall enable customers to submit feedback on services using written reviews through the mobile application reviews section or the website.

6.3: Feedback forms shall be integrated into the portal and chatbot to streamline the submission process.

6.4: The system shall analyze feedback using AI-powered sentiment analysis to detect customer sentiment (positive, negative, or neutral). If the feedback is identified as negative, it shall be escalated to the management team to address the issue quickly, find a solution, and prevent similar problems from happening again, as well as focusing on positive reviews in order to reinforce excellent employee behavior.

6.5: A feature shall allow customers to submit feedback anonymously to encourage honest input.

6.6: The system shall allow customers to edit or cancel the feedback, and all the feedback shall be securely stored to protect customer privacy, as well as accessibility for customer reference and transparency, all while storing it in the database.

6.7: The system shall implement a reward mechanism, such as discounts or loyalty points, to encourage meaningful feedback from the customer.

6.8: The system shall schedule and manage regular feedback review meetings for employees and managers to act on collected feedback.

6.9: Automated notifications shall be sent to customers for service request updates, feedback acknowledgments, and actions taken based on feedback.

6.10: The system should support customers with disabilities and it should include: speech-to-text for voice queries, real-time sign language interpretation using camera input, and Braille translation to ensure inclusivity for visually impaired users.

6.11: The system shall include an AI chatbot available 24/7 for immediate responses to customer inquiries. Complex or urgent queries shall be escalated to human support through a real-time issue escalation feature.

7.The system shall provide a mobile application to cover all aspects across the board for all different roles in the company, ranging from the technicians to managers to supervisors to normal employees, for better communication and updates in real time.

7.1: The system shall notify technicians of assigned tasks through the mobile application, including task details (location, customer, materials needed).

7.2: The application shall incorporate GPS into the system to provide optimized routes (shortcuts, steering away from traffic) in order for technicians to reach their destinations efficiently, thus reducing time.

7.3: The system shall allow technicians to update the status of tasks (ongoing, completed) in real time through the application, as well as provide time and location of completion.

7.4: The application shall support in-app communication (calls, messages) for technicians to communicate with customers or supervisors or even each other.

7.5: The system shall display tasks to the technician that is closest in location to the customer, thus increasing rapid delivery and customer satisfaction.

7.6: The application shall provide technicians with the contact details of customers (name, location, number) for direct communication when necessary.

7.7: The application should include a checklist for technicians to tick off completed tasks during service delivery.

7.8: The system shall allow supervisors to track technician locations in real time for better coordination and transparency.

7.9: The application should provide technicians with feedback options on task completion, including challenges encountered to avoid future challenges.

7.10: The system shall enable technicians to update inventory usage (name of the item used) and report equipment (name of tool) directly from the application to avoid running out of inventory.

7.11: The system should allow technicians to report unforeseen issues encountered during tasks with photo or document attachments.

7.12: The application shall display the schedule organized on a calendar view available for all the technicians.

8.The system shall provide a comprehensive logging system that tracks user activity and assigns unique identifiers for organizing and managing service requests efficiently, as well as secure user authentication and role-based access control for employees and customers.

8.1: The system shall allow users to log in using their registered username or email , as well as their password.

8.2: The system shall provide a secure registration form that requires the following: username (must be unique), email address (must be validated with regular expressions and confirmed), password (must abide by the security requirements), phone number, and optional fields (profile picture).

8.3: The system shall maintain detailed logs of user activities, including logins, data modifications, and service request submissions, for audit purposes:

8.3.1: User Logins: Record details when a user logs in, including: username or unique identifier, timestamp of the login, IP address (if applicable), device/browser details.

8.3.2: Data Modifications: Log changes to critical data, including: (What data was modified, the original data and the new data, who made the modification, when the modification occurred).

8.3.3: Service Request Submissions: Capture details of any service requests submitted, including the request ID, type of request (network maintenance, Wi Fi, PC and laptop maintenance, printing machine maintenance), username, and timestamp.

8.4 The system shall generate a unique identifier for every service request submitted, linking it to the respective user and assigned employee.

8.5:The system shall allow users to reset their passwords through a secure link sent to their registered email.

8.6: The system shall send real-time notifications to users upon successful logins or failed attempts (if it was a successful login the user will receive this notification “Successful login” otherwise the user will receive “Unsuccessful login” which usually occurs when the wrong information is entered).

8.7: The system shall validate credentials and provide clear and user-friendly error messages such as “Invalid username or password.” (When the user enters the wrong username or password, or even both in some cases) or "Email is already in use." (When a user tries to register for a service and enters an email address that is already

associated with an existing account in the system, the system prevents them from creating a duplicate account using the same email).

8.8: The system shall allow users to update their profile details, including email, phone number, and password, while maintaining validation rules.

8.9: If the phone number is verified and the proper data is entered, then the system will proceed with storing the user data in the database.

8.10: Provide a "Remember Me" option to keep users logged in securely by remembering their username and password that are kept as defaults under user permission to speed up the logging in process .

The system's non-functional requirement:

9.The system shall protect customer data, employee information, and sensitive business records from unauthorized access, breaches, and data loss while keeping up compliance with industry standards and regulations.

9.1: The system shall implement role-based access controls to ensure that users can only access data and features relevant to their privileges given to them depending on their roles.

9.2: The system shall encrypt sensitive data, both used and unused, using industry-standard encryption protocols.

9.3: The system shall offer two-factor authentication for an additional layer of login security (FaceID and Password).

9.4: The system shall maintain detailed logs of all user activities, including data access, modifications, and logins, for security and law audits.

9.5: The system shall enforce strong password policies, including complexity requirements (a minimum of 8 characters, including upper/lowercase letters, numbers, special characters, as well as avoiding personal information). Moreover, regular expiration intervals (every six months the password must be updated).

9.6: The system shall freeze user accounts temporarily (one hour) after five failed attempts of login to prevent any possible attack.

9.7: The system shall provide strong encryption for the backups that have been done by the database management system. Moreover, recovery processes to protect against accidental data loss or system failures.

9.8: The system shall do security updates regularly in order to address any sudden vulnerabilities.

9.9: The system shall automatically log users out if they've been inactive for a while, helping to protect against unauthorized access and manipulation.

9.10: The system shall continuously provide alert notifications of any possible attacks, unauthorized access, or breaches to the management team.

The cost and effort:

- Function Points (FP): Assigned on a scale of 0-5.
- Effort per FP: 8 hours per function point.
- Working Days per Month: 22 days.
- Developer Salary: \$2,000/month (\$90.91 per day, \$11.36 per hour).
- Profit Margin: 40%-60%.

Requirement	Function Points (FP)	Total Function Points	Number of Days
Inventory Management System	4	32	4
Intelligent Scheduling	5	40	5
Service Request Management	3	24	3
Centralized Database	5	40	5
Reporting System	4	32	4
Data Security	4	32	4
Advanced Customer Portal	5	40	5
Mobile Application	5	40	5

Requirement	Function Points (FP)	Total Function Points	Number of Days
Logging System	4	32	4
Total	38	321	39

Explanation of Totals:

- **Function Points (FP):** 38 points total across all requirements.
- **Total Function Points:** 312 hours of total effort.
- **Number of days:** 39 working days based on 8 hours/day.

Costs:

- **Base Cost (\$):** \$4,090.91 based on \$11.36/hour developer rate.
- **Min Offer (40%) (\$):** \$5,727.28
- **Max Offer (60%) (\$):** \$6,545.46

Step 1: Identify Functionality

Functionality	Description
Inventory System Management	Notify the maintenance team about stock status to prevent shortages and overordering.
Intelligent Scheduling	Optimize task allocation and routing using AI and GPS.
Service Request Management	Allow users to submit, modify, and cancel service requests with real-time tracking and validation.
Centralized Database	Replace Excel and Google Sheets with a centralized database to handle high-volume data.
Reporting System	Provide reliable operational performance and customer feedback reports.
Data Security and Privacy	Ensure protection of sensitive data, compliance with regulations, and prevent unauthorized access.
Advanced Customer Portal	Provide feedback management and communication capabilities to enhance satisfaction.
Mobile Application	Enable real-time updates and communication for all roles via a mobile app.
Logging and Authentication	Track user activity, assign unique identifiers, and ensure secure authentication.

Step 2: Assign Function Points

Functionality	Complexity Level (Function Points)
Inventory System Management	4
Intelligent Scheduling	5
Service Request Management	3
Centralized Database	5
Reporting System	4
Data Security and Privacy	4
Advanced Customer Portal	5
Mobile Application	5
Logging and Authentication	4

Step 3: Calculate Total Function Points:

Add up the function points for all functionalities:

$$\text{Total Function Points} = 4+5+3+5+4+4+5+5+4 = 39$$

Step 4: Convert Function Points to Effort:

Use the given conversion factor (8 hours per function point) to calculate the total effort in hours:

$$\text{Effort (Hours)} = \text{Total Function Points} \times 8 = 39 \times 8 = 312 \text{ hours}$$

Convert hours into days (assuming 8 working hours per day):

$$\text{Effort (Days)} = \text{Effort (Hours)} / 8 = 312/8 = 39 \text{ days}$$

1. Base Cost (\$):

The Base Cost is calculated as:

$$\text{Base Cost} = \text{Effort (Hours)} \times \text{Hourly Rate of Developer}$$

$$\text{Hourly Rate of Developer: } \frac{\text{Monthly Salary}}{\text{Hours Worked per Month}}$$

$$\text{Hourly Rate} = \frac{2000}{22 \times 8} = \frac{2000}{176} \approx 11.36 \text{ \$/hour}$$

For each requirement:

$$\text{Base Cost} = \text{Effort (Hours)} \times 11.36$$

2. Min Offer (40% Profit):

The Min Offer adds a 40% profit margin to the Base Cost:

$$\text{Min Offer} = \text{Base Cost} \times (1 + 0.40)$$

For example:

$$\text{Min Offer for Stock Notification} = 363.64 \times 1.40 = 509.10 \$$$

3. Max Offer (60% Profit):

The Max Offer adds a 60% profit margin to the Base Cost:

$$\text{Max Offer} = \text{Base Cost} \times (1 + 0.60)$$

For example:

$$\text{Max Offer for Stock Notification} = 363.64 \times 1.60 = 581.82$$

General Calculation Steps for Each Requirement:

1. Calculate Base Cost: $\text{Effort (Hours)} \times 11.36$
2. Calculate Min Offer: $\text{Base Cost} \times 1.40$
3. Calculate Max Offer: $\text{Base Cost} \times 1.60$

Phase 3 :

Task 3.1:

Deema:

Scenario Analysis: View Operational Performance

Reports

1. Initial assumption:

The manager and/or supervisor is logged into the system and has navigated their way to the “View Operational Performance” feature. They would like to generate a report that is either chosen from a list of options or customize a report based on certain metrics .

2. Normal Flow:

2.1 The Manager and/or Supervisor makes their way to the reports feature that is available at all times.

2.2 The Manager and/or Supervisor has the option to filter reports based on the desire to ensure compatibility with every specialty of work.

2.3 The system displays the reports on dashboard based on information that is retrieved from real time data.

2.4 The Manager and/or Supervisor will proceed to review and analyze report , as well as having the option to export the report in desired format.

3. Alternative Flow 1(Customized Reports):

3.1.1 The Manager and/or Supervisor requires a more customized view of operational performance.

3.1.2 The Manager and/or Supervisor makes their way to the Customized Reports feature from the dashboard.

3.1.3 The system displays options where the user can select from choices like (Date range, Service location, Performance thresholds)

3.1.4 The system validates the user's selected options and proceeds with request.

3.1.5 The system generates a detailed report that matches the specified options using real-time or previously stored data.

3.1.6 The Manager and/or Supervisor reviews the report and exports it in a chosen format (PDF, Excel, Docx, or CSV).

4. Error Flow (Report Generation Failure):

4.1 The user attempts to generate a report, but the system fails to proceed with operation due to issues like missing data or system failure.

4.2 The system displays an error message with details to clarify reason for incompleteness of task ("Dear user please be patient we are encountering system failure or no data is available").

4.3 The user may choose to retry or ask for technical support.

4.4 Once the issue is reported to technical support and a solution is found the user may proceed with the report request.

5. Outcome:

5.1 The user successfully accesses, views, and exports operational performance reports tailored to their requirements.

5.2 Insights taken from report allow for better decision-making and increasing operational efficiency through conclusions drawn from reports.

Mustafa:

Scenario Analysis: Submit Feedback with Rewarding Mechanism

1. Initial assumption

The Customer has logged in and completed a service request. The service is marked as "completed" in the system, and the customer wants to submit feedback for the service.

2. Normal Flow of Events

2.1 The Customer navigates to the "Feedback" section.

2.2 The system displays a list of completed services.

2.3 The Customer selects a specific service and opens the feedback form.

2.4 The system presents fields for:

2.4.1 Rating (mandatory, 1–5 scale).

2.4.2 Comments (optional but only allowed with a rating).

2.4.3 Option to submit feedback anonymously.

2.5 The **Customer** provides a 4 stars rating, writes comments (for example : "Excellent service!"), and submits the feedback.

2.6 The system validates the input:

2.6.1 Ensures the rating is provided.

2.6.2 Checks those comments, if present, are accompanied by a rating.

2.7 The feedback is securely stored in the database.

2.8 The system analyzes the feedback and determines it has positive sentiment.

2.9 The system awards 10 loyalty points to the Customer.

2.10 The system notifies the Customer: "Thank you for your feedback! You earned 10 loyalty points."

3. Alternative Flows

3.1: Submit Feedback Anonymously

3.1.1. The Customer selects the "**Submit Feedback Anonymously**" option.

3.1.2. The system stores the feedback without linking it to the customer's profile.

3.1.3. The system does not award any loyalty points.

3.1.4. The system notifies the Customer: "Thank you for your feedback! Your input is valuable to us."

3.2: Rating Only (No Comments)

3.2.1 The Customer provides a rating (e.g., 4 stars) but does not add any comments.

3.2.2 The system validates the input:

3.2.2.1 Ensures the rating is provided.

3.2.3 The system stores the feedback, classifies it as positive, and awards 5 loyalty points.

3.2.4 The system notifies the Customer: "Thank you for your feedback! You earned 5 loyalty points."

4. Error Flows

4.1 Incomplete Feedback Submission:

4.1.1. The Customer attempts to submit comments without a rating.

4.1.2. The system displays an error message: "A rating is required to submit feedback."

4.1.3. The Customer corrects the input and resubmits the feedback.

4.2 Attempt to Submit Feedback Without a Completed Service:

4.2.1 The Customer tries to access the feedback form but has no completed service requests.

4.2.2 The system displays an error message:

4.2.2.1 "Feedback cannot be submitted without a completed service request."

4.2.3 The system redirects the Customer to the "Submit Service Request" section.

4.3 System Error During Submission:

4.3.1 The system encounters a database error while saving the feedback.

4.3.2 An error message is displayed: "An unexpected error occurred. Please try again later."

4.3.3 The feedback is not saved, and the Customer must retry later.

5. Concurrent Activities

5.1 While the Customer is filling out the feedback form, the system dynamically validates input fields to ensure:

5.1.1 A rating is provided.

5.1.2 Comments, if present, are correctly linked to a rating.

6. Ending State

6.1 The feedback is securely stored in the database.

6.2 Loyalty points (if applicable) are awarded to the customer's account:

6.2.1 Rating only → Minimal points (5 points).

6.2.2 Rating + comments → Better points (10 points).

6.2.3 Anonymous feedback → No points.

6.3 Positive feedback is logged, and negative feedback is escalated to the Manager for review.

Aseel:

Scenario Analysis: Update Inventory Usage

1. Initial Assumption:

The maintenance member and/or manager are logged into the system. They would like to update the inventory through using the “Update Inventory Usage” feature. They may choose to add a new item to the inventory or update the status of a present item.

2. Normal Flow:

2.1 The Maintenance member accesses the “Update Inventory” feature on the mobile app or website.

2.2 The Maintenance member is then prompted to input inventory details such as the item code or name, quantity used, and optional notes.

2.3 The system double checks the inputs existence in the inventory to ensure accuracy. If the item exists, the system proceeds to the next step. However, if the item does not exist, an alternative flow is triggered.

2.4 After the system double checks the item is then added to the inventory based on the entered data.

2.5 The system calculates the updated stock level by subtracting the entered quantity from the existing quantity.

2.6 If an item's stock reaches zero, the Manager is notified to restock the item.

3. Alternative Flow 1(Item not found):

3.1 The Maintenance member enters an item(either name or code) but are not able to locate it in the inventory system.

3.2 The system prompts the Maintenance member to submit a request to add the new item, which must include (Item Name,Item Description, Stock Quantity).

3.3 The Manager receives the request for approval.

3.4 Once the request is approved, the Manager adds the new item into the system.

3.5 The system notifies the Maintenance member that the item has been successfully added.

3.6 The Maintenance member may now proceed to log the inventory usage for the newly added item.

4. Error Flow (Invalid Data Input):

4.1 The Maintenance member enters invalid data such as item codes that are non existent or Non-numeric values in the quantity field, Irrelevant or no item names.

4.2 The system checks if the input is valid and detects errors.

4.3 The system displays an error message (“Dear user please enter valid input”) indicating the cause of the invalid input and prompts the team to re-enter the correct information.

4.4 The Maintenance member re-enters valid data, and the system proceeds with the inventory update process ensuring no invalid data is taken into consideration.

5. Outcome:

5.1 The inventory provides the manager and supervisor with accurate and up-to-date stock levels .

5.2 Low stock and out of stock alerts have been successfully recorded and brought to the Managers and Supervisors attention.

5.3 A complete record of inventory updates, including quantities , Maintenance member actions, and item details, is maintained for reporting and analysis.

5.4 The system ensures seamless communication between the Maintenance member and Manager to ensure restocking and inventory management is done in proper manner thats up to standard.

Fadi:

Scenario Analysis: Task Scheduling

1. Initial Assumption:

- 1.1. A customer requires IT maintenance services and submits a service request through the company's online website or mobile application.
- 1.2. The system is operational, and technicians are available for assignment.
- 1.3. The intelligent scheduling system is active and ready to process incoming tasks.

2. Normal Flow:

- 2.1. A customer submits a service request through the company's online website or mobile application.
- 2.2. The system validates the request, ensuring all necessary details (task type, urgency, and location) are provided.
- 2.3. The intelligent scheduling system identifies the most suitable technician based on proximity, workload, and skillset.
- 2.4. The system assigns the task to the selected technician and sends a notification to their mobile application with all relevant details (task type, customer location, tools required, and task description).
- 2.5. The technician accepts the task and uses the integrated GPS to navigate to the location.
- 2.6. After completing the task, the technician updates the task status to "completed" in the mobile application, which notifies both the customer and the supervisor.

3. Alternative Flow 1: No Available Technician

- 3.1. If no technician is available at the requested time, the system notifies the supervisor of the unassigned task.
- 3.2. The supervisor reviews the task and manually adjusts the schedule, either by reassigning existing tasks or negotiating a rescheduled time with the customer.
- 3.3. The system updates the task assignment and sends notifications to the supervisor and the manager.

4. Alternative Flow 2: Technician Cannot Reach the Location

- 4.1. If the technician faces issues reaching the location due to any potential circumstances, they report the issue through the mobile application.
- 4.2. The system recalculates the optimal route or reassigns the task to another available technician based on proximity and availability.
- 4.3. Both the customer and supervisor are updated with the new estimated arrival time or technician details.

5. Error Flow: Scheduling or System Malfunction

- 5.1. If the scheduling system encounters a failure (data error or system downtime), the task is flagged as “Unscheduled.”
- 5.2. The system logs the error and notifies the support team for resolution.
- 5.3. Meanwhile, the supervisor is prompted to assign the task manually to ensure no delay in service.
- 5.4. The customer is notified about the delay and provided with an estimated resolution time.

6. Concurrent Activities

- 6.1. Supervisors can monitor multiple technicians in real-time to identify delays or resource allocation inefficiencies.
- 6.2. Technicians can communicate with supervisors and customers through the mobile app while tasks are ongoing.
- 6.3. The inventory management system runs concurrently, ensuring that technicians have the necessary tools and parts for their assigned tasks.
- 6.4. Real-time data synchronization ensures that task updates are visible to all authorized users (technicians, supervisors, managers, maintenance team, and customers).

7. System state on completion

- 7.1. The task is successfully assigned, tracked, and completed.
- 7.2. (Technician, supervisor, manager, maintenance member, and customer) are notified upon task completion.
- 7.3. The system logs all activities for future reference and performance analysis.
- 7.4. Overall, operational efficiency is improved, and customer satisfaction is enhanced.

Amro:

Scenario Analysis: maintain user profile

1. Initial Assumption:

- 1.1. The user has access to the system and is logged in with valid credentials.
- 1.2. The "Maintain Profile" feature is available and functional.
- 1.3. The system has a stable connection to the database for retrieving and updating profile details.
- 1.4. The actor is aware of the required input formats (e.g., email format, password strength).

2. Normal Flow

- 2.1. The actor selects the "Maintain Profile" option.
- 2.2. The system displays the current profile details.
- 2.3. The actor updates their contact information and submits the changes.
- 2.4. The system validates the entered information (e.g., proper email formatting, strong password).
- 2.5. The validation is successful.
- 2.6. The system updates the user profile in the database.
- 2.7. A success message is displayed to the actor: "Profile updated successfully."
- 2.8. The system logs the update activity for auditing purposes.

3. Alternative Flow 1: Validation Fails – Invalid Email Format

- 3.1. The actor selects the "Maintain Profile" option.
- 3.2. The system displays the current profile details.
- 3.3. The actor updates their email address but enters an invalid format (e.g., user@@example).
- 3.4. The system validates the information and detects the error.
- 3.5. An error message is displayed: "Invalid email format. Please enter a valid email."
- 3.6. The actor corrects the email address and resubmits.
- 3.7. The validation is successful.
- 3.8. The system updates the user profile in the database.
- 3.9. A success message is displayed to the actor.

4. Alternative Flow 2: Validation Fails – Weak Password

- 4.1. The actor selects the "Maintain Profile" option.
- 4.2. The system displays the current profile details.
- 4.3. The actor updates their password and submits the changes.
- 4.4. The system validates the password but detects that it does not meet the required strength criteria (e.g., minimum length, special characters, etc.).
- 4.5. An error message is displayed: "Password is too weak. Please choose a stronger password with at least 8 characters, including uppercase, lowercase, numbers, and special characters."
- 4.6. The actor updates the password to meet the requirements and resubmits.
- 4.7. The validation is successful.
- 4.8. The system updates the user profile in the database.
- 4.9. A success message is displayed to the actor.

5. Alternative Flow 3: Missing Field Error

- 5.1. The actor selects the "Maintain Profile" option.
- 5.2. The system displays the current profile details.
- 5.3. The actor attempts to submit the changes but leaves mandatory fields blank (e.g., name or email).
- 5.4. The system detects the missing fields and displays an error message: "Please fill in all required fields."
- 5.5. The actor completes the missing information and resubmits.
- 5.6. The system validates the changes, updates the profile, and logs the update activity.
- 5.7. A success message is displayed: "Profile updated successfully."

6. Error Flow: System Error

- 6.1. The actor selects the "Maintain Profile" option.
- 6.2. The system displays the current profile details.
- 6.3. The actor updates their password and submits the changes.
- 6.4. The system attempts to validate and update the database but encounters a connection error.
- 6.5. An error message is displayed: "Unable to update profile at this time. Please try again later."
- 6.6. The system logs the error for further investigation.
- 6.7. The actor remains on the profile editing screen to retry later.

7. Concurrent Activities

- 7.1. System monitors user activity to prevent simultaneous profile edits from multiple devices.
- 7.2. All profile updates are logged for auditing and security purposes.
- 7.3. Notifications about successful updates or errors are displayed to the user.
- 7.4. Real-time synchronization ensures that updated profile data is available across all connected systems and applications.

8. System State on Completion

- 8.1. The user profile is successfully updated in the database.
- 8.2. Relevant success or error notifications are displayed or sent to the user.
- 8.3. The system logs all profile update activities for future reference and audits.
- 8.4. The system remains ready for the next profile update request.

9. Outcome

- 9.1. The profile update process ensures accurate and up-to-date user information.
- 9.2. User satisfaction is maintained through clear error messages and feedback.
- 9.3. The system's integrity is preserved with robust validation, logging, and monitoring mechanisms.

Task 3.2:

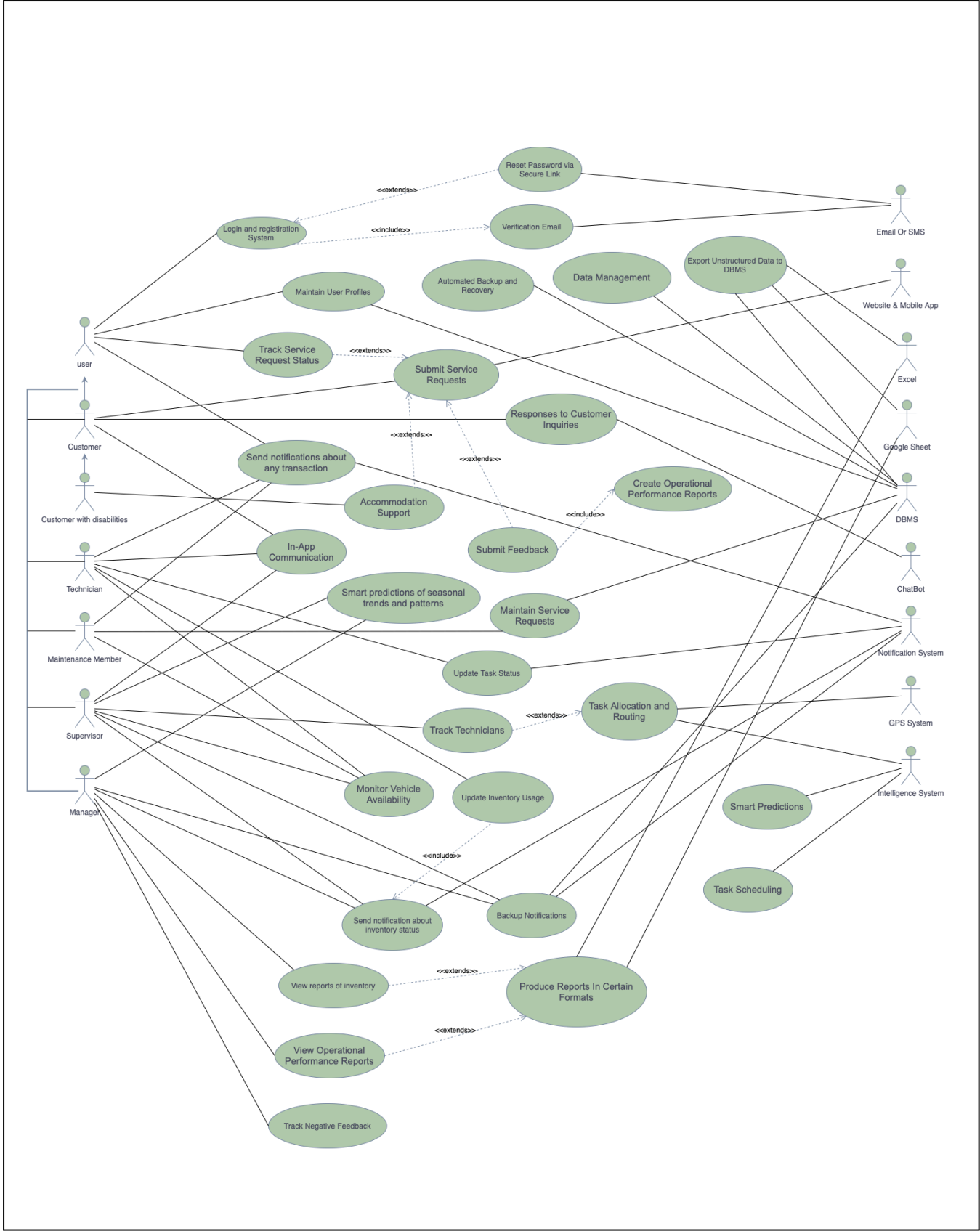
Actors:

- User
- Customer
- Customer with disabilities
- Supervisors
- Manager
- Maintenance member
- GPS System
- Intelligence System
- Technicians
- Notification system
- Email or SMS
- EXCEL AND GOOGLE SHEET
- DBMS
- chatBot
- Website & Mobile Application

Use cases :

1. Login and registration system
2. Reset password via secure link
3. Verification Email
4. Maintain user profiles
5. Smart predictions of seasonal trends and patterns
6. Track technicians
7. Track service request status
8. Monitor Vehicles Availability
9. Submit service request
10. Track negative feedback

11. Create operational performance reports
12. View operational performance reports
13. View reports of inventory
14. Produce reports in certain formats
15. In App communication
16. Backup notifications
17. Data Management
18. Automated backup and recovery
19. Responses to customer inquiries
20. Export unstructured data to DBMS
21. Task Scheduling
22. Smart predictions
23. Update Inventory usage
24. Update Task Statues
25. Send notification about any transaction
26. Send notification about inventory status
27. Task Allocation & Routing
28. Maintain Service Requests
29. Submit Feedback
30. Accommodation Support



Task 3.3:

Deema:

Sahab System Use-Case Specification

View Operational Performance Reports

Version 1.0

Revision History

Date	Version	Description	Author
17/Dec/2024	Draft	First Draft	Deema Abu Nimeh

View Operational Performance Reports

1.1 Brief Description

The managers are able to access, view, and analyze reports on operational performance, inventory status, and customer satisfaction. These reports aid in improving decision-making, tracking performance, and note areas requiring improvement.

1.2 Actor

- 1.1 Maintenance member
- 1.2 Managers
- 1.3 Supervisors

2. Preconditions

- 2.1 The Manager and Supervisor are logged into the system with appropriate privileges.
- 2.2 The Manager and Supervisor have direct access to the latest reports from the system.
- 2.3. Real-time data is available for generating reports.

3 Flow of Events

3.1 Basic Flow - Access and View Reports

- 3.1.1 The Manager and supervisor log into the system and make their way to the “Operational Performance Reports” feature
- 3.1.2 The system provides a dashboard with report options (Such as :Services, Inventory, Customer Feedback).
- 3.1.3 The Manager or Supervisor select the type of report they care to view (employee performance, inventory levels, customer satisfaction).
- 3.1.4 The system allows for filtering options such as date range to track latest updates , locations, and service types for utmost efficiency .
- 3.1.5 The user applies the desired filters, and the system retrieves real-time data to generate the report.
- 3.1.6 The report is displayed on the user’s device.

3.2 Alternative Flow - Customized Reports

- 3.2.1 The user requests a tailored report that focuses on specific metrics.
- 3.2.2 The user selects the “Custom Report” option and specifies the required metrics and filters.
- 3.2.3 If request is approved then the system will proceed to display the custom report for the user to review, analyze, and export as needed.

3.3 Error Flow - Report Generation Failure

- 3.3.1 The user attempts to generate a report, but the system faces an issue(System is down, No data available)
- 3.3.2 The system displays an error message (“Dear user system is down or no data is available ”) indicating the cause of the failure and prompts the user to retry and if failure continues then user is prompted to request support.
- 3.3.3 Once issue is resolved then a report will be successfully generated.

4 Post-conditions (Exit Condition)

- 4.1 The user successfully views and/or exports the necessary operational performance report.
- 4.2 Insights from the report help improve decision-making and operational performance.

5 Special Requirements

- 5.1 The system shall allow reports to be generated in under 10 seconds to ensure efficiency.
- 5.2 Reports are able to be exported in multiple formats (PDF, Excel, Docx, CSV) for easy sharing.
- 5.3 Reports shall be accessible across all kinds of devices .
- 5.4 The system needs to provide tools for viewing historical data to take note of trends and recurrences.

6 Extension Points

- 6.1 This use case extends to "Create Operational Performance Reports" for employee evaluation and performance.

Mustafa:

Sahab System
Use-Case Specification

Submit Feedback with Rewarding Mechanism

Version 1.0

Revision History

Date	Version	Description	Author
16/Dec/2024	Draft	Draft version	Mustafa Alayasa

1. Brief Description

This use case allows the **Customer** to submit feedback for completed service requests. Feedback can include ratings, comments, and an option to submit anonymously. Rewards are provided based on the type of feedback, and negative feedback is escalated to the **Manager** for review.

Actors:

- **Customer**
- **Manager**

2. Flow of Events

2.1 Basic Flow – Submit Feedback

2.1.1. The Customer logs into the customer portal or mobile app.

2.1.2. The Customer navigates to the "Feedback" section.

2.1.3. The system displays a list of completed services.

2.1.4. The Customer selects a service and opens the feedback form.

2.1.5. The system displays fields for:

2.1.5.1. Rating (1–5 scale)

2.1.5.2 Feedback comments (text input)

2.1.5.3 Option to submit feedback anonymously.

2.1.6. The **Customer** provides:

2.1.6.1 A rating (mandatory).

2.1.6.2 Comments (optional but only allowed if a rating is provided).

2.1.6.3 Optionally selects the "Submit Anonymously" option.

2.1.7. The system validates the input fields:

2.1.7.1 A rating must be provided (1–5 scale).

2.1.7.2 Comments cannot be submitted without a rating.

2.1.8. The system securely stores the feedback in the database.

2.1.9. The System Performs AI-Powered Sentiment Analysis

2.1.9.1. The system analyzes the submitted feedback to determine its sentiment:

2.1.9.1.1 Positive Sentiment: Feedback is classified as positive if the rating ≥ 3 .

2.1.9.1.2 Negative Sentiment: Feedback is classified as negative if the rating < 3. Negative feedback is escalated to the Manager for review.

2.1.9.2 Rewards are granted based on the feedback type:

2.1.9.2.1 Rating only: Minimal points are awarded (5 points).

2.1.9.2.2. Rating + comments: Better points are awarded (10 points).

2.1.9.2.3 Anonymous feedback: No points are awarded, regardless of the sentiment.

2.1.10. The system notifies the Customer of the submission status:

2.1.10.1. "Thank you for your feedback! You earned X loyalty points." (for rewarded feedback)

2.1.10.2. "Thank you for your feedback! Your input is valuable to us." (for anonymous feedback).

2.1.11. The Customer can view or track their feedback in the "My Feedback" section.

2.2 Alternatives

2.2.1: Incomplete Feedback Submission

2.2.1.1. The Customer attempts to submit comments without providing a rating.

2.2.1.2. The system displays an error message: "A rating is required to submit feedback."

2.2.1.2. The Customer provides a rating and resubmits.

2.2.2: Attempt to Submit Feedback Without a Completed Service

2.2.2.1 The Customer tries to access the feedback form but has no completed service requests.

2.2.2.2 The system displays an error message:

2.2.2.2.1 "Feedback cannot be submitted without a completed service request."

2.2.2.3 The system redirects the Customer to the "Submit Service Request" section.

2.2.3: Anonymous Submission

2.2.3.1. The Customer selects the "Submit Feedback Anonymously" option.

2.2.3.2. The system stores the feedback without linking it to the customer's profile.

2.2.3.3. Rewards are not provided for anonymous feedback.

2.2.3.4. The system notifies the customer:

2.2.3.4.1 "Thank you for your feedback! Your input is valuable to us."

2.2.4: System Error During Submission

2.2.4.1. The system encounters a database error while saving feedback.

2.2.4.2. The system displays an error message: "An unexpected error occurred. Please try again later."

2.2.4.3. The feedback is not saved, and the Customer must retry later.

3. Special Requirements

3.1. Feedback submission must be available via the web portal and mobile app.

3.2. The system must support accessibility features:

3.2.1 Speech-to-text for voice queries.

3.2.2 Real-time sign language interpretation using camera input.

3.2.3 Braille translation for visually impaired users.

3.3 Feedback analysis must be completed within 5 seconds to ensure responsiveness.

3.4 Anonymous submissions must ensure customer privacy while storing the feedback securely.

3.5 Notifications must be sent to customers for:

3.5.1 Successful feedback submission.

3.5.2 Rewards (if applicable).

3.5.3 Actions taken on negative feedback.

4. Entry Conditions

4.1. The Customer must log into their account.

4.2 At least one service request is marked as completed for the customer.

5. Exit Conditions

- 5.1.** Feedback is securely stored in the system database.
- 5.2.** Rewards are granted (if applicable) based on feedback type.
- 5.3.** Negative feedback is escalated to the Manager.

6.Extension Points

6.1: Validate Service Completion

- Purpose: Makes sure the customer can only submit feedback if their service request is marked as completed.
- When It Happens: The customer tries to give feedback without a completed service.
- Related Use Case: Submit Service Requests

6.2: Escalate Negative Feedback

- Purpose: Sends negative feedback (a rating below 3) to the Manager for review.
- When It Happens: When the feedback is labeled as negative.
- Related Use Case: Track negative feedback

6.3: Generate Feedback Report

- Purpose: Prepares summaries of feedback for managers to review
- When It Happens: When all feedback data is saved and ready.
- Related Use Case: Create Operational Performance Reports

Aseel:

Sahab System
Use-Case Specification

Update Inventory Usage

Version 1.0

Revision History

Date	Version	Description	Author
17/Dec/2024	Draft	First Draft	Aseel Abd El Haq

Update Inventory Usage

1.1 Brief Description

The maintenance member can track inventory usage to stay up to date on stock shortages and stocks consumption to ensure efficient inventory management.

1.2 Actor

- 1.1 Maintenance member
- 1.2 Manager
- 1.3 DBMS

2. Preconditions

- 2.1 The maintenance member is logged into the system.
- 2.2 Maintenance member has the ability to access the mobile or web platform to log inventory usage in real time.

3 Flow of Events

3.1 Basic Flow - Logging Inventory Usage

- 3.1.1 The Maintenance member opens the “Update Inventory” feature through the mobile app or website.
- 3.1.2 The Maintenance member proceeds to enter information about the inventory item/s used, like quantity and relevant notes.
- 3.1.3 The system proceeds to verify if item was in database for upmost accuracy.
- 3.1.4 Once verified the system updates stock levels in the inventory.
- 3.1.5 If said item becomes out of stock after use then the Manager is notified to restock the item.

3.2 Alternative Flow - Item Not Found

- 3.2.1 The Maintenance member cannot locate item in the systems inventory.
- 3.2.2 The Maintenance member sends a request for a new item to be added to the system.
- 3.2.3 If request is approved then item will be added and Maintenance member will be able to use item when necessary.

3.3 Error Flow - Invalid Data Input

- 3.3.1 If the maintenance member attempts to input invalid data(numeric input thats not an item code or name of items irrelevant to items in stock) then team will be prompted with error message (“Dear user please enter valid data”) and an option to re-enter proper data.
- 3.3.2 Maintenance member re-enters proper data and the update process continues smoothly.

4 Post-conditions (Exit Condition)

- 4.1 The inventory reflects the latest and accurate stock levels.
- 4.2 Low stock alerts have been processed and sent to Manager.
- 4.3 Current status of inventory is recorded for report purposes.

5 Special Requirements

- 5.1 The system shall process inventory updates instantly with minimal delays (under 5 seconds).
- 5.2 The system should keep a history of all inventory updates for analysis.
- 5.3 Low-stock alerts must be triggered automatically .

6 Extension Points

- 6.1 This use case includes “Send notifications about any transaction” in order to notify Manager and Supervisor about ongoing inventory changes or status.

Fadi:

Sahab System
Use-Case Specification

Track Technicians Use Case

Version 1.0

Revision History

Date	Version	Description	Author
16/Dec/2024	Draft	Draft version	Fadi Abu Aletham

Track Technicians Use Case

1.1 Brief Description

Supervisors use this feature to monitor the real-time location and progress of technicians for better task allocation and transparency.

1.2 Actor

- 1.2.1 Supervisor
- 1.2.2 Technician
- 1.2.3 GPS System

2. Preconditions (Entry Condition)

- 2.1 The supervisor is logged into the system with proper permissions.
- 2.2 Technicians have the mobile application running with location tracking enabled.

3 Flow of Events

3.1 Basic Flow - Tracking Technician

- 3.1.1 The supervisor accesses the "Technician Tracking" interface on their dashboard.
- 3.1.2 The system retrieves the real-time locations of all technicians using GPS integration.
- 3.1.3 The supervisor selects a specific technician to view details.
- 3.1.4 The system displays the technician's assigned tasks, current status (en route, on-site, completed), and estimated task completion time.
- 3.1.5 The supervisor can reassign tasks or provide updates to the technician if needed.

3.2 Alternative Flow - Technician Location Unavailable

- 3.2.1 If a technician's location cannot be retrieved, the system displays a warning message "The technician's location is currently unavailable." and an option to contact the technician directly.
- 3.2.2 The supervisor attempts to reach the technician via the in-app communication feature.
- 3.2.3 If the issue persists, tasks are reallocated to another technician.

3.3 Error Flow - System Error in Location Retrieval

- 3.3.1 If the GPS service fails, the system logs an error and notifies the support team.
- 3.3.2 Supervisors are prompted to use manual communication and task tracking until the issue is resolved.

4 Post-conditions (Exit Condition)

- 4.1** The supervisor has successfully tracked or reassigned tasks to ensure continued service delivery.

5 Special Requirements

- 5.1** The system must refresh technician locations in real-time with minimal delay (under 5 seconds).

6 Extension Points

- 6.1** This use case extends to "Technician Task Allocation" allowing supervisors to reassign or modify tasks based on the technician's current status.

Amro:

Sahab System Use-Case Specification

Maintain User Profile Use Case

Version 1.0

Revision History

Date	Version	Description	Author
15/Dec/2024	Draft	Initial Draft	Amro Deek

Maintain User Profile Use Case

1. Brief Description

This use case allows users (Customer, Supervisor, Manager) to maintain their profiles in the system. Users can update personal information such as contact details, passwords, and preferences. Supervisors and Managers can additionally update role-specific settings.

1.2 Actors

- 1.2.1 Customer
- 1.2.2 Supervisor
- 1.2.3 Manager
- 1.2.4 Maintenance Member
- 1.2.5 Technician

2. Preconditions (Entry condition)

2.1 Before this use case begins, the actor must be logged into the system.

3. Flow of Events

3.1 Basic Flow - Update Profile

3.1.1 The actor selects "Maintain Profile."

3.1.2 The system displays the current profile details of the user.

3.1.3 The actor modifies fields such as name, phone number, email, password, or profile picture.

3.1.4 The actor submits the changes.

3.1.5 The system validates the entered data.

3.1.6 If validation succeeds, the system updates the profile information in the database and confirms the changes by displaying "Profile updated successfully."

3.1.7 The system logs the update activity for audit purposes.

3.1.8 The use case ends.

4. Alternative Flows

4.1 Validation Fails

- If the system detects invalid data (an incorrectly formatted email or weak password or missing fields), it displays appropriate error messages (incorrectly formatted email or weak password or missing fields) and prompts the actor to correct the information.

4.2 Role-Specific Updates

- For Supervisors or Managers, additional fields related to their roles (team assignments , Work Schedules) can be updated. These updates follow the same validation and submission process as the basic flow.

4.3 Missing Field Error

4.3.1 If the actor attempts to submit the profile update with mandatory fields left blank (e.g., name, email), the system detects the missing information and displays an error message: "Please fill in all required fields."

4.3.2 The actor completes the missing fields and resubmits.

5. Post-conditions (Exit condition)

5.1 Upon successful completion, the user's profile is updated in the database and a confirmation is displayed.

5.2 In case of errors, the system remains in the profile editing state.

6. Special Requirements

6.1 The system shall enforce strong password policies during updates.

6.2 Profile updates must be logged for security and audit purposes.

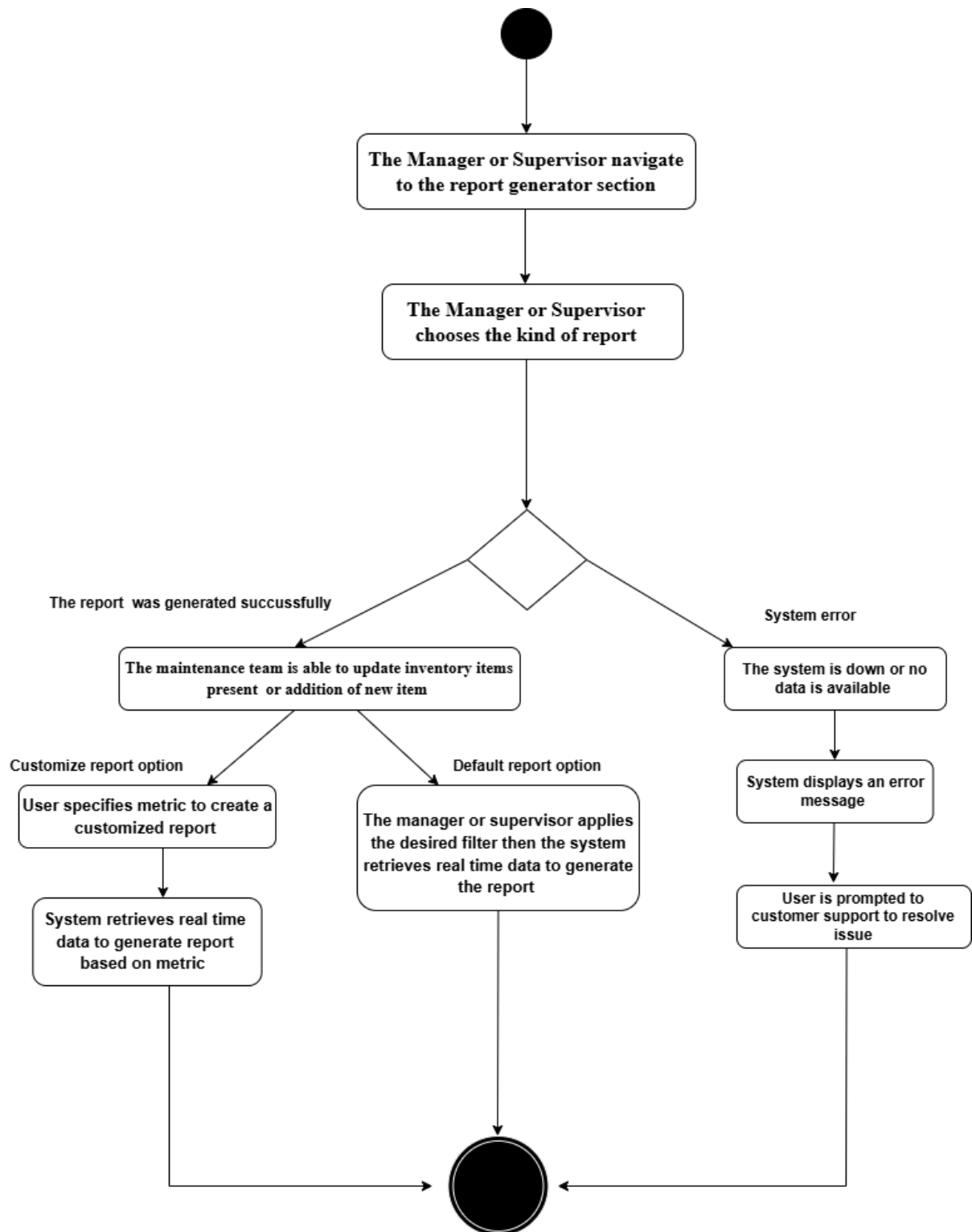
6.3 The system shall support real-time validations for fields like email and phone numbers.

7. Extension Points

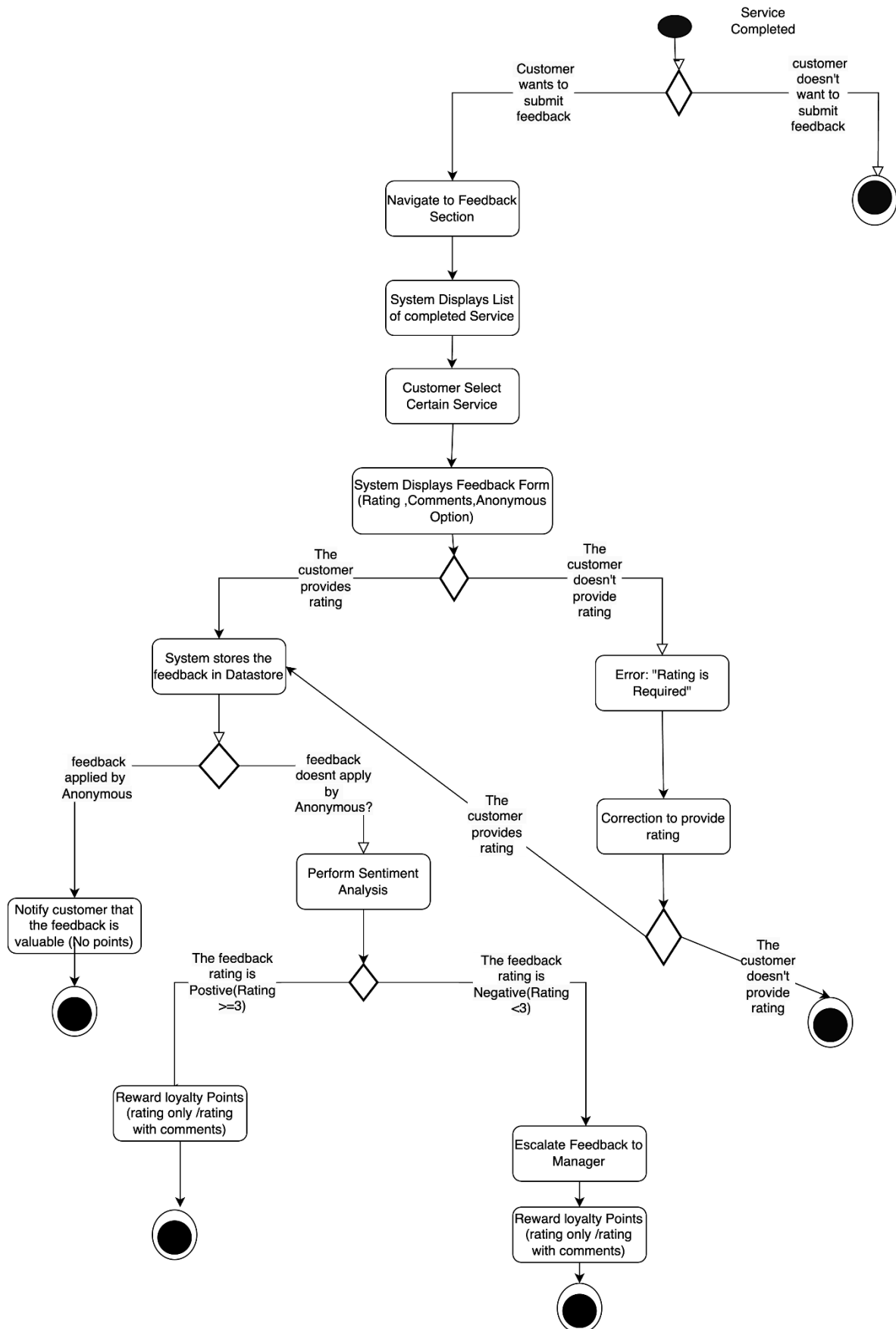
7.1 None identified.

Task 3.4:

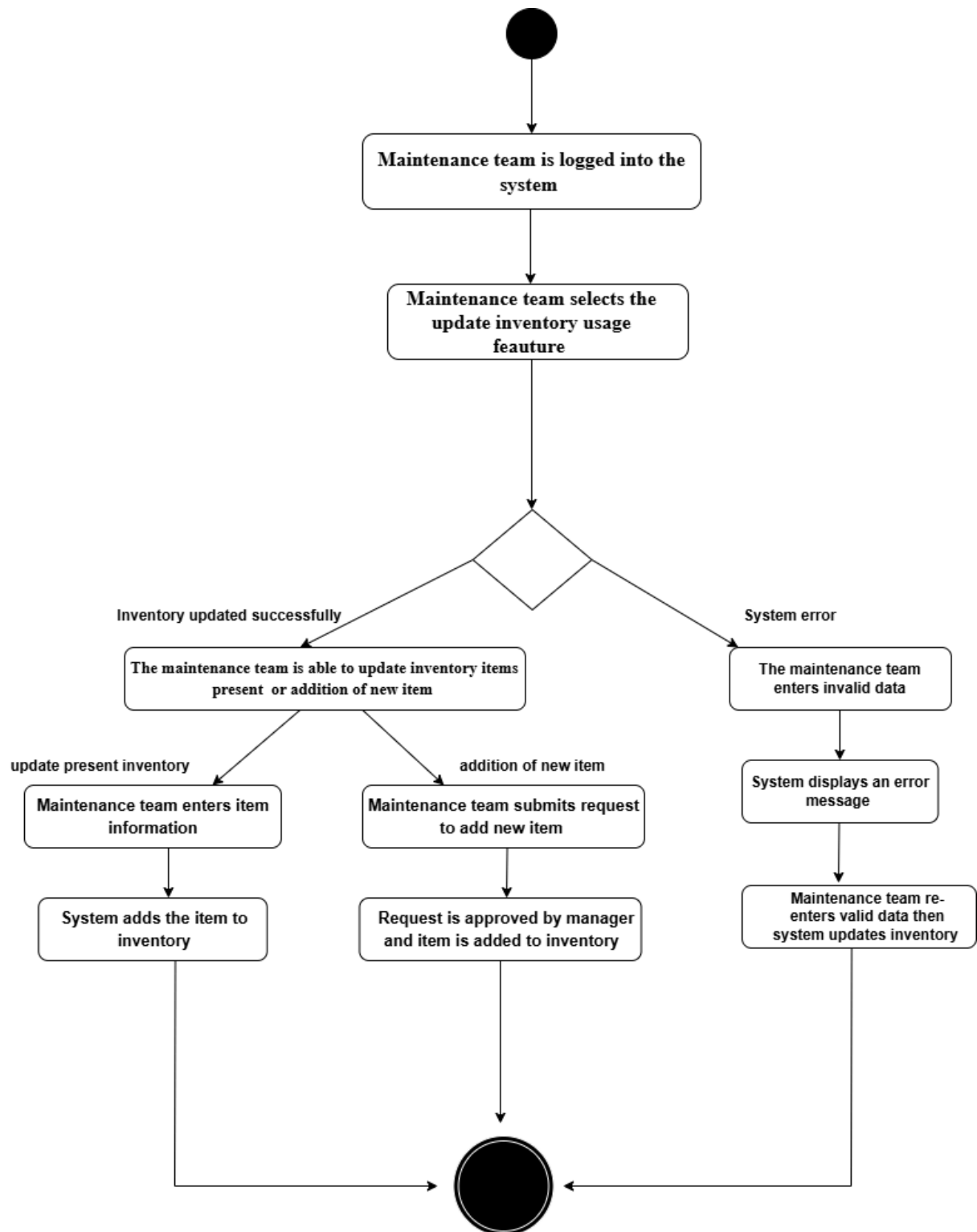
Deema:



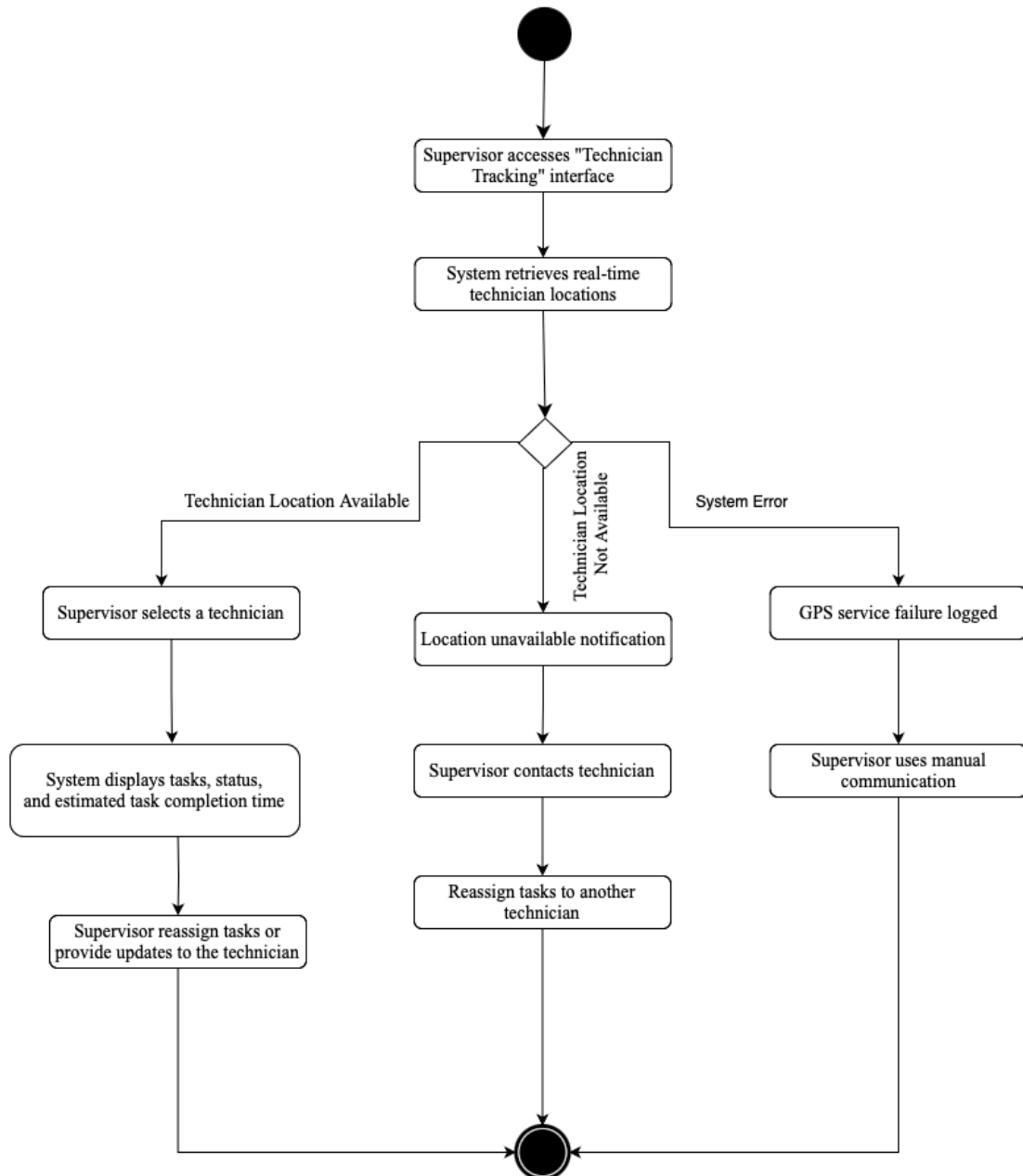
Mustafa:



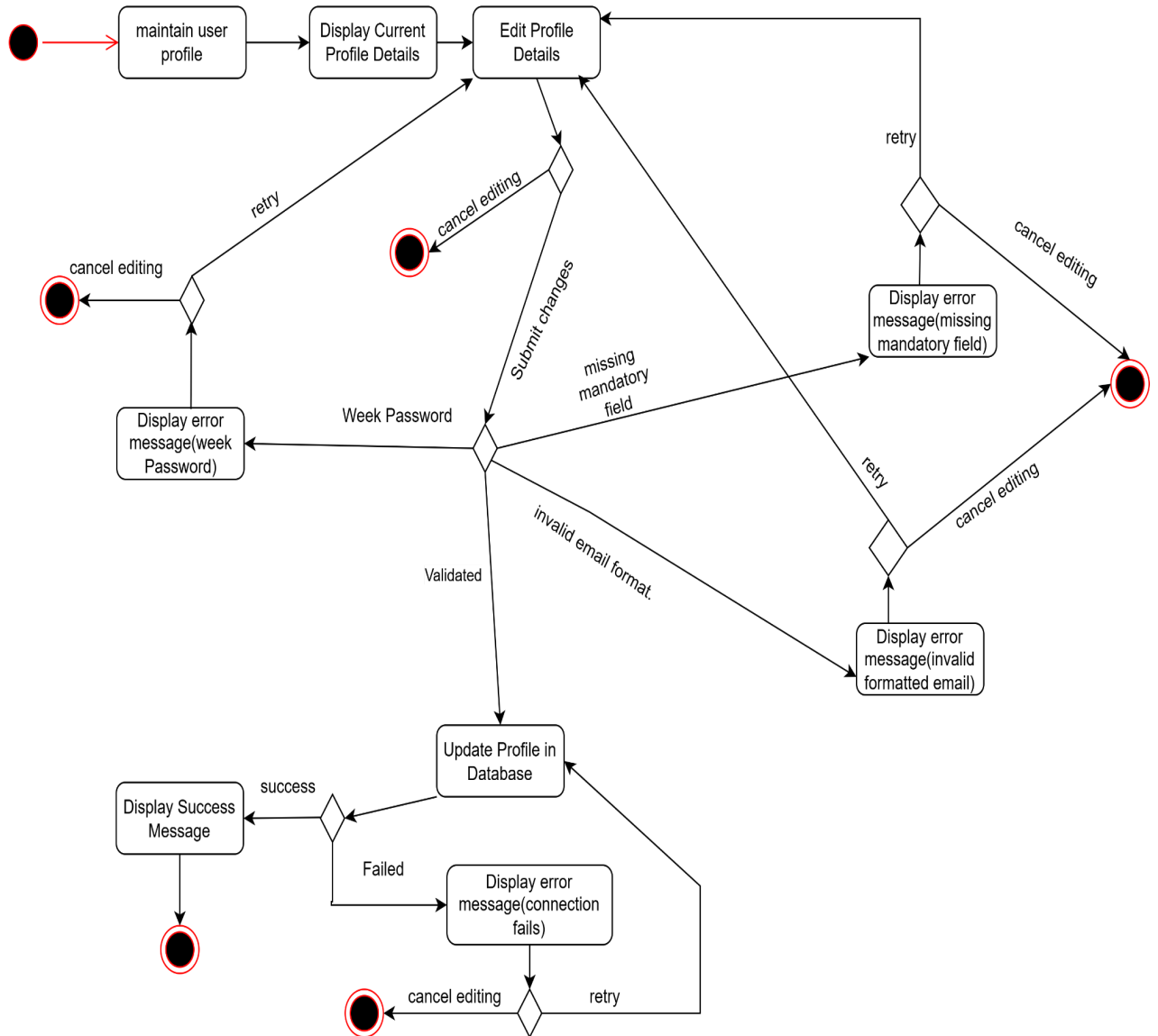
Aseel



Fadi:



Amro:



Phase 4 :

Task 4.1:

Sahab systems classes:

1. User
2. Customer
3. Technician
4. Maintenance Member
5. Service Request
6. Task
7. Inventory Item
8. Feedback
9. Report
10. Notification
11. Communication
12. Schedule
13. GPS
14. AI
15. Database Management System
16. Manager
17. Supervisor
18. Customer with disabilities
19. UI Controller

Analysis of Sahab System Classes:

1. User:

The user class captures the most important traits that signify a user regardless of their position through defining basic information like their name, ID, email, password, role, phone number, address, date of birth (aka" DOB), and location in order to avoid redundancy throughout classes. Alongside traits, there are also operations that user can perform, like logging in and updating profiles using their email and password for authentication, whether its for simple access or updates, and lastly a logging-out operation to exit the system when necessary.

User Attributes:

1. Id : String
2. Name: String
3. Email: String
4. Password: String
5. Role: String
6. Phone number: String
7. Address: String
8. DOB: Date
9. Location: Coordinates

User Operations:

1. Login(Email: String , Password: String): Boolean
2. Logout():Void
3. UpdateProfile(Email:String , Password:String):Boolean

2. Customer:

The Customer class extends the User class by inheriting its basic traits like name, ID, email, password, and location while introducing additional features suitable for customer activities. Activities include keeping track of the customer's service history, a list of service requests, and rewarding their engagement through loyalty points. Along with these attributes, the Customer class provides operations that allow the customer to interact with the system. They can submit service requests based on their needs, provide feedback on completed services, check the status of their active requests, and view their list of completed services. These features ensure the system is user-friendly and efficient for managing customer interactions.

Customer Attributes:

1. ServiceHistory: List<ServiceRequest>
2. LoyaltyPoints: Integer

Customer Operations:

1. SubmitServiceRequest(Description: String, Urgency: String, Location: String): ServiceRequest
2. SubmitFeedback(ServiceId: String, Rating: Integer, Comments: String, AnonymousFlag: Boolean): Feedback
3. ViewServiceStatus(ServiceId: String): String
4. getCompletedServices(): List<ServiceRequest>

3. Technician:

The Technician class inherits from User, capturing the essential attributes such as name, ID, email, and location while focusing on operations related to managing tasks and inventory. Technicians are responsible for accepting tasks, updating their progress, and reporting inventory usage to ensure tasks are completed efficiently and resources are tracked accurately.

Technician Attributes:

1. assignedTasks: List <Task>
2. availability: Boolean
3. schedule: Schedule

Technician Operations:

1. AcceptTask(TaskId: String): Boolean
2. UpdateTaskStatus(TaskId: String, NewStatus: String): Boolean
3. ReportUsage(ItemId: String, QuantityUsed: Integer): Boolean

4. Maintenance Member:

The Maintenance class inherits from the User class, capturing core traits like name, ID, and location, while introducing additional attributes and operations related to inventory through the assigned inventory attribute. Maintenance members are responsible for updating stock levels, validating inventory, notifying managers about low stock, and generating restock reports, ensuring that inventory remains sufficient and operational requirements are met.

Maintenance Member Attributes:

- 1. Responsibilities: List<Task>**
- 2. AssignedInventory: List<InventoryItem>**
- 3. Notifications: List<Notification>**
- 4. schedule: Schedule**

Maintenance Member Operations:

- 1. UpdateStock(item: InventoryItem, NewQuantity: Integer): Boolean**
- 2. ValidateInventory(ItemId: String): Boolean**
- 3. ReceiveNotification(notification: Notification): Void**
- 4. GenerateRestockReport(): Report**

5.Service Request:

The Service Request class is used to manage customer service requests by upholding attributes that store all the important details of a request. These attributes include a RequestID, the ID of the customer who created the request, which is known as CustomerID, a description of the issue named Description, the urgency level known as urgency, the status of the request known as Status, and location. These attributes ensure that every request is tracked and handled appropriately. Operations for this class allow creating new requests, updating existing ones by entering the attribute that's to be updated and the new value, and canceling them when needed. This functionality ensures the system meets customer needs.

Service Request Attributes:

1. **ServiceRequestID: String**
2. **description: String**
3. **urgency: String**
4. **status: String**
5. **location: String**
6. **creationDate: Date**

Service Request Operations:

1. **CreateServiceRequest(customerId: String, description: String, urgency: String, location: String): ServiceRequest**
2. **UpdateServiceRequest(requestId: String, updatedDetails: Map<String, String>): Boolean**
3. **CancelServiceRequest(ServiceRequestID: String): Boolean**

6. Task:

The Task class is responsible for managing tasks within the system and assigning them to technicians. It tracks essential details about the task, including the task ID, the task description, the technicians assigned, and its current status. The Task class also provides operations for managing task assignments and updating task statuses, ensuring that tasks are handled properly.

Task Attributes:

1. ID: String
2. TaskDetails: String
3. AssignedTechnician: List<Technician>
4. Status: String

Task Operations:

1. assignTechnician(taskId: String, technicianId: String): Boolean
2. updateStatus(taskId: String, newStatus: String): Boolean

7. Inventory Item:

The Inventory Item class is used to manage the items in inventory. It holds details about each item, including its ID, name, quantity, location, and whether a low-stock alert is triggered (determined by AI). It also includes operations for updating stock quantities, ensuring that inventory is properly maintained.

Inventory Item Attributes:

1. **ItemId: String**
2. **Name: String**
3. **Quantity: Integer**
4. **Location: String**
5. **LowStockAlert: Boolean**

Inventory Item Operations:

1. **updateStock(newQuantity: Integer): Boolean**

8. Feedback:

The Feedback class uses feedback provided by customers about completed services. It contains important details such as feedback ID, rating, comments, and whether the feedback is anonymous. The Feedback class provides operations for submitting and validating feedback, helping maintain service quality.

Feedback Item Attributes:

- 1. FeedbackId: String**
- 2. Rating: Integer**
- 3. Comments: String**
- 4. AnonymousFlag: Boolean**

Feedback Item Operations:

- 1. CreateFeedback(serviceId: String, customerId: String, rating: Integer, comments: String, anonymousFlag: Boolean): Boolean**
- 2. validateFeedback(serviceId: String, rating: Integer): Boolean**

9. Report:

The Report class is used to generate and manage reports within the system, providing valuable insights into various aspects of operations. It holds attributes such as report ID, export type (e.g., Excel, PDF, Google Sheets), the person who generated the report, and the date it was created. Reports can be customized using a wide range of filtration criteria (the filtration will be based on two metrics, the first one being the object type that the report is created for and the second filter will be for the attribute for that object that we want to base the report off of), including date range, export format, and status of tasks or services. Additional filters allow users to refine reports by technician details (e.g., assigned tasks, workload, or performance), customer-specific information (e.g., service requests, feedback ratings), task attributes (e.g., priority, type, or location), and inventory-related details (e.g., item stock levels, location, or usage trends).

Report Attributes:

1. **ReportId: String**
2. **ExportType: String (Details: Excel, Google Sheets, PDF)**
3. **GeneratedBy: String**
4. **DateGenerated: Date**

Report Operations:

1. **generateReport(exportType: String, filters: Map<String, String>): Report**
2. **filterReport(filters: Map<String, String>): Report**

10. Notification:

The Notification class allows the system to send notifications to users. It tracks the notification ID, message content, recipient, and the timestamp of when the notification was sent. Operations within this class enable the system to send notifications to users.

Notification Attributes:

- 1. NotificationId: String**
- 2. Message: String**
- 3. Recipient: User**
- 4. Timestamp: Date**

Notification Operations:

- 1. SendNotification(recipientId: String, message: String): Boolean**

11. Communication:

The Communication class manages messages between users. It includes the message ID, sender, receiver, content, and timestamp. The class supports sending, receiving, and logging messages for record-keeping.

Communication Attributes:

- 1. MessageId: String**
- 2. Sender: User**
- 3. Receiver: User**
- 4. Content: String**
- 5. Timestamp: Date**

Communication Operations:

- 1. sendMessage(senderId: String, receiverId: String, content: String): Boolean**
- 2. receiveMessage(messageId: String): String**
- 3. logMessage(messageId: String): Void**

12. Schedule:

The Schedule class manages task assignments, including task scheduling for technicians and maintenance members. It holds attributes like the task ID, role of the user assigned to the task, start and end times, status, priority, and location. The class offers operations to update times, reassign users, change task status, and check for overdue tasks.

Schedule Attributes:

- 1. ScheduleId: String**
- 2. Task: Task**
- 3. Role: User (Technician or Maintenance Member)**
- 4. StartTime: DateTime**
- 5. EndTime: DateTime**
- 6. Status: String**
- 7. Priority: Integer**
- 8. Location: String**

Schedule Operations:

- 1. updateStartTime(newStartTime: DateTime): Boolean**
- 2. updateEndTime(newEndTime: DateTime): Boolean**
- 3. changeUser(newUser: User): Boolean**
- 4. updateStatus(newStatus: String): Boolean**
- 5. isOverdue(currentTime: DateTime): Boolean**

13. GPS:

The GPS System class is used to track technicians' locations and optimize routes for task assignments. It manages location data and provides operations for tracking technicians and calculating the best route between two locations.

GPS Attributes:

- 1. LocationData: Coordinates**

GPS Operations:

- 1. TrackTechnician(technicianId: String): Coordinates**
- 2. OptimizeRoute(startLocation: Coordinates, endLocation: Coordinates): Route**

14. AI:

The AI Class is responsible for handling predictive tasks and analyzing historical data to provide insights and recommendations. It includes training data, the prediction model, and a time period for predictions. The class offers operations to predict restocking quantities, analyze consumption patterns, evaluate technician performance, and more. It leverages AI algorithms to improve operational efficiency and optimize resource management.

AI Attributes:

- 1. TrainingData: List<DataSet>**
- 2. PredictionModel: String**
- 3. SmartPredictionPeriod: Integer**

AI Operations:

- 1. predictRestockingQuantity(items: List<InventoryItem>, timePeriod: Integer): void**
- 2. analyzeConsumptionPatterns(userId: String, startDate: Date, endDate: Date): Map<String, Integer>**
- 3. detectSeasonalTrends(itemId: String): Boolean**
- 4. evaluateTechnicianPerformance(technicianId: String): Map<String, Object>**
- 5. generateOperationalReport(): Report**
- 6. allocateTasksBasedOnPerformance(tasks: List<Task>, users: List<User>): Map<Task, User>**
- 7. predictWorkload(users: List<User>, timePeriod: Integer): Map<User, Integer>**
- 8. recommendScheduleAdjustments(tasks: List<Task>, schedule: List<Schedule>): List<Schedule>**
- 9. analyzeSentiment(feedbackId: String): String**

15. DBMS:

The DBMS class is designed to manage and interact with the system's database. It includes operations to execute queries, back up the database, restore it, and import data from external sources. It also logs executed queries and manages connections to ensure smooth database operations.

DBMS Attributes:

- 1. ConnectionInfo: List<Connection>**
- 2. QueryLogs: List<String>**
- 3. BackupId: String**
- 4. DataSources: List<String>**

DBMS Operations:

- 1. executeQuery(query: String): ResultSet**
- 2. backupDatabase(): Boolean**
- 3. restoreDatabase(backupId: String): Boolean**
- 4. importData(sourceType: String, filePath: String): Boolean**
- 5. processUnstructuredData(data: String): Map<String, Object>**

16. Manager:

The Manager oversees the overall operations of the system, reviewing reports and ensuring tasks are properly managed. The Manager is responsible for approving schedules, generating reports, assigning supervisors, and reviewing customer feedback to improve service quality.

Manager Attributes:

- 1. AssignedReports: List<Report>**
- 2. Notifications: List<Notification>**

Manager Operations:

- 1. approveSchedule(scheduleId: String): Boolean**
- 2. generateReport(reportType: String): Report**
- 3. assignSupervisor(supervisorId: String, taskId: String): Boolean**
- 4. reviewFeedback(feedbackId: String): String**

17. Supervisor:

The Supervisor is responsible for overseeing tasks and ensuring their successful completion. They assign technicians to tasks, monitor the task status, and ensure the progress of each task. The Supervisor can also send notifications and update task statuses as necessary.

Supervisor Attributes:

- 1. AssignedTasks: List<Task>**

Supervisor Operations:

- 1. monitorTask(taskId: String): String assignTechnician(taskId: String, technicianId: String): Boolean**
- 2. updateTaskStatus(taskId: String, newStatus: String): Boolean**
- 3. sendNotification(userId: String, message: String): Boolean**

18. Customer with Disabilities:

The Customer with Disabilities class is a specialized subclass of the Customer class, designed to accommodate accessibility needs. It includes additional attributes for the type of disability, preferred assistance mode, and a list of enabled accessibility features. The class provides operations for requesting assistance, enabling accessibility features, and submitting feedback with accessibility features highlighted.

Customer with disabilities Attributes:

- 1. DisabilityType: String**
- 2. PreferredAssistanceMode: String**
- 3. AccessibilityFeatures: List<String>**

Customer with disabilities Operations:

- 1. requestAssistance(assistanceType: String): Boolean**
- 2. enableAccessibilityFeature(feature: String): Boolean**
- 3. provideFeedbackWithAccessibility(featuresUsed: List<String>, feedback: String): Feedback**

19. UIManager Controller:

The UIManager acts as the central controller for managing the user interface, coordinating navigation, forms, feedback, notifications, and accessibility features.

UIManager Controller Attributes:

- 1. currentScreen: String**
- 2. theme: String**
- 3. language: String**
- 4. notifications: List<Notification>**
- 5. accessibilitySettings: Map<String, Boolean>**
- 6. activeUser: User**

UIManager Controller Operations:

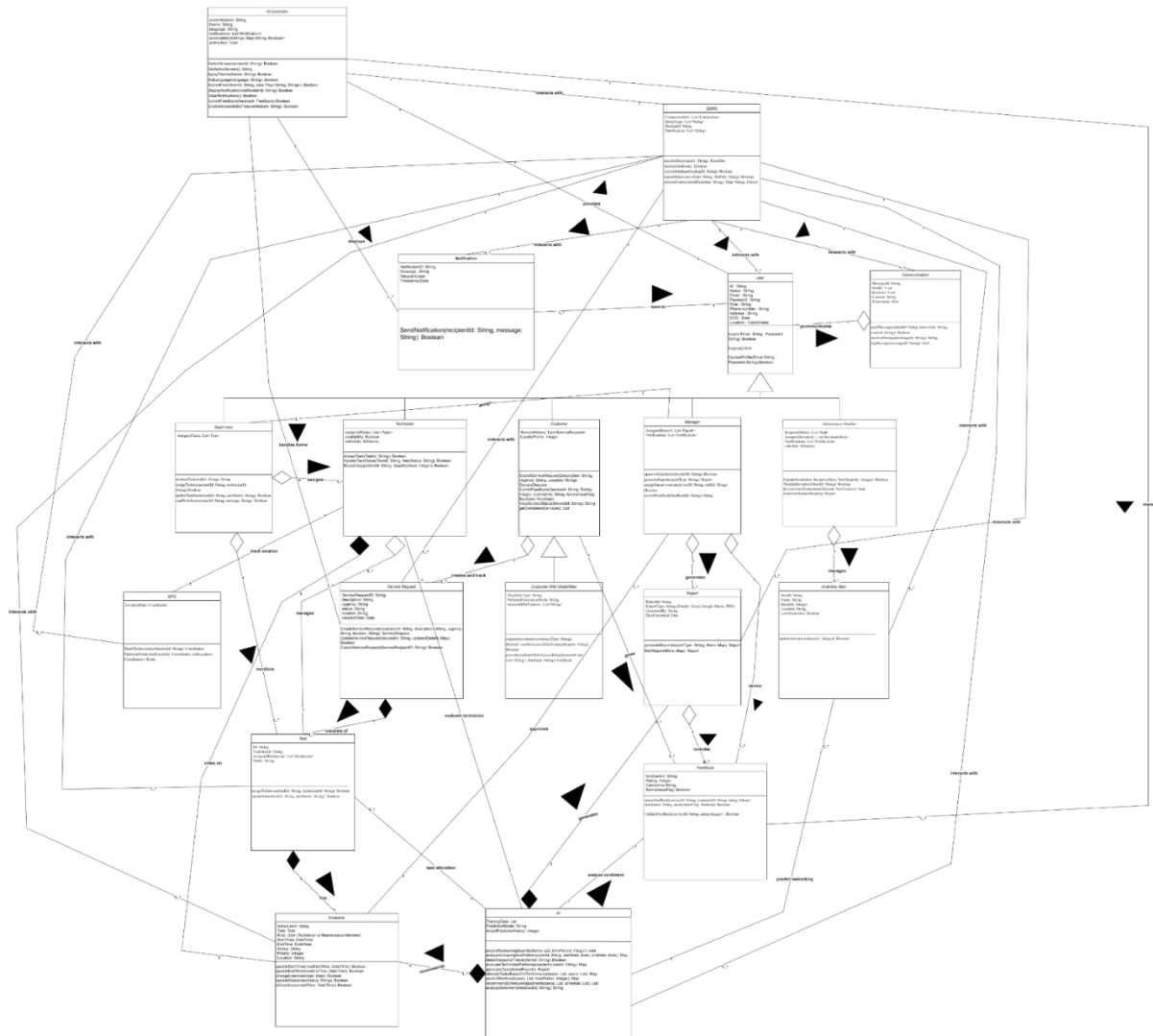
- 1. SwitchScreen(screenId: String): Boolean**
- 2. GetActiveScreen(): String**
- 3. ApplyTheme(theme: String): Boolean**
- 4. SetLanguage(language: String): Boolean - Changes the language for the UI.**
- 5. SubmitForm(formId: String, data: Map<String, String>): Boolean**
- 6. DisplayNotification(notificationId: String): Boolean**
- 7. ClearNotifications(): Boolean**
- 8. SubmitFeedback(feedback: Feedback): Boolean**
- 9. EnableAccessibilityFeature(feature: String): Boolean**

Class Diagram :

Link :

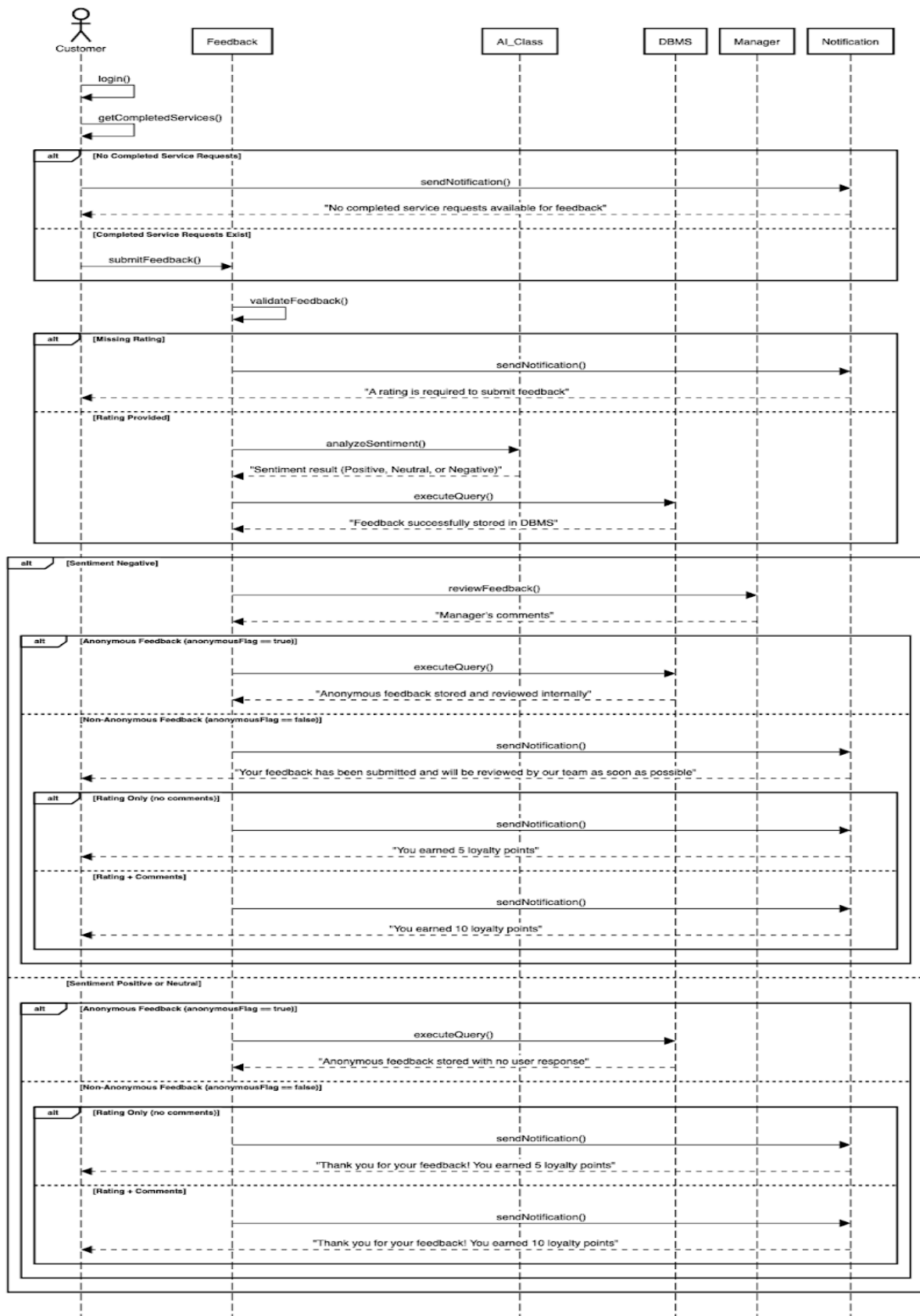
<https://drive.google.com/file/d/14YDXph9UrZ4fLpWmcgr2GmPwlpG9EYKe/view?usp=sharing>

g

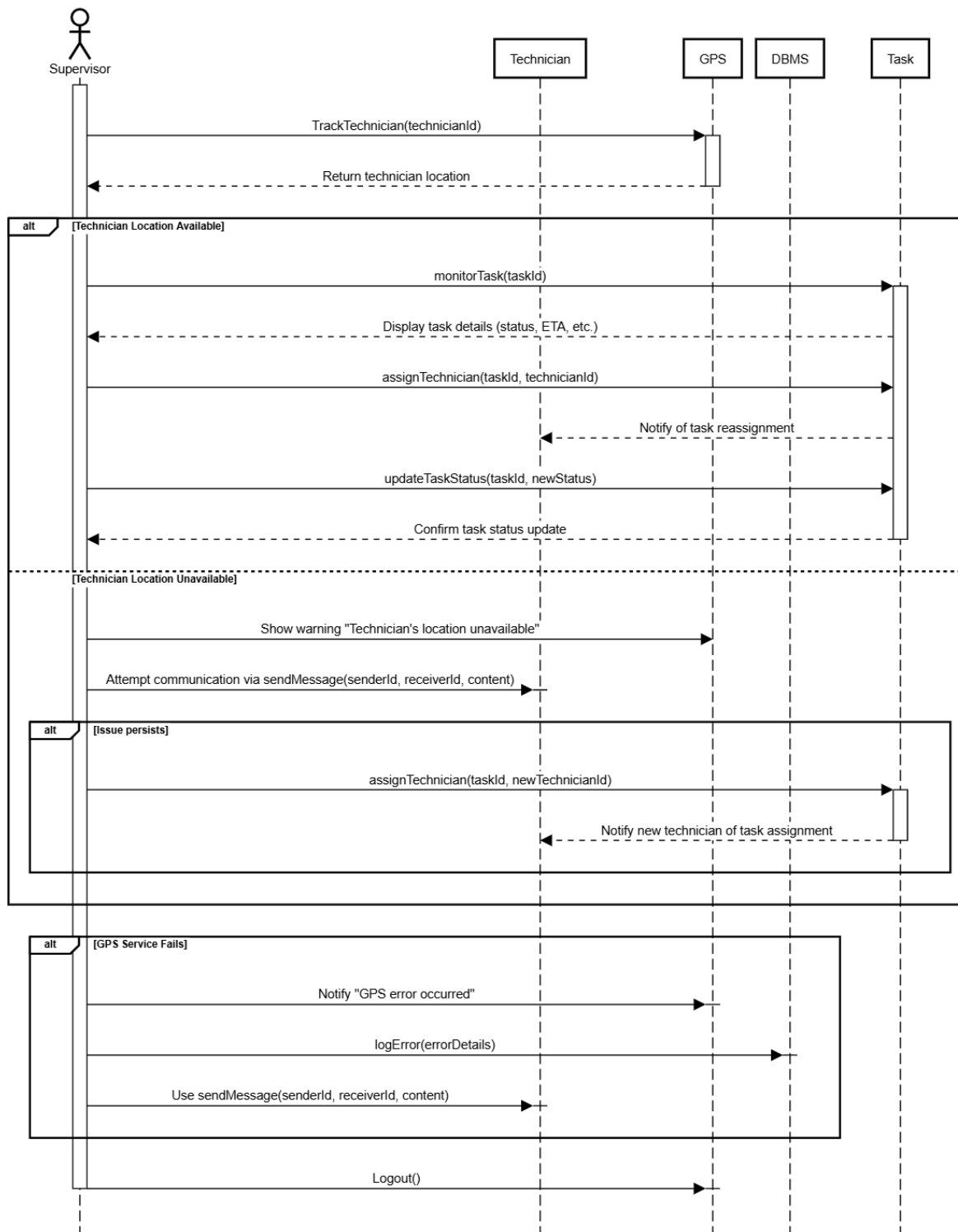


Task 4.2:

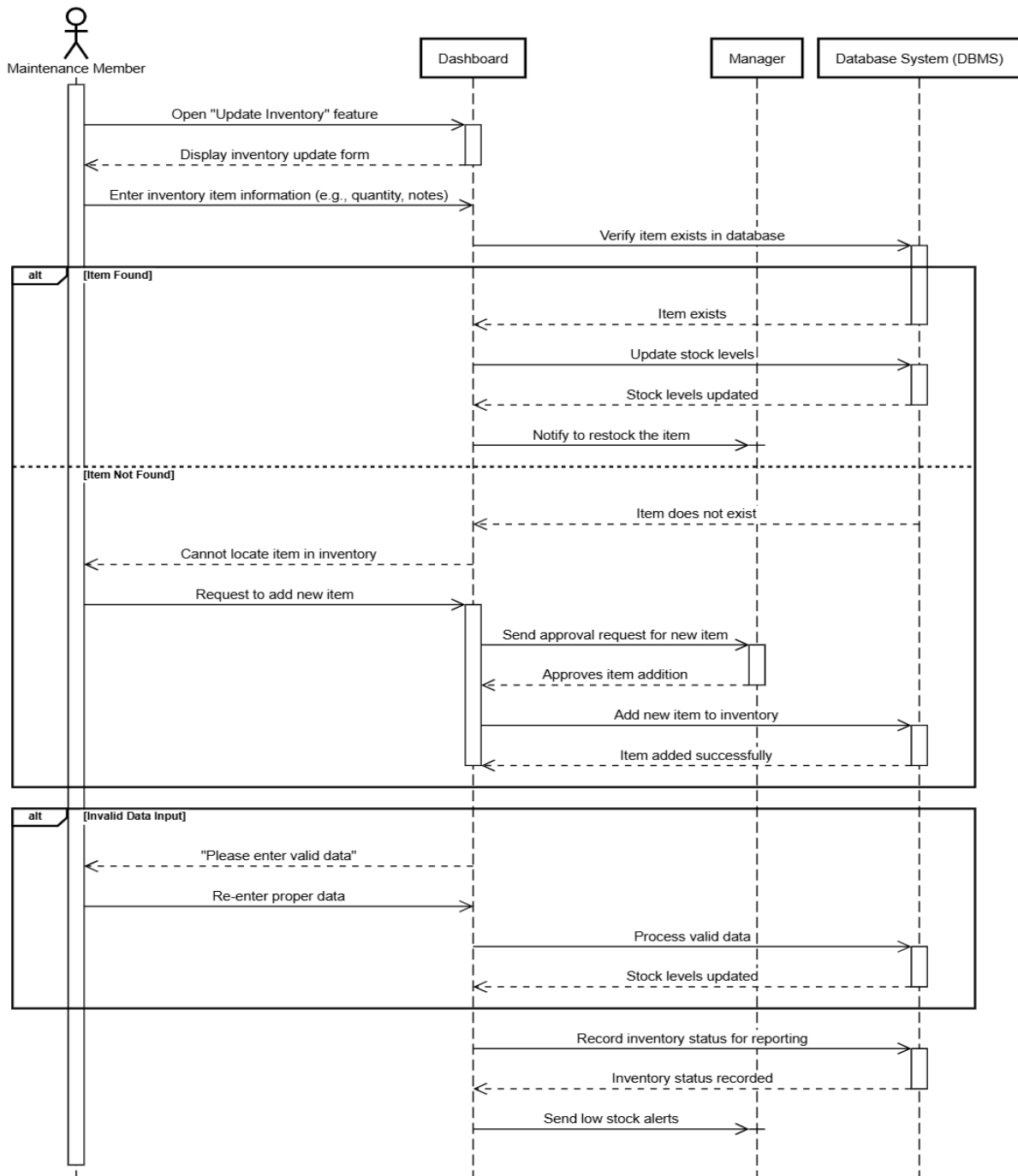
Mustafa :



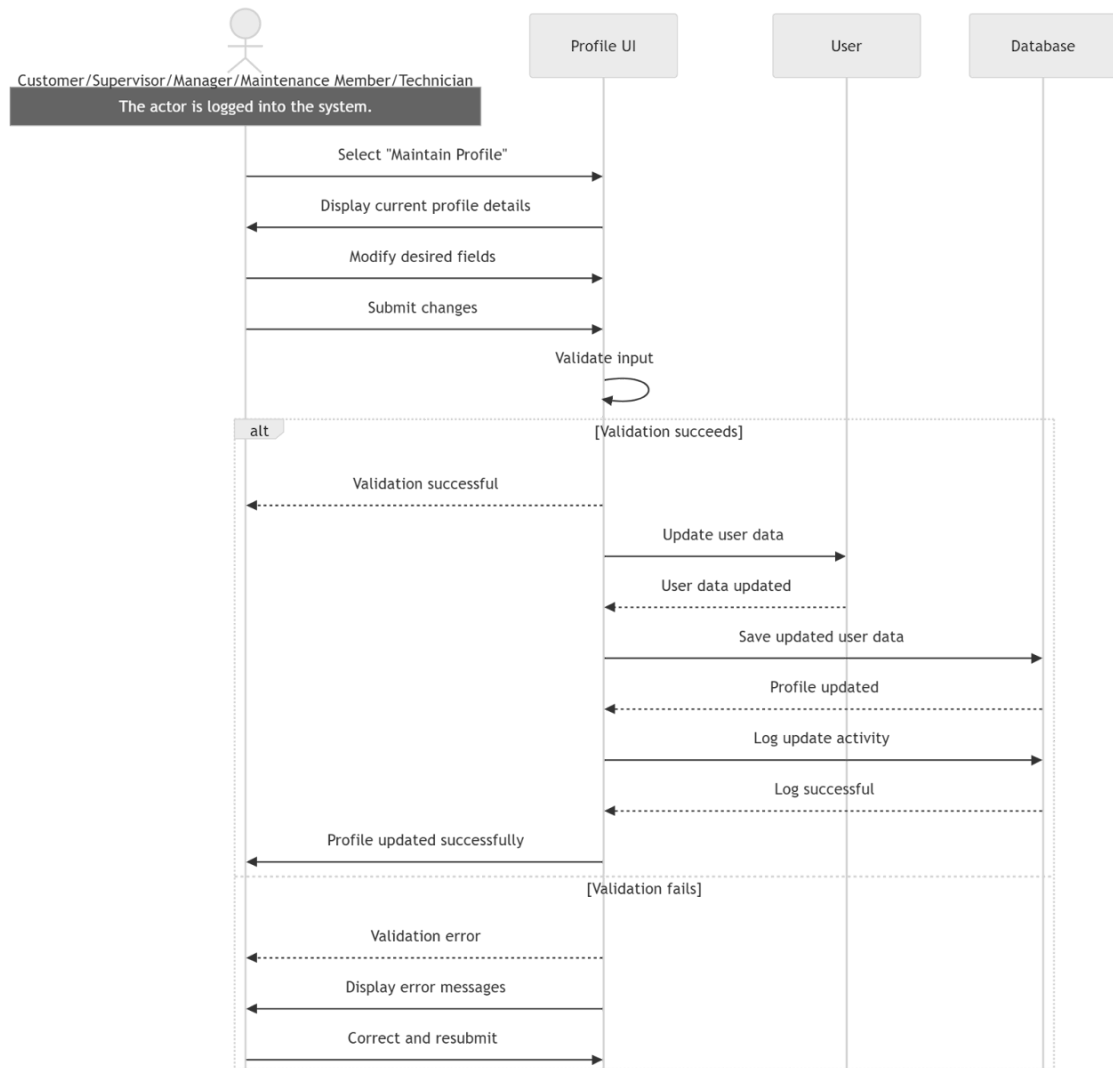
Fadi:



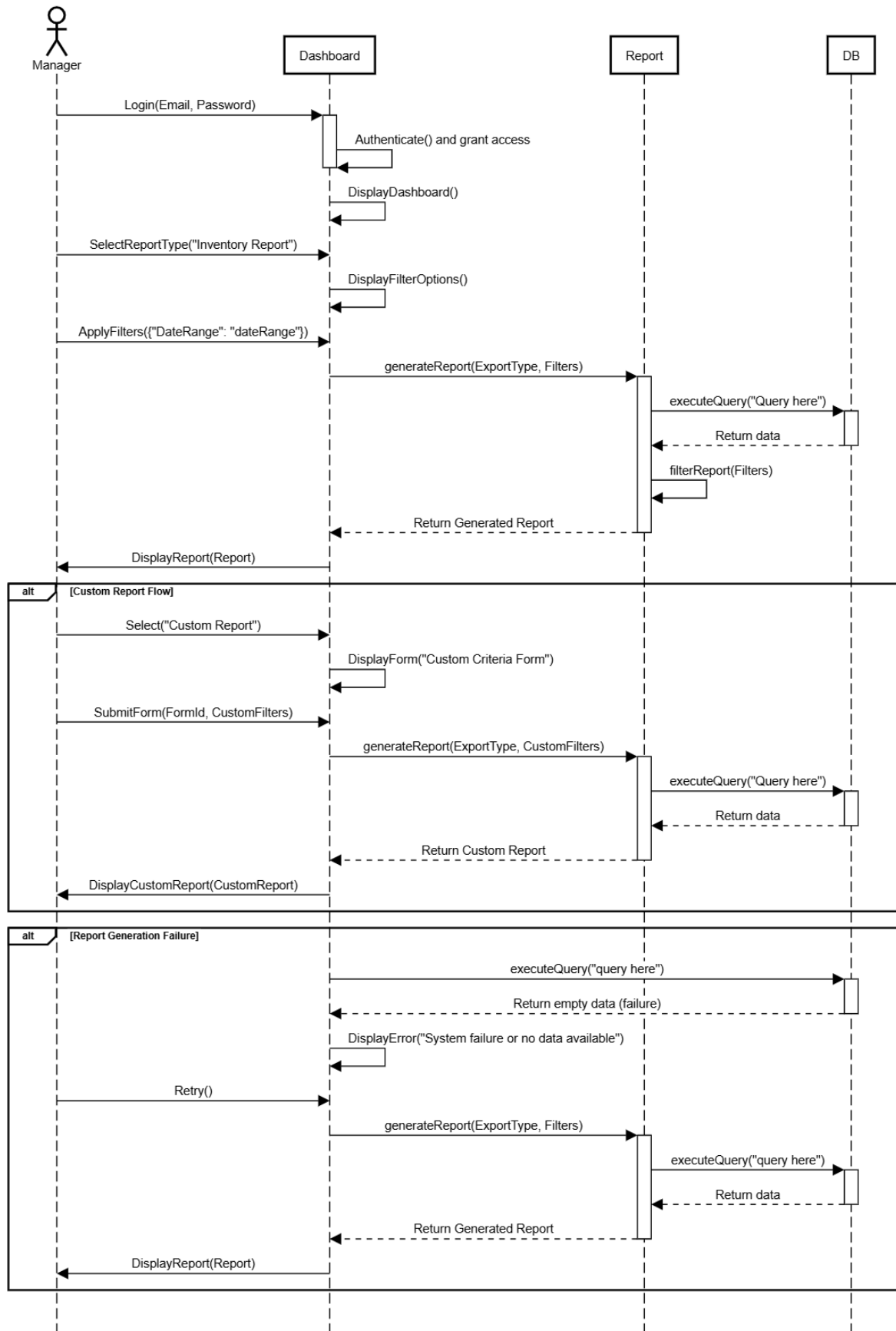
Aseel:



Amro:



Deema:



Task 4.3:

Sahab System Design Goals:

General Design Goals:

1. Low Coupling:

Low coupling essentially ensures that the various components of the Sahab system are only minimally dependent on one another. This means that the process of making changes in a specific part of the system becomes significantly easier to manage, as there's less concern about potential issues arising in other areas. As a result, it helps ensure that adjustments and modifications can proceed much more smoothly overall.

2. High Cohesion:

High cohesion is incredibly important to make sure that components within the Sahab system are greatly related and focused on certain tasks and have clear and known responsibilities. Each component should be able to perform its functions, which in turn allows for maintainability and lowers the chances of ambiguity.

What cohesion can look like in Sahab Systems:

- The Feedback Class is responsible for handling customer reviews, ratings, and loyalty points. It does not manage tasks outside of the feedback scope, like inventory updates or technician scheduling. This ensures that feedback-related processes, such as sentiment analysis and escalation of negative feedback to the manager are properly taken care of within the component in charge.

Specific Design Goal:

- **Robust Data Security and Privacy**

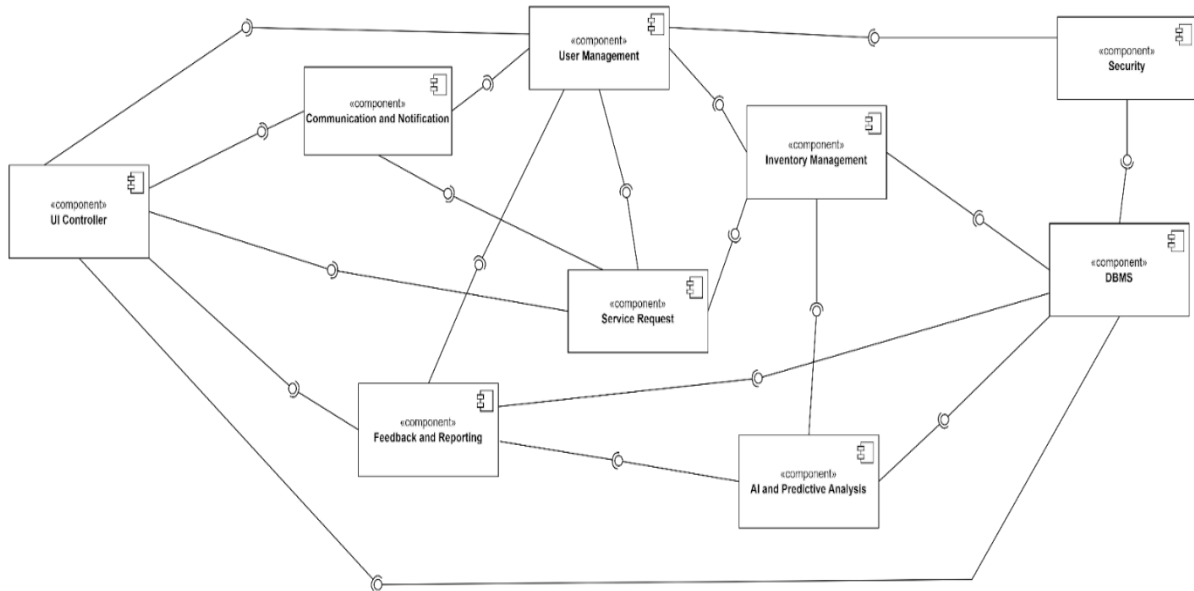
At Sahab Systems, we place the highest priority on the protection of sensitive customer and employee data. As part of our commitment to ensuring robust data privacy, the system must fully comply with established industry standards and regulations. This includes not only adhering to these guidelines but also implementing advanced security measures to safeguard against unauthorized access, data breaches, and other potential threats. We understand the critical importance of maintaining confidentiality and integrity in all interactions, and as such, security is deeply embedded in our system's architecture and processes.

- **Examples:**

- 1. End-to-End Encryption:** The system will use encryption to protect sensitive data both in transit and at rest, ensuring secure handling of all information.
- 2. Two-Factor Authentication:** Sahab Systems will require two-factor authentication for all users, adding an extra layer of security to prevent unauthorized access.
- 3. Regular Audits and Logging:** The system will conduct regular security audits and maintain access logs to detect and address potential security issues.

Task 4.4:

Component Diagram:



Task 4.5:

Deployment Diagram

