

Deliverable 2

Phantom Syndicates

2025-04-11

Contents

Problem Statement	2
Problems	2
Proposed Solutions	2
Smart Inventory Management System:	2
Login and Register Subsystem	3
Use Case: Register	3
Use Case: Log in	4
Use Case: Reset Password	5
Use Case: Send Confirmation Email	6
Use Case: Send Reset Link	7
Repairs and damages Subsystem	10
Use Case: View Suppliers Returns	10
Use Case: Add New Returns	11
Use Case: View Customer Repairs	13
Use Case: Add New Repair	14
Use Case: Notify Customer	15
Order Management Subsystem	17
Use case: View orders	17
Use case: Approve orders	18
Use case: Reject orders	19
Use case: Add purchase order	20
Use case: Set Re-order points	20
Supplier Management Subsystem	21
Use case: View Suppliers	22
Use case: Add supplier	22
Use case: Edit supplier	23
Use case: Remove supplier	24
Warehouse Management Subsystem	25
Use case: View warehouses	26
Use case: Add warehouse	26
Use case: Edit warehouse	27
Use case: Remove warehouse	28

Export documents subsystem	29
Use case: Generate Stock Reports	29
Use case: Export Document	29
Inventory Management Subsystem	30
Use Case: View Inventory	30
Use Case: Add product/Spare part	31
Use Case: : Remove product/spare part	32
Use Case: Edit product/spare part	33
Use Case: Scan product	33
Use Case: Manually Update product information	35
Database Design	37

Problem Statement

Problems

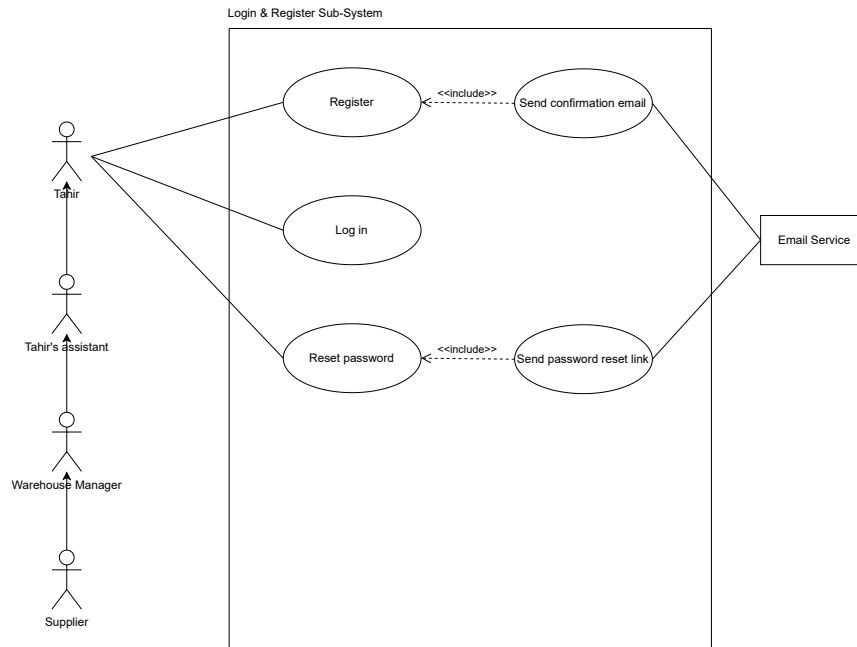
- There is lack of real-time inventory tracking. Where stock can be tracked in a faster way.
- Data entry in Excel spreadsheets and paper records increases the risk of errors, such as incorrect stock counts or misplaced invoices
- Inefficient order management.
- Difficulty managing spare parts.
- increased operational costs
- There is no analysis and reports about the stock.

Proposed Solutions

Smart Inventory Management System:

- To address these challenges, we propose the development of a Smart Inventory Management System for MTM Distributors. This is a software tool that helps businesses to keep track of their stock by providing real-time updates on stock levels. Invoices will be generated automatically within the system. Spare parts will be tracked digitally, making it easy to search for and update information. This system helps to manage sales and purchase orders and can suggest when to reorder . The system will automatically generate reports based on stock levels.
- This system will make use of a mobile app used by the staff on-the-go to do stock updates , receiving notifications and also for task management.

Login and Register Subsystem



Use Case: Register

Inputs: User credentials (Email, password, role, name, reg key).

Outputs: New user account, confirmation email

Preconditions: User is not registered.

Basic Path:

1. User accesses the login page. 2. User selects "Register." 3. System displays registration form (email, password, role, name, regKey) 4. User fills in details and submits. 5. System validates data and creates an account. 6. System sends confirmation email 7. System confirms successful registration and redirects the user to the log in page.

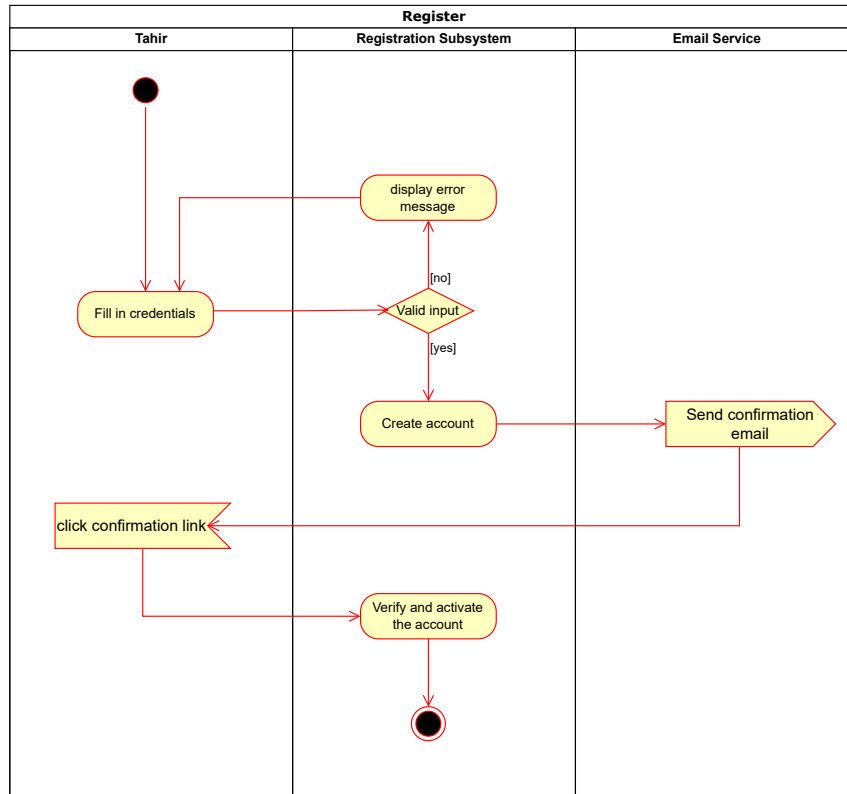
Extensions:

5a. Invalid data.

1. System prompts user to correct input errors.

5b. Email already exists.

1. System notifies user and suggests login.



Use Case: Log in

Inputs: User credentials (Email, password).

Outputs: Access to dashboard, session token.

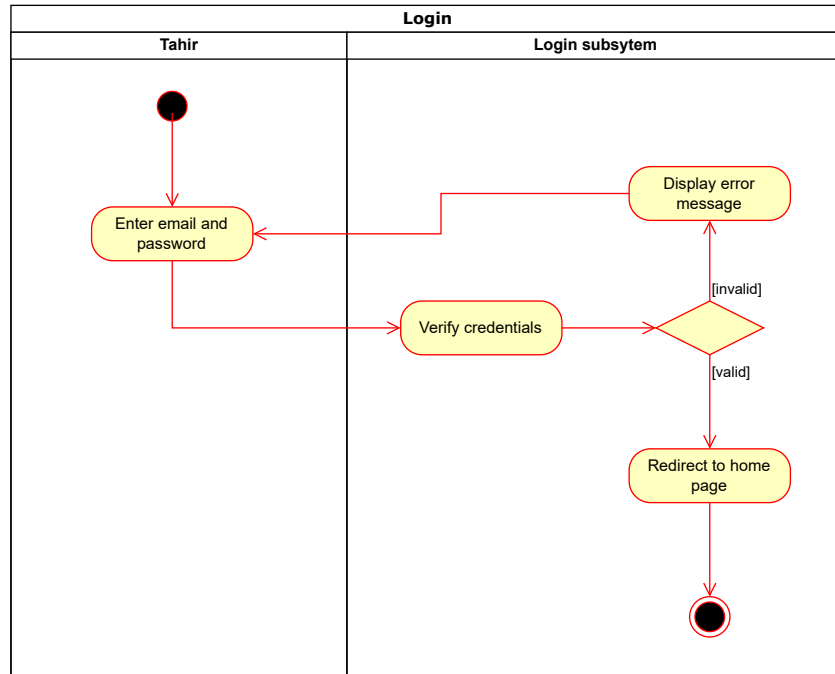
Preconditions: User has a registered account.

Basic Path:

1. User selects “Log in”. 2. System prompts for email and password. 3. User enters credentials. 4. System verifies credentials. 5. System grants access based on role and redirects to the home page.

Extensions:

- 4a. Incorrect credentials
1. System displays error and allows retry.



Use Case: Reset Password

Inputs: Registered email, new password.

Outputs: Password reset confirmation.

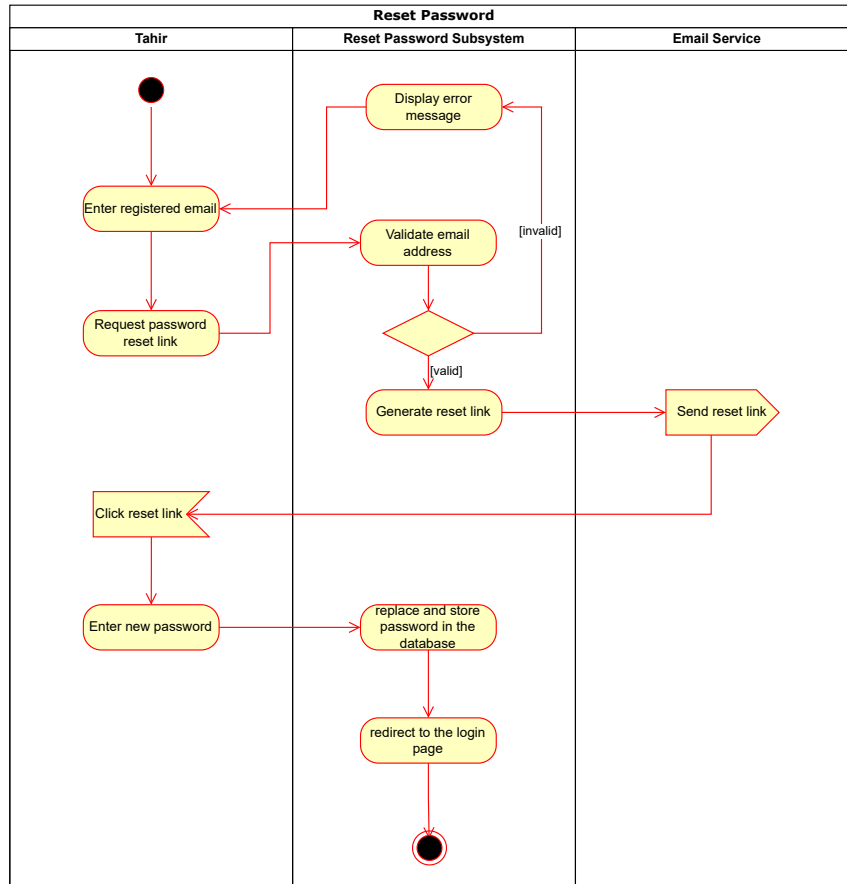
Preconditions: User is registered but cannot log in.

Basic Path:

1. User selects the “Forgot Password” option on the login page.
2. System prompts for registered email.
3. User enters and submit the email.
4. System sends reset link to email.
5. User follows link and sets new password.
6. System confirms password update.

Extensions:

- 3a. Email not found
 1. System notifies user.
- 5a. Expired link
 1. System regenerates reset link.



Use Case: Send Confirmation Email

Inputs: User email.

Outputs: Email sent status, user notification.

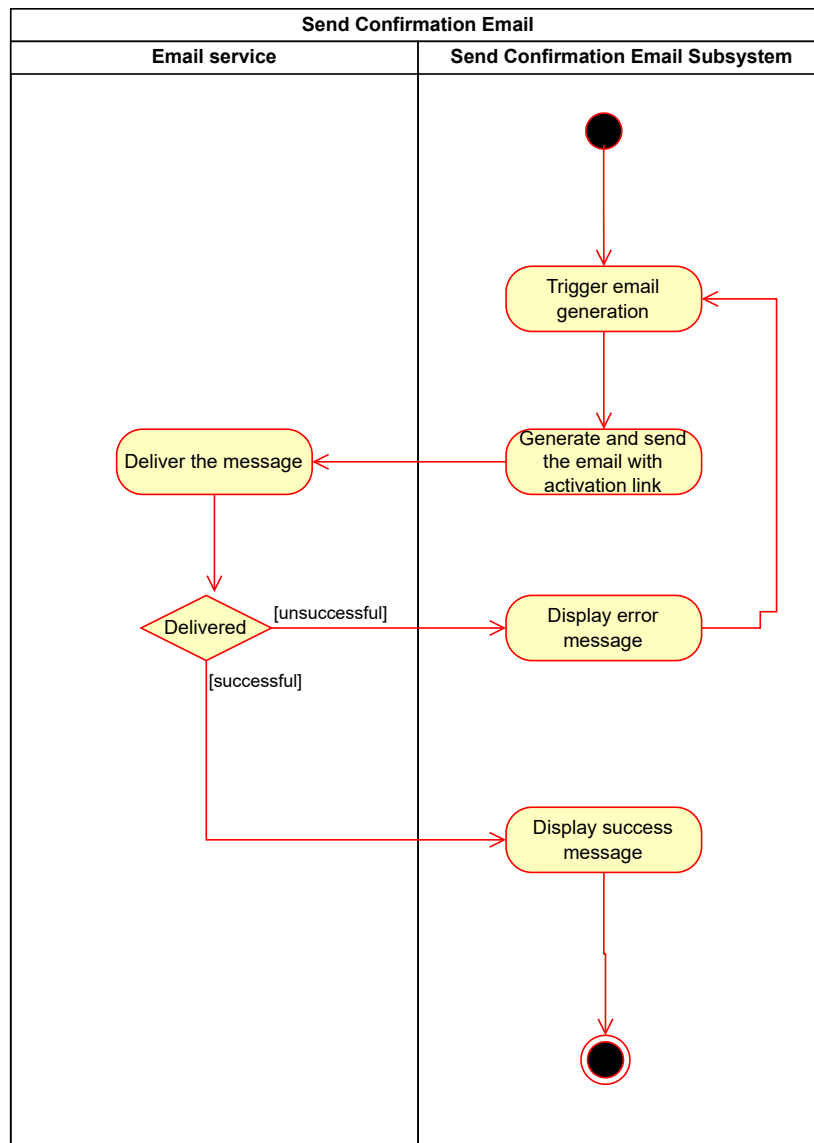
Preconditions: User completes registration.

Basic Path:

1. System triggers email generation.
2. System sends an email with activation link.
3. Email service delivers message.

Extensions:

- 2a. Email delivery failed.
1. System displays error.



Use Case: Send Reset Link

Inputs: User email.

Outputs: Email sent status.

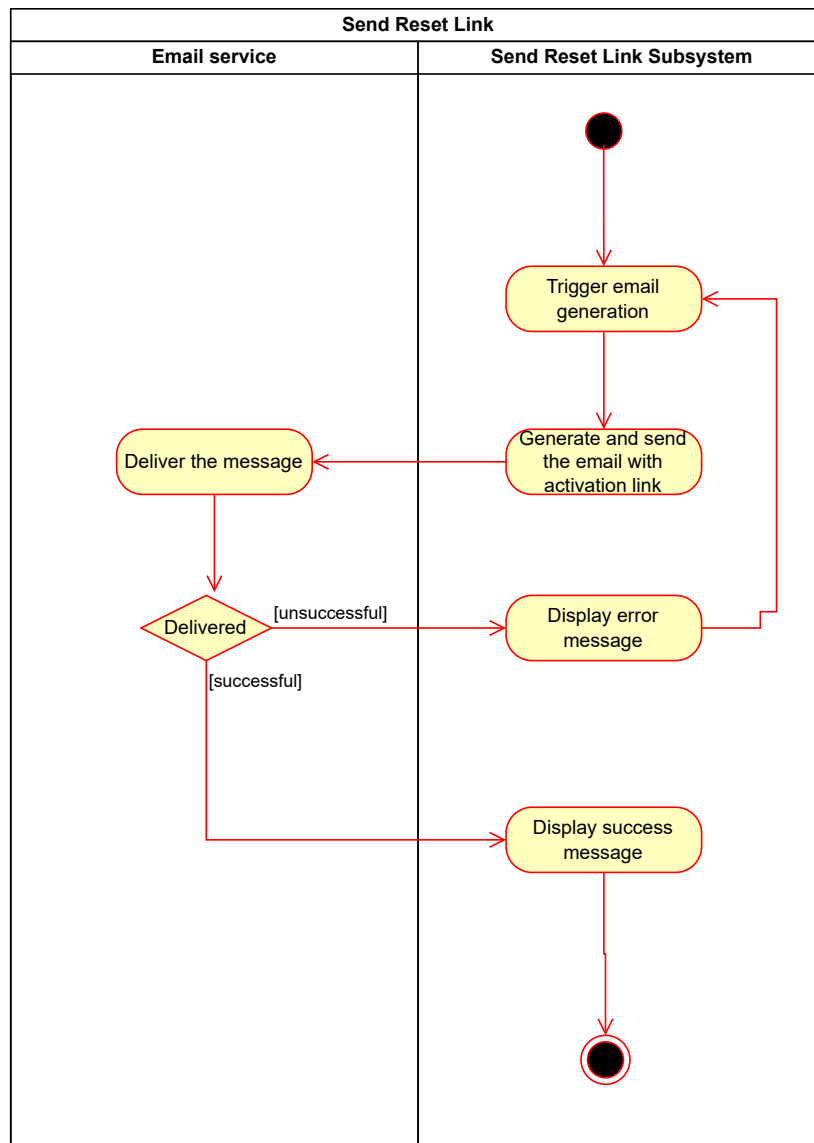
Preconditions: User requests password reset.

Basic Path:

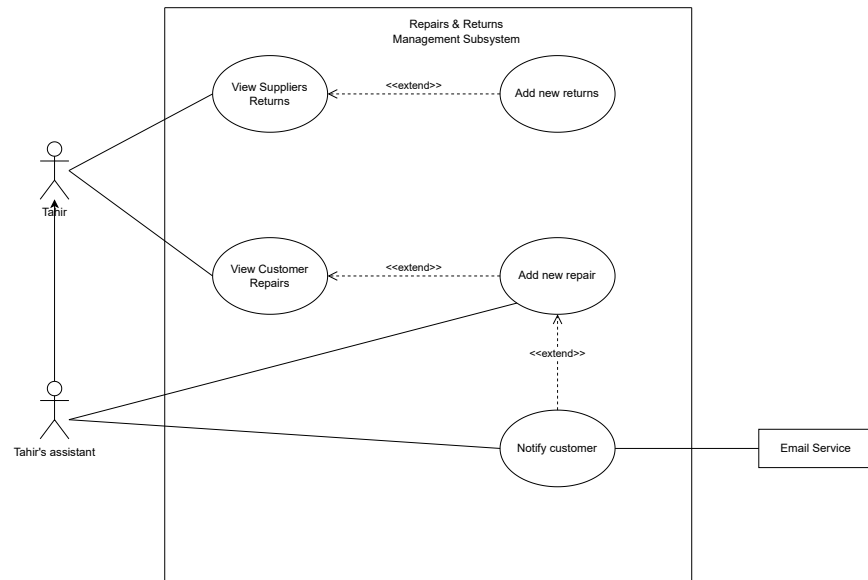
1. System triggers email generation.
2. System sends email with reset link.
3. Email service delivers message.

Extensions:

- 2a. Email delivery failed.
1. System displays error.



Repairs and damages Subsystem



Use Case: View Suppliers Returns

Inputs: N/A

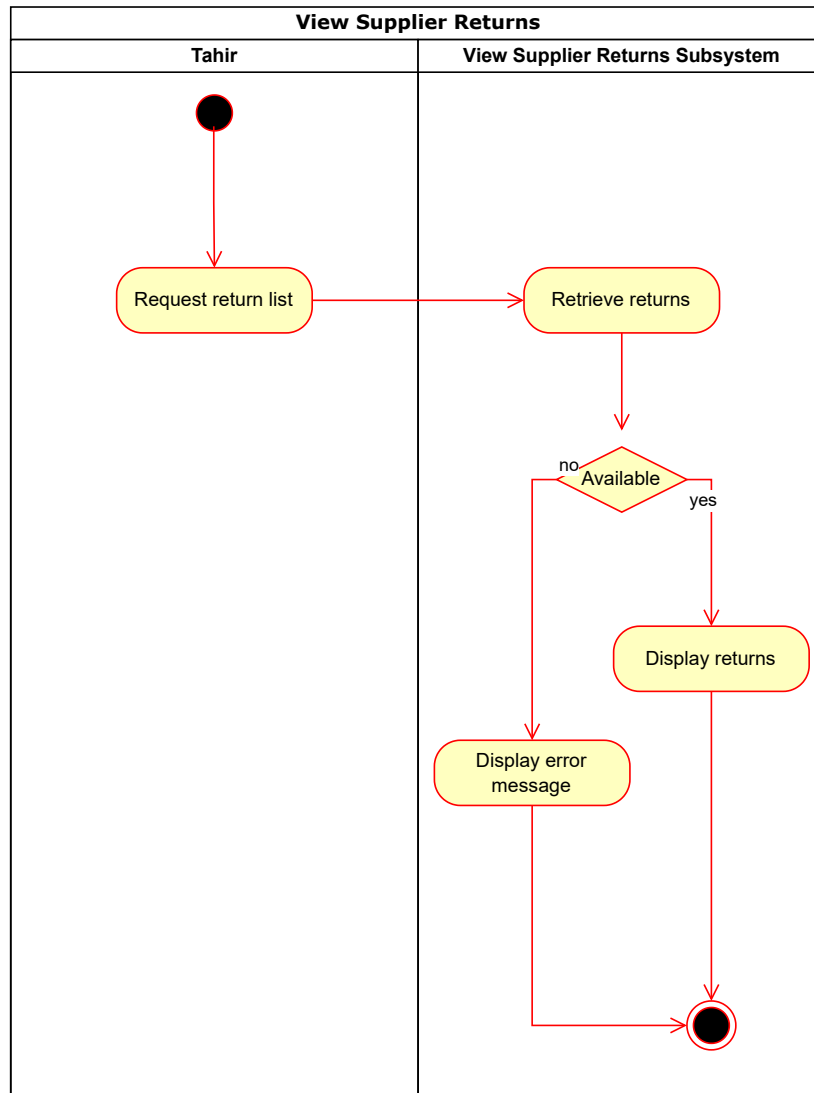
Outputs: List of all the products that need to be returned to suppliers

Preconditions: Supplier returns exist in the system.

Basic Path: 1. Tahir or Tahir's assistant selects "View Suppliers Returns." 2. System retrieves return records. 3. System displays returns

Extensions:

2a. No returns found. 1. System notifies Tahir or Tahir's assistant.



Use Case: Add New Returns

Inputs: Supplier details, product details, return reason.

Outputs: New return record, inventory update.

Preconditions: Product is identified as faulty/damaged.

Basic Path: 1. Tahir's assistant selects "Add New Returns." 2. System

prompts for supplier and product details. 3. Tahir's assistant provides supplier and product details. 4. Tahir's assistant enters return reason. 5. System validates and stores information. 6. System notifies the supplier. 7. System updates inventory.

Extensions:

5a. Invalid input data.

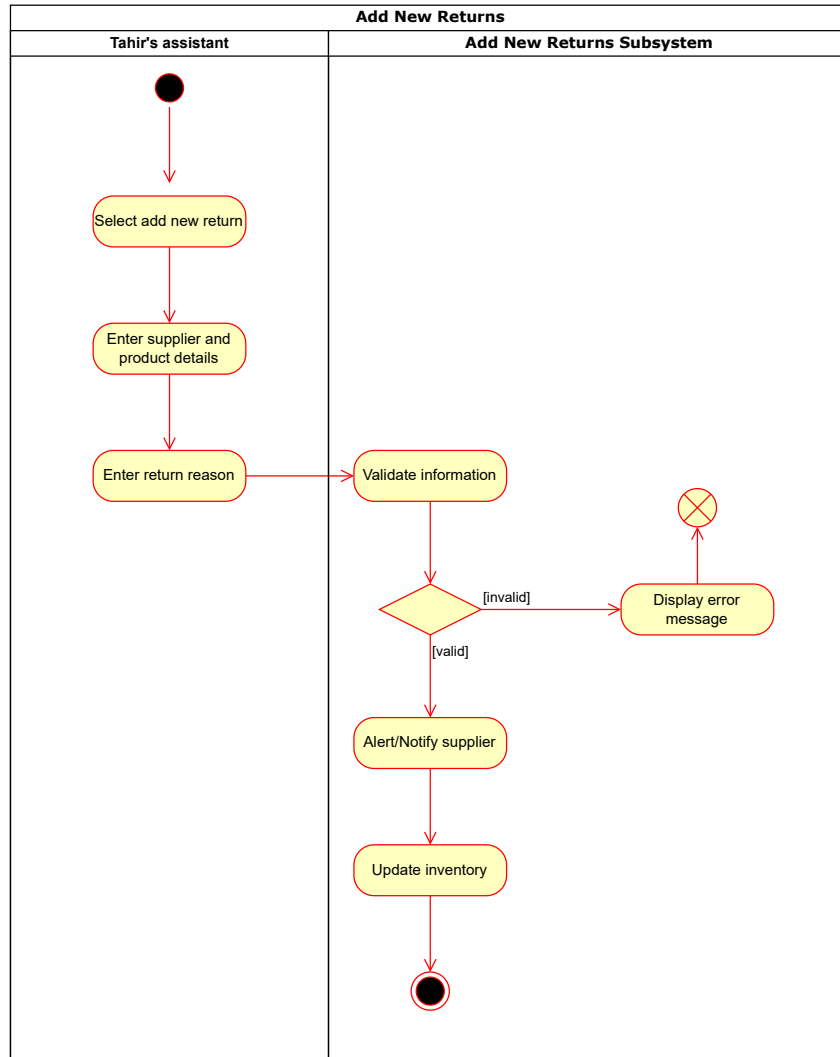
1. System rejects input and prompts correction.

5b. Duplicate entry (Return already exists).

1. Alert Tahir's assistant.

5c. Product does not exist in the database.

1. Alert Tahir's assistant and prompts for another entry.



Use Case: View Customer Repairs

Inputs: N/A

Outputs: List of customer products that need to be repaired.

Preconditions: Repairs exist in the system.

Basic Path: 1. Tahir or Tahir's assistant selects "View Customer Repairs."
 2. System retrieves repair records. 3. System displays repairs (e.g., product details, status, customer name).

Extensions:

2a. No customer repairs found. 1. System notifies Tahir or Tahir's assistant.

Use Case: Add New Repair

Inputs: Customer details, product details, repair description.

Outputs: New return record, inventory update.

Preconditions: Customer reports a product issue.

Basic Path: 1. Tahir's assistant selects "Add New Repair." 2. System prompts for customer and product details. 3. Tahir's assistant provides customer and product details. 4. Tahir's assistant enters the repair description. 5. System validates and stores information. 6. Assistant notifies the customer. 7. System updates inventory.

Extensions:

5a. Invalid input data.

1. System rejects input and prompts correction.

5b. Duplicate entry (Repair already exists).

1. Alert Tahir's assistant.

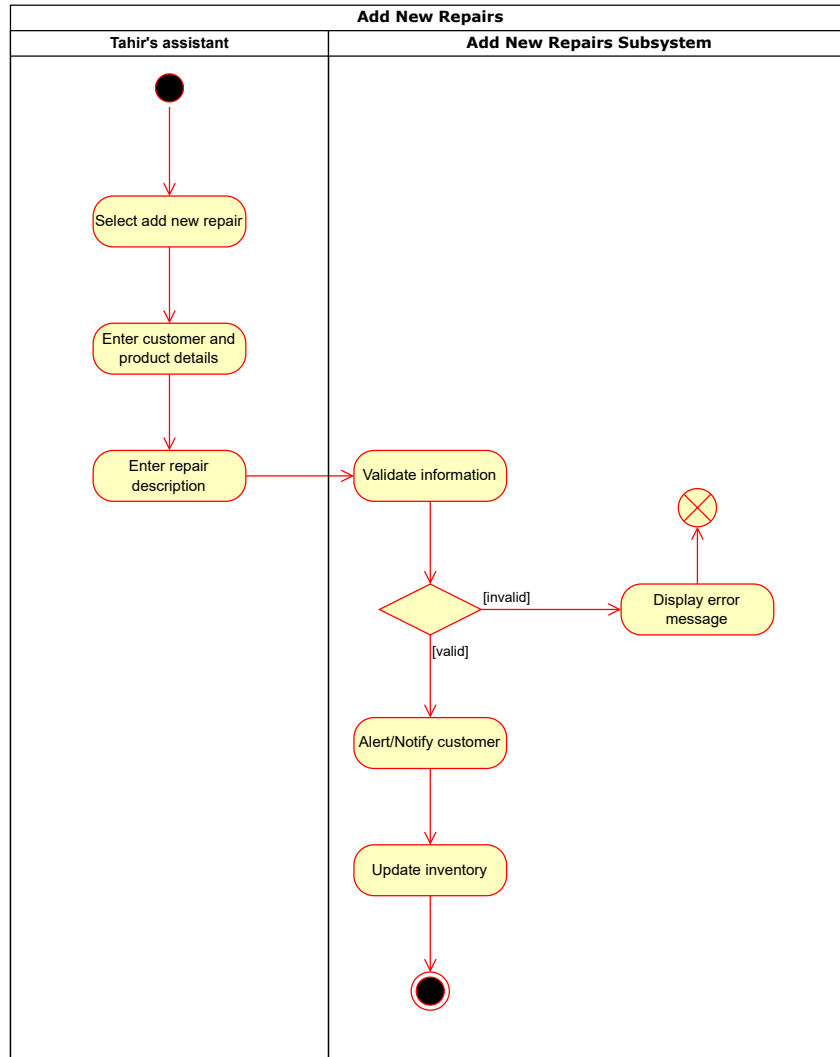
5c. Product not under warranty.

1. Alert Tahir's assistant and prompts for another entry.

5d. Customer does not exist.

1. Reject entry.

2. Notify Tahir's assistant.



Use Case: Notify Customer

Inputs: Repair ID, customer contact details.

Outputs: Notification sent confirmation.

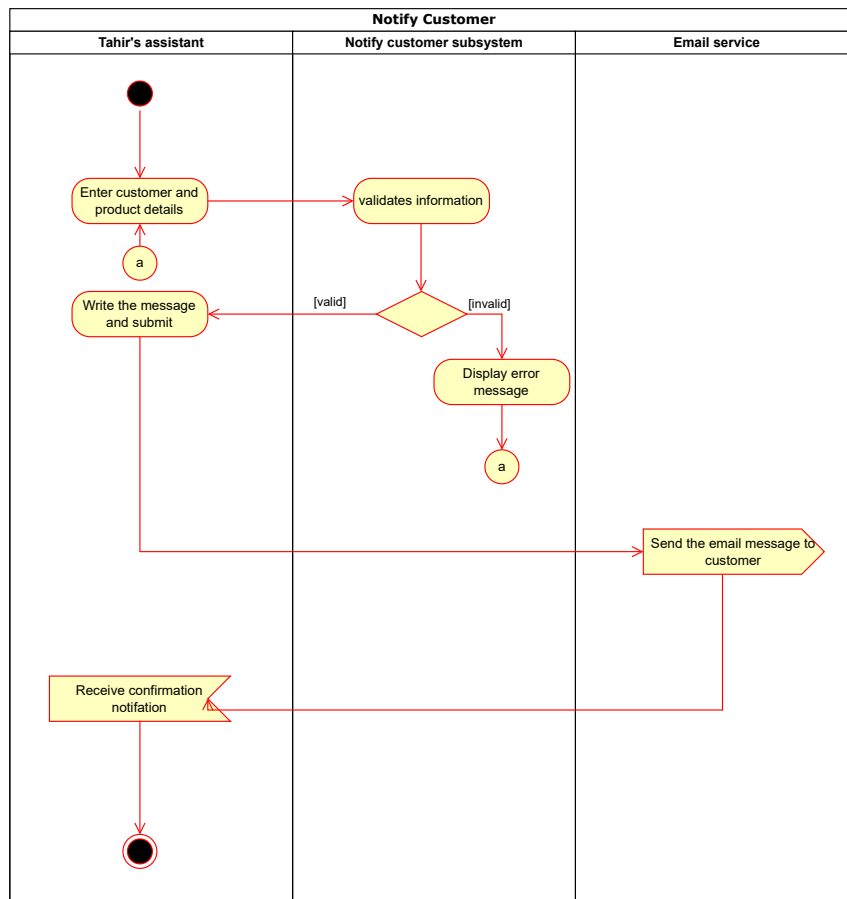
Preconditions: Repair request is logged or repair updates.

Basic Path: 1. Tahir's assistant selects "Notify customers." 2. System prompts for customer ID and product details. 3. Assistant fills in the required information. 4. System validates information. 5. Assistant writes the message. 6.

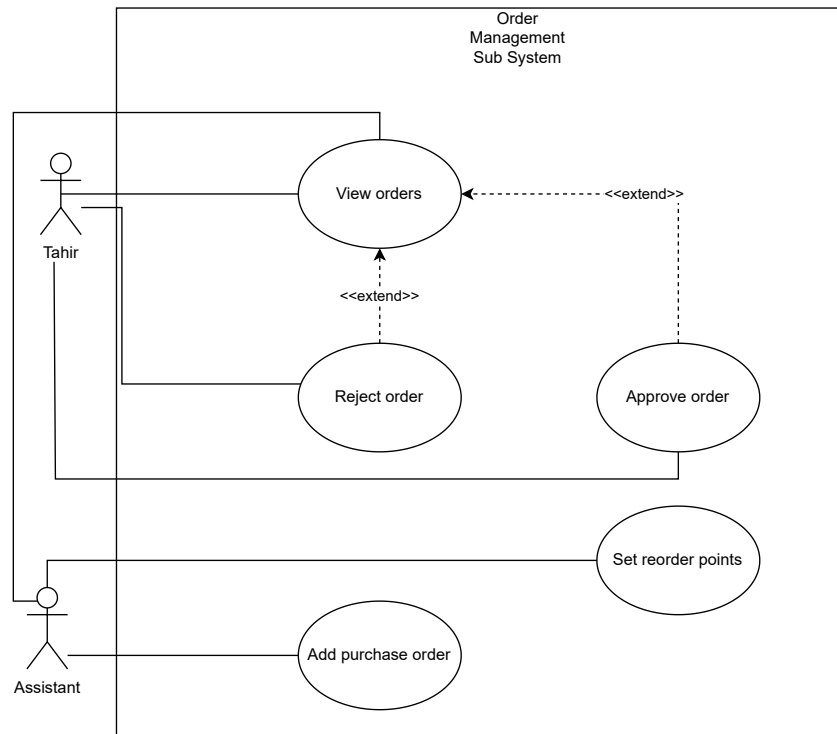
Email service delivers the message

Extensions:

- 5a. Invalid input data.
 - 1. System rejects input and prompts correction.
- 5d. Product does not exists.
 - 1. Reject entry.
 - 2. Notify Tahir's assistant.
- 5d. Customer does not exists.
 - 1. Reject entry.
 - 2. Notify Tahir's assistant.



Order Management Subsystem



Use case: View orders

Actor(s): Tahir, Assistant

Description: Allows the user to be able to view orders.

Preconditions: - User is signed in as an owner or assistant. - There must be existing orders.

Postcondition: The user must be able to see the list of orders if available.

Input:

Output: List of orders.

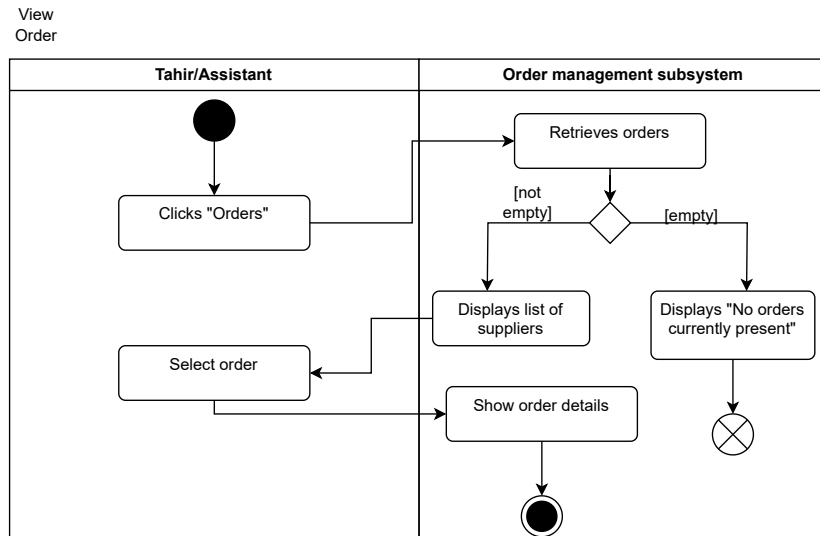
Basic Flow:

1. The user selects the “View Orders” option.
2. The system retrieves and displays the list of orders.
3. The user clicks on a specific order to see its details.
4. The system shows the order information.

Extensions:

- **1.a No orders available** - System displays a message: “No orders available.”

- **1b. Database error while retrieving orders:** - System logs the error and shows: “Unable to load orders. Please try again later.”



Use case: Approve orders

Actor(s): Tahir

Descriptions: Approve the recommended orders. **Preconditions:** - The owner must have viewed the order. - The order must be in a pending state.

Postcondition: The order is approved and the purchased order is created and sent to supplier.

Input:

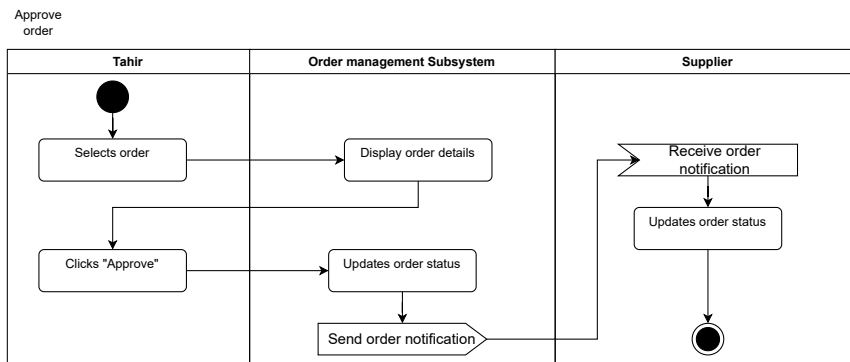
Output:

Basic Flow:

1. The owner selects an order from the list. 2. The owner clicks the “**Approve**” button. 3. The system checks if the order is still in pending status. 4. The system updates the order status.

Extensions:

- **2a. Order is already approved:** - System prevents re-approval and displays: “This order has already been processed.”



Use case: Reject orders

Actor(s): Tahir

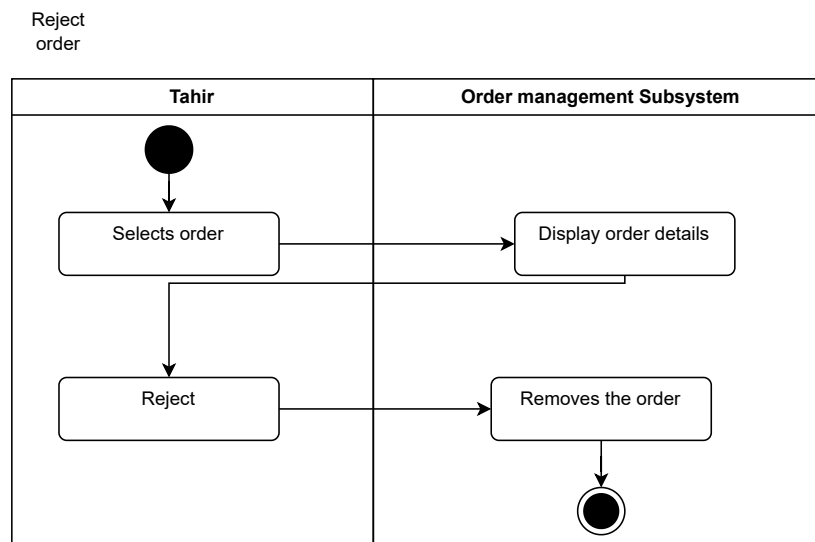
Descriptions: Reject the recommended orders. **Preconditions:** - The owner must have viewed the order. - The order must be in a pending state.

Input:

Output:

Basic Flow:

1. The owner selects an order from the list.
2. The owner clicks the “**Reject**” button.
3. The system checks if the order is still in pending status.
4. The system removes the order.



Use case: Add purchase order

Actor(s): Assistant

Description: Add a new purchase and send it to the owner.

Preconditions: - Must have logged in as an assistant.

Input: Order details.

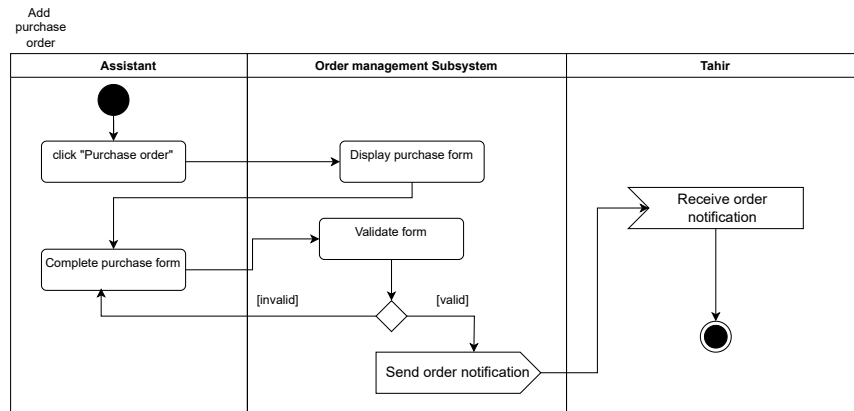
Output:

Basic Flow:

1. The assistant clicks “Purchase order”. 2. The system shows an order form for completion. 3. Assistant completes the order details and click submit. 4. The system validates the information and send the order to the owner.

Extensions:

- **4.a Missing information** - System shows an error message and allows the assistant to enter missing details or cancel. - **4.b Incorrect information** - System tells the assistant about the incorrect information enter and tell them to put relevant information or cancel.



Use case: Set Re-order points

Actor(s): Assistant

Description: Set a re-order point for a certain inventory.

Preconditions: - Must have logged in as an assistant.

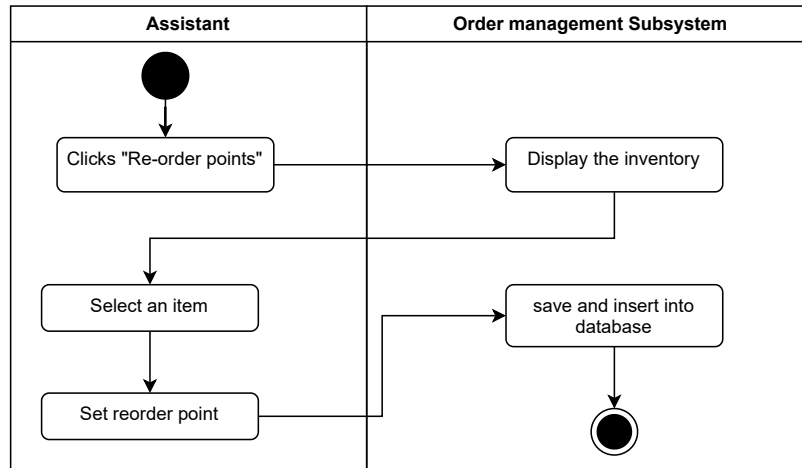
Input: Reorder levels.

Output:

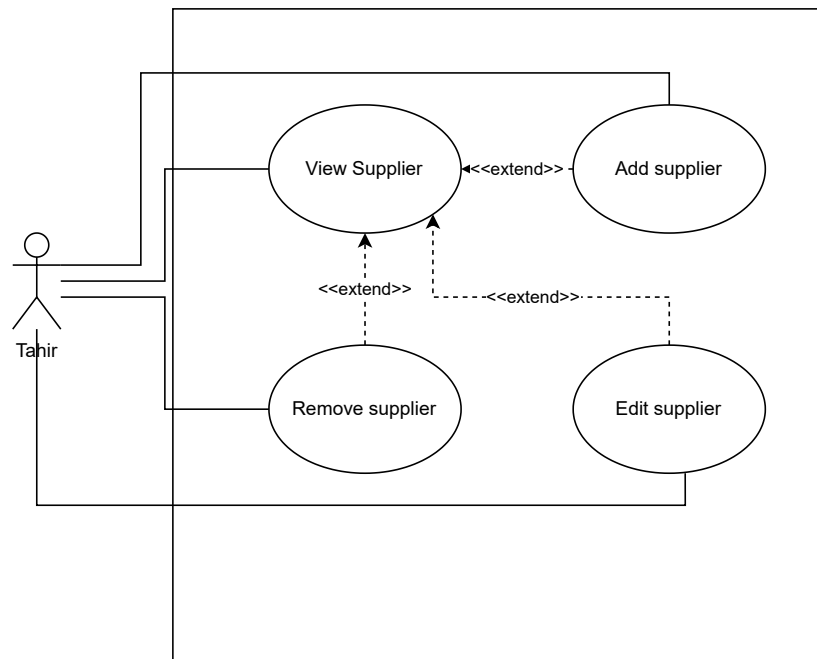
Basic Flow:

1. The assistant clicks “Re-order points”. 2. The system shows all the available inventory. 3. Assistant selects one of the items in the inventory and set re-order point. 5. The system save the re-order level and input in the inventory.

Set Re-
order
points



Supplier Management Subsystem



Use case: View Suppliers

Actor(s): Tahir

Description: Allows the owner to view suppliers that he was/is in contact with.

Preconditions: - User is signed in as an owner. - There must be existing suppliers.

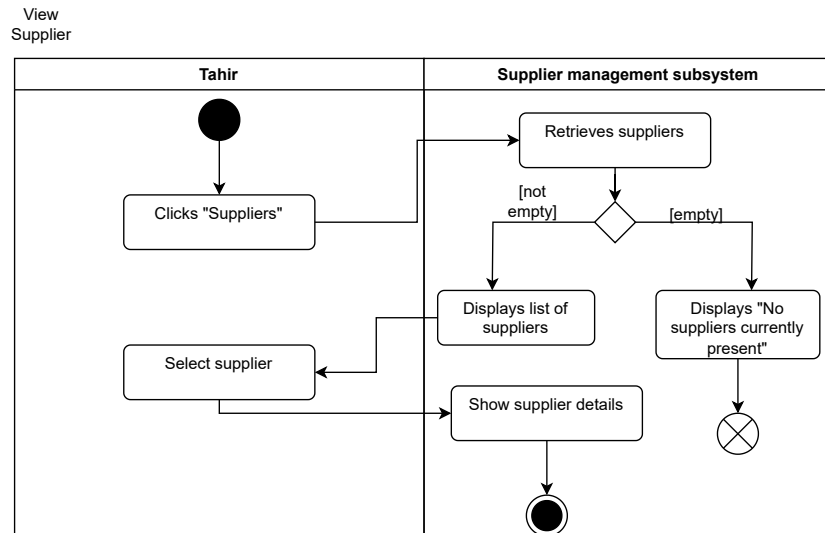
Input:

Output: List of company suppliers.

Basic Flow: 1. The owner selects the “Suppliers” option.
2. The system retrieves and displays the list of suppliers.
3. The owner clicks on a specific supplier to see its details.
4. The system shows the supplier information.

Extensions:

- **1.a No suppliers available** - System displays a message: “No suppliers available.” - **1b. Database error while retrieving suppliers:** - System logs the error and shows: “Unable to load suppliers. Please try again later.”



Use case: Add supplier

Actor(s): Tahir

Descriptions: Allows the owner to add a new supplier.

Include: View supplier

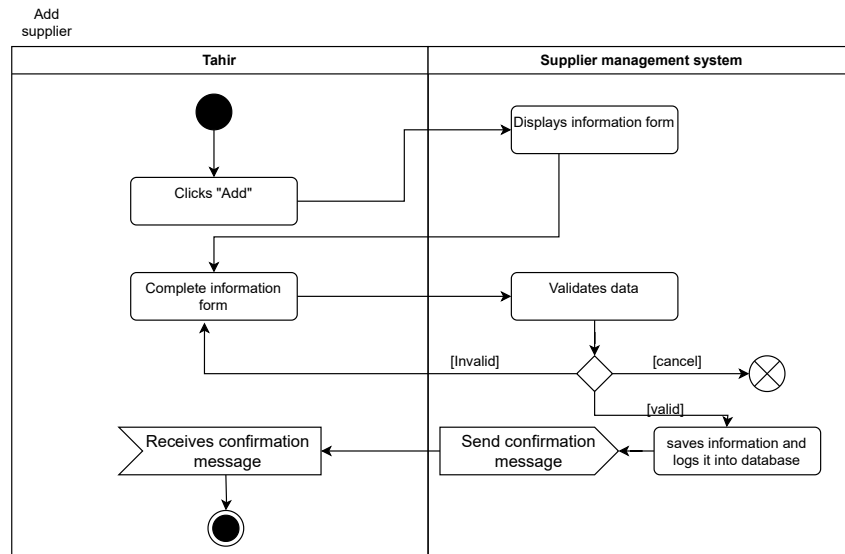
Preconditions: User is on the supplier management page.

Basic Flow:

1. Tahir selects “Add supplier”. 2. System displays a form for entering supplier details. 3. Tahir fills in the required fields . 4. Tahir clicks “Save”. 5. System validates input and adds the supplier. 6. Confirmation message is displayed.

Extensions:

- **3a. Missing required fields:** - System highlights missing fields and prompts user to complete them.” - **5a. Invalid input:** - System shows an error and does not save the supplier.” - **5b. supplier already exists:**
- System displays an error message: “supplier with this name already exists.”



Use case: Edit supplier

Actor(s): Tahir

Descriptions: Allows the owner to edit supplier information.

Include: View supplier. **Preconditions:** User has selected a supplier from the view.

Input:

Output:

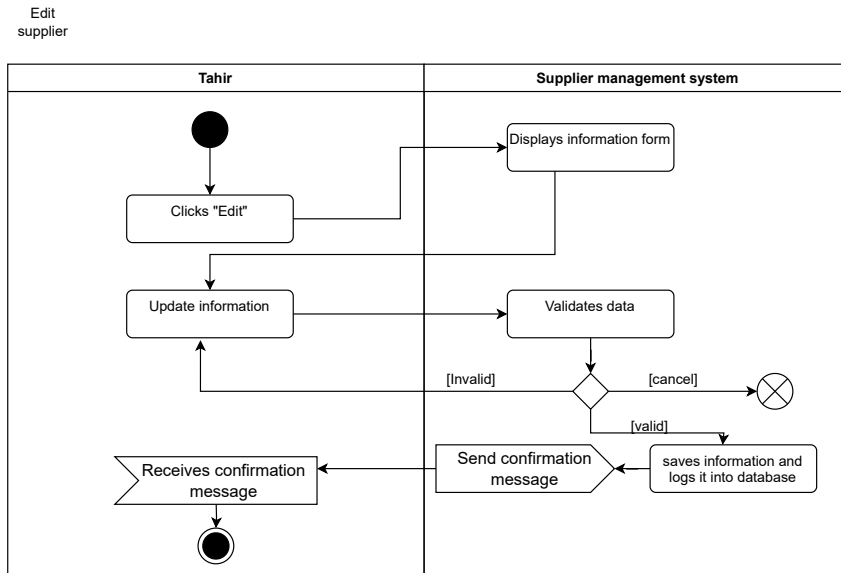
Basic Flow:

1. Tahir selects a supplier from the “View supplier” list. 2. Tahir clicks “Edit”. 3. System displays current supplier details in editable form. 4. Tahir updates the information. 5. Tahir clicks “Save”. 6. System validates and updates the supplier details. 7. Confirmation message is displayed.

Extensions:

- **4a. User leaves fields empty or adds invalid data:** - System highlights

errors and does not proceed. - **6a. System error during update:** - System shows an error message and logs the issue.



Use case: Remove supplier

Actor(s): Tahir

Descriptions: Allows the owner to remove the supplier from the system.

Include: View supplier.

Preconditions: Owner has selected a supplier from the list.

Input:

Output:

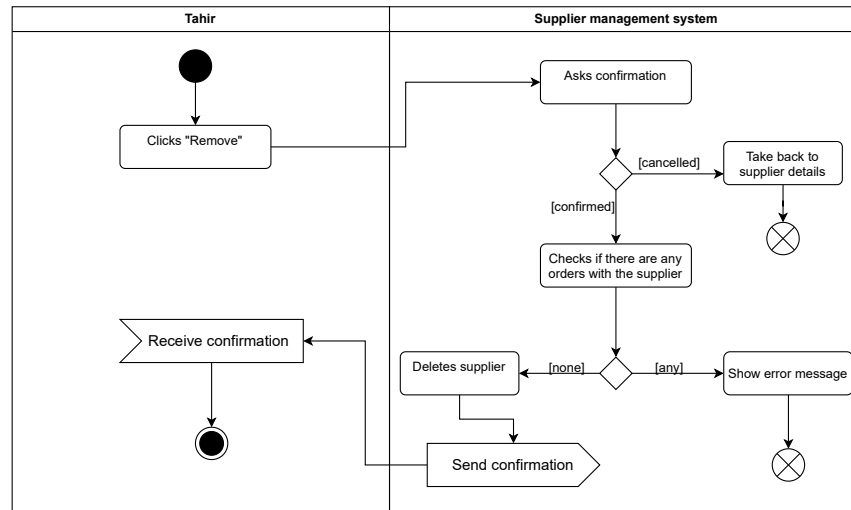
Basic Flow:

1. Tahir selects a supplier from the "View supplier" list. 2. Tahir clicks "Remove". 3. System prompts for confirmation. 4. Tahir confirms deletion. 5. System deletes the supplier. 6. Confirmation message is shown.

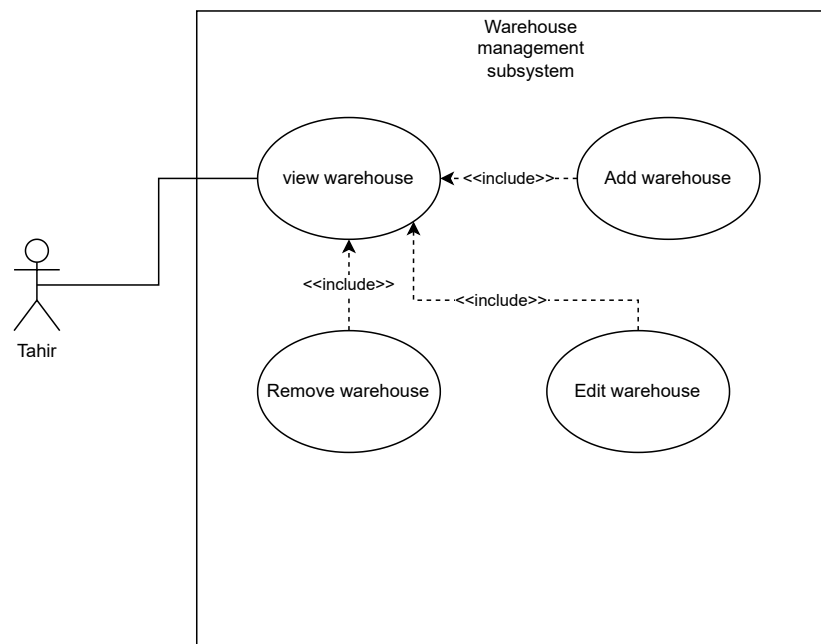
Extensions:

- **a. User cancels deletion:** - System aborts the deletion and returns to the previous view supplier details." - **5a. supplier is linked to existing inventory or records:** - System prevents deletion and shows message: "supplier cannot be deleted as it is linked to active inventory."

Remove
supplier



Warehouse Management Subsystem



Use case: View warehouses

Actor(s): Tahir

Description: Allows the owner to view all the company warehouses in different places.

Preconditions: - User is signed in as an owner. - There must be existing warehouses.

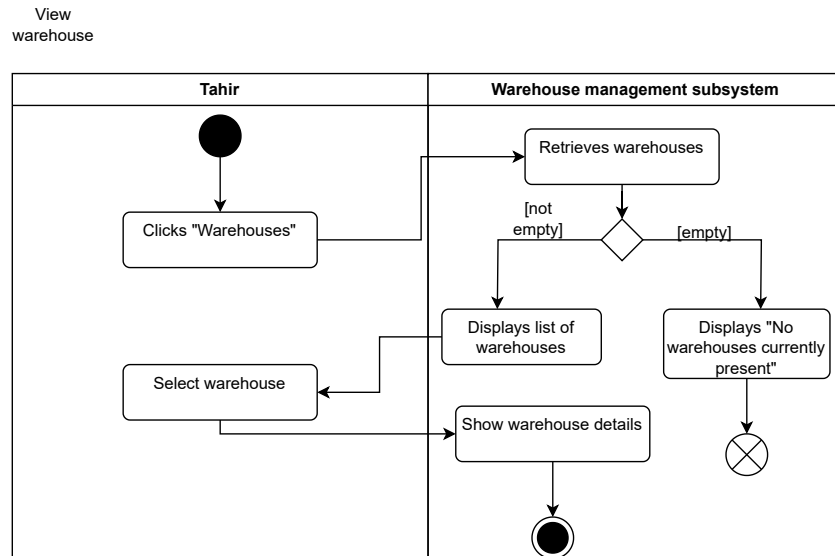
Input:

Output: List of company warehouses.

Basic Flow: 1. The owner selects the “Warehouses” option.
2. The system retrieves and displays the list of warehouses.
3. The owner clicks on a specific warehouse to see its details.
4. The system shows the warehouse information.

Extensions:

- **1.a No warehouses available** - System displays a message: “No warehouses available.” - **1b. Database error while retrieving warehouses:** - System logs the error and shows: “Unable to load warehouses. Please try again later.”



Use case: Add warehouse

Actor(s): Tahir

Descriptions: Allows the owner to add a new warehouse.

Include: View warehouse

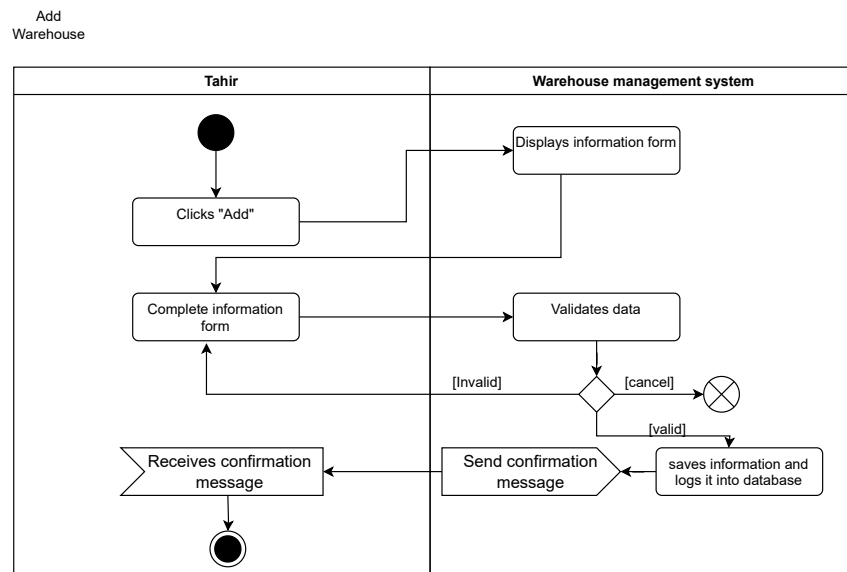
Preconditions: User is on the warehouse management page.

Basic Flow:

1. Tahir selects “Add Warehouse”. 2. System displays a form for entering warehouse details. 3. Tahir fills in the required fields . 4. Tahir clicks “Save”. 5. System validates input and adds the warehouse. 6. Confirmation message is displayed.

Extensions:

- **3a. Missing required fields:** - System highlights missing fields and prompts user to complete them.” - **5a. Invalid input:** - System shows an error and does not save the warehouse.” - **5b. Warehouse already exists:** - System displays an error message: “Warehouse with this name already exists.”



Use case: Edit warehouse

Actor(s): Tahir

Descriptions: Allows the owner to edit warehouse information.

Include: View warehouse. **Preconditions:** User has selected a warehouse from the view.

Input:

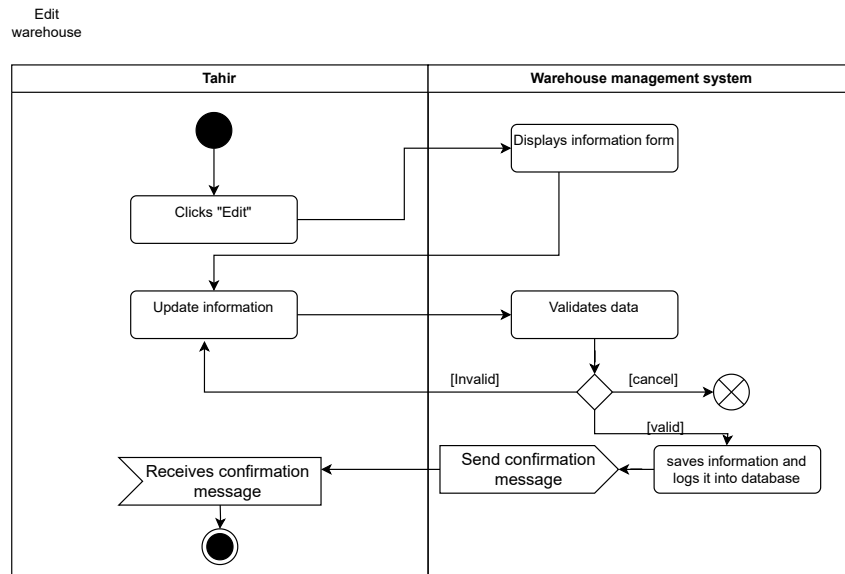
Output:

Basic Flow:

1. Tahir selects a warehouse from the “View Warehouse” list. 2. Tahir clicks “Edit”. 3. System displays current warehouse details in editable form. 4. Tahir updates the information. 5. Tahir clicks “Save”. 6. System validates and updates the warehouse details. 7. Confirmation message is displayed.

Extensions:

- **4a. User leaves fields empty or adds invalid data:** - System highlights errors and does not proceed.
- **6a. System error during update:** - System shows an error message and logs the issue.



Use case: Remove warehouse

Actor(s): Tahir

Descriptions: Allows the owner to remove the warehouse from the system.

Include: View warehouse.

Preconditions: Owner has selected a warehouse from the list.

Input:

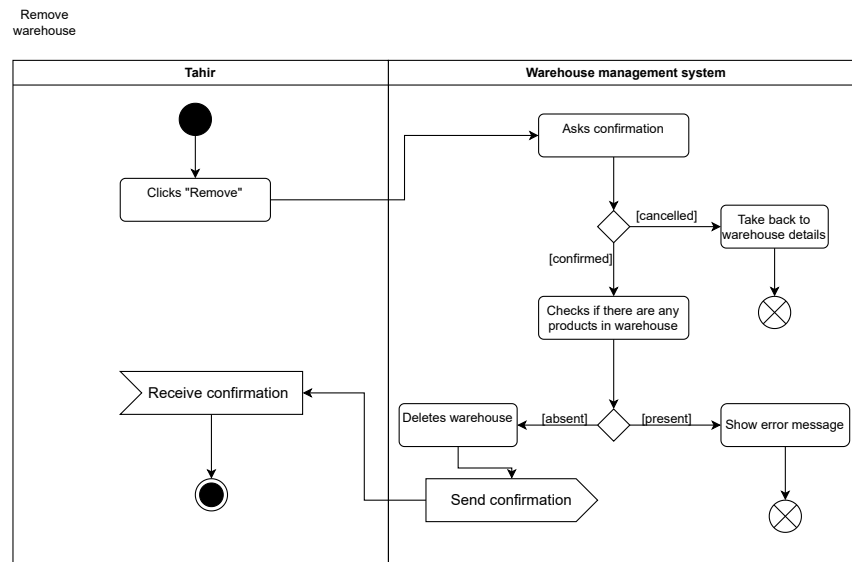
Output:

Basic Flow:

1. Tahir selects a warehouse from the "View Warehouse" list.
2. Tahir clicks "Remove".
3. System prompts for confirmation.
4. Tahir confirms deletion.
5. System deletes the warehouse.
6. Confirmation message is shown.

Extensions:

- **a. User cancels deletion:** - System aborts the deletion and returns to the previous view warehouse details."
- **5a. Warehouse is linked to existing inventory or records:** - System prevents deletion and shows message: "Warehouse cannot be deleted as it is linked to active inventory."



Export documents subsystem

Use case: Generate Stock Reports

Description: Gathers stock data in the system to be compiled and exported into a document.

Actors: Owner, assistant

Input: inventory data.

Preconditions: - Existing data in the database or inventory data.

Postconditions: - The stock report is generated in the system. - The report is exportable into PDF.

Flow: 1. User navigates to "Reports and Analysis" tab. 2. System displays report options. 3. User selects desired option(s). 4. User selects "Create report". 5. User selects the type of report. 6. The user chooses desired report data 7. System queries relevant data from the database. 8. System generates a report. 9. User confirms generation. 10. Document is stored in the system.

Extensions: 7a. Data is unavailable. 1. System notifies the user to wait or try again later.

Use case: Export Document

Description: Exports the report document into a PDF file.

Actors: Owner, assistant

Input: Generated report

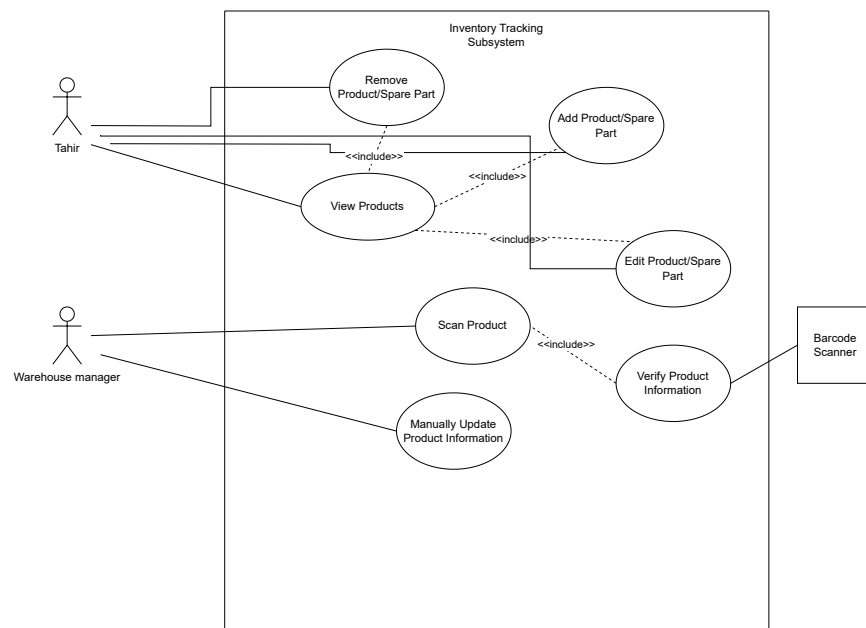
Output: PDF file report

Preconditions: - An existing report that has been generated.

Postconditions: - The report is saved as a PDF file and is available for download.

Flow: 1. User clicks “Download”. 2. System exports the file into PDF. 3. System generates file. 4. System sends file to the user. 5. User receives file. 6. User views file.

Inventory Management Subsystem



Use Case: View Inventory

Actor(s): Tahir, Assistant

Description: Allows the owner to view the available inventory.

Preconditions: - User is signed in as an owner or assistant. - There must be existing orders.

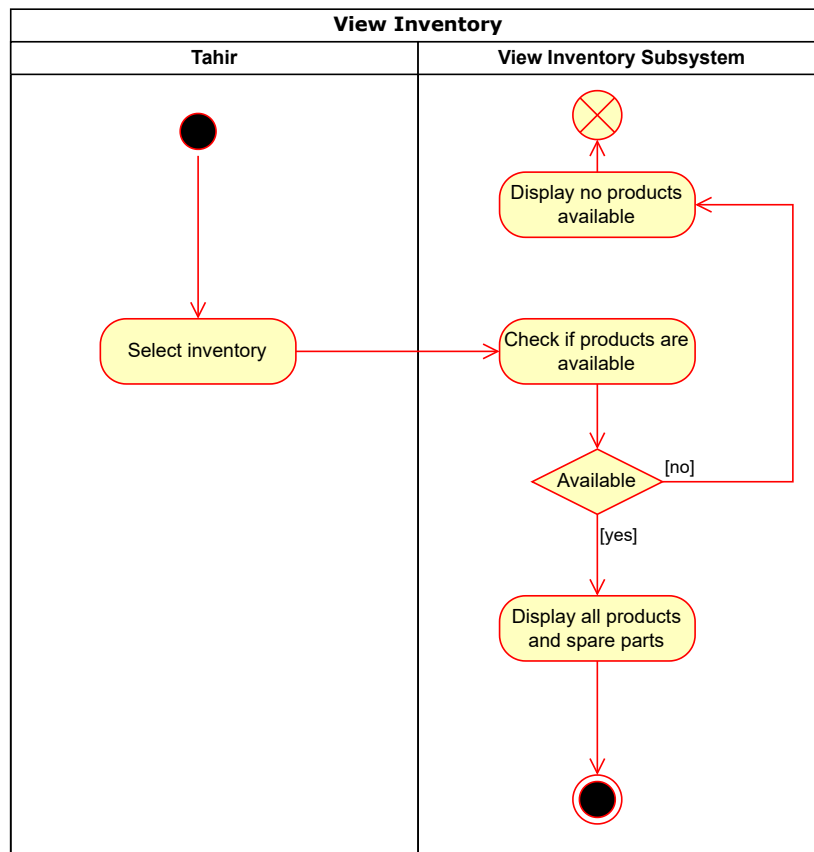
Input:

Output: 1. Tahir selects inventory 2. System displays all products and spare parts

Extensions

2a No products available

1. System informs the user to add product



Use Case: Add product/Spare part

Actors(s): Tahir , assistant

Descriptions: Adds new inventory items to the system.

Preconditions:

- User is logged in as owner. - Barcode scanner or keyboard is working

Input: Product details or Spare part details

Output: Confirmation

1. Tahir selects a relevant product or spare part 2. Tahir clicks on add product
3. Systems display a form 4. Tahir fills in the form 5. System validates input 6.
System stores product information 7. System sends a confirmation email

Extensions

5a Invalid data entered

1. System informs the user about the invalid input
2. Tahir can fix the input or cancel

Use Case: : Remove product/spare part

Actors(s): Tahir , assistant

Descriptions: Permanently deletes an item from inventory.

Preconditions:

- Item exists and has zero stock. - No pending orders linked to the item.

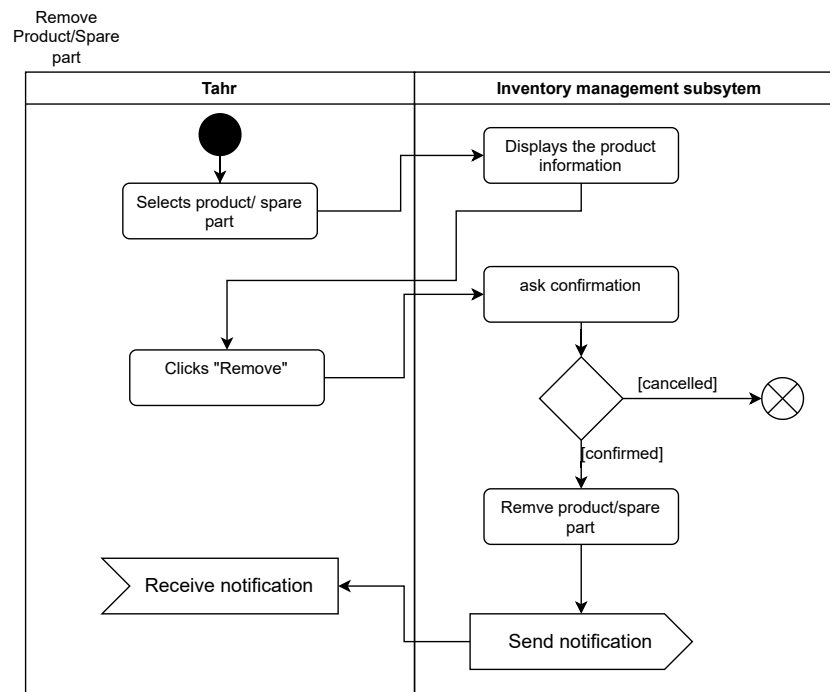
Input:

Output: Confirmation 1. Tahir selects a relevant product or spare part 2. The system displays the product or spare part details 3. Tahir clicks on remove product 4. The system asks for confirmation 5. System deletes product 6. System sends confirmation message

Extensions

4a. Tahir chooses no as answer

1. The system draws the change and go back to product information



Use Case: Edit product/spare part

Actors(s): Tahir , assistant

Descriptions: Modifies existing item details..

Preconditions:

- Item exists in the database. - User has edit permissions.

Input: product details or spare part details

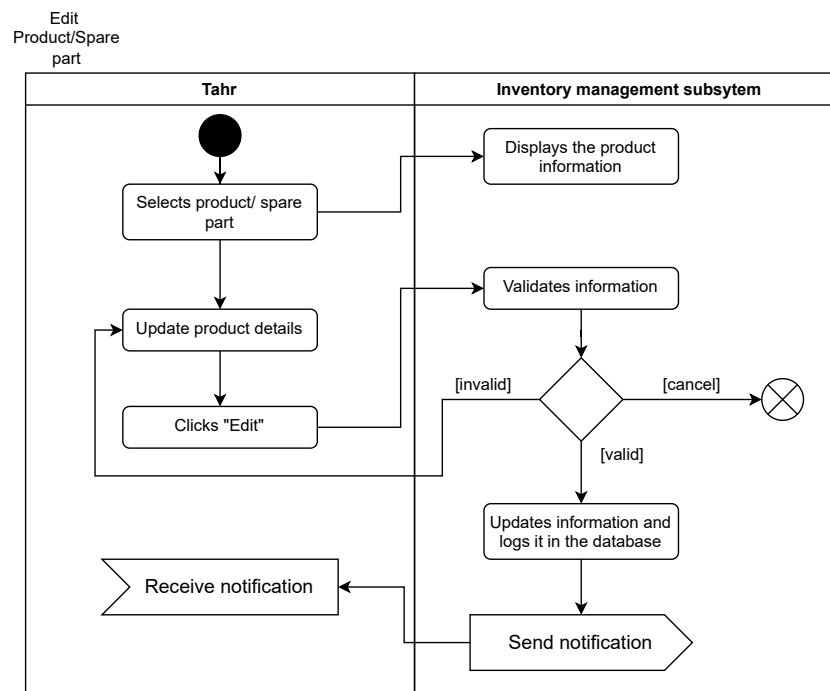
Output: Confirmation
1. Tahir selects a relevant product or spare part
2. The system displays the product information
3. Tahir clicks Edit product
4. Tahir changes product details
5. System validates data
6. System stores product or spare part details
7. System sends confirmation notification

Extensions

5a Invalid data entered

1. System informs the user about the invalid input

2. Tahir can fix the data or cancel



Use Case: Scan product

Actors(s): Warehouse Manager

Descriptions: Updates stock via barcode scan.

Preconditions:

- Mobile app is connected to the internet (or offline mode enabled). - Barcode scanner is functional.

Input: Product Quantity

Output: Confirmation

1. Warehouse Manager opens “Scan product” feature on mobile app
2. Warehouse Manager Scans the barcode of the product
3. System validates barcode
4. System display product information
5. Warehouse Manager confirms quantity
6. System updates product information
7. System displays confirmation message

Extensions

1a Barcode scanner offline

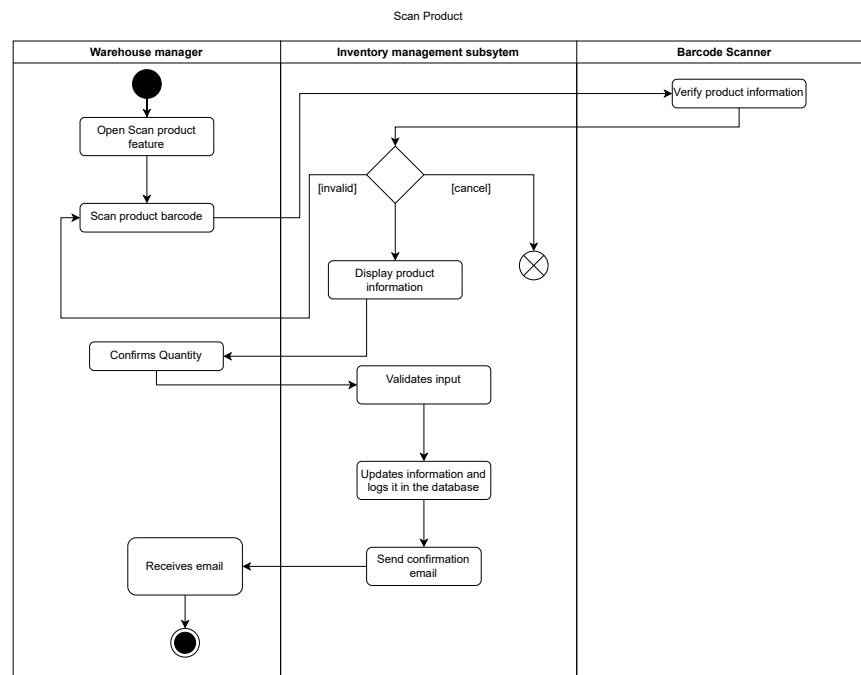
1. Systems display barcode scanner not available
2. Warehouse Manager click manually update product information or cancel

3a Invalid Barcode

1. Systems detect invalid barcode and display Barcode not found
2. Warehouse Manager can rescan or manual update product

6a Product does not exist

1. System informs the warehouse manager that product is not in the database
2. Warehouse manager can notify Tahir or cancel



Use Case: Manually Update product information

Actors(s): Warehouse Manager

Descriptions: Adjust stock without scanning.

Preconditions:

- Product exists in the database.

Input: Quantity

Output: Confirmation
1. Warehouse Manager selects update product
2. System displays all products and spare parts
3. Warehouse manager search the product using product name or product id
4. System display the product
5. Warehouse manager selects relevant product
6. Warehouse manager confirms quantity
7. System validates input
8. System updates the product information
9. System displays confirmation message

Extensions

3a Product does not exist

1. System informs the user that the product is not available
2. Warehouse manager retype product name or cancel

7a Invalid data entered

1. System informs the user that invalid data entered
2. Warehouse manager can retype the data or cancel

