

Construction 2

Document 2: Use Cases & Models



HexaTech

INFO3003A

Table of Contents: Construction 2

Version Control	2
Description of System Boundary.....	3
Overview of Actors and Goals	4
Use-Case Set.....	5
Core and Maintenance Use Cases	5
Use Case Diagram	5
Administrator Use Case Diagram	6
Technician Use Case Diagram	7
Narrative Explanation for Use Case Diagrams.....	8
Fully Dressed Use Cases	8
Create Repair	8
Sequence Diagram for Create Repair	9
Update Repair.....	10
Sequence Diagram for Update Repair.....	10
Create Job	11
Sequence Diagram for Create Job	12
Update Job	12
Sequence Diagram for Update Job	13
Update Part.....	13
Sequence Diagram for Update Part.....	14
Create RepairPart.....	14
Sequence Diagram for Create RepairPart.....	15
Maintenance Use Cases	15
Read Repair Use Case.....	15
Create Customer Use Case.....	15
Update Customer Use Case.....	15
Read Customer Use Case.....	15
Create Part Use Case.....	16
Read Part Use Case	16
Update RepairPart Use Case	16
Read RepairPart Use Case	16
Create Unit Use Case.....	16
Update Unit Use Case	16
Read Unit Use Cases	16
Create Staff Use Case.....	17
Update Staff Use Case	17

Read Staff Use Case	17
Read Job Use Case.....	17
Create Manufacturer Use Case	17
Update Manufacturer Use Case	17
Read Manufacturer Use Case	18
State Machine Diagram	18
Repair Entity.....	18
Domain and Data Models	19
Entity Relationship Diagram (ERD).....	19
Class Diagram.....	20
Business Process Modelling (Activity Diagrams).....	21
Create Repair	21
Explanation	22
Update Repair	22
Explanation	22
Create Job.....	23
Explanation	23
Update Job	24
Explanation	24
Update Part	25
Explanation	25
Create RepairPart	26
Explanation	26
System Feature List.....	26

Version Control

Updates to documentation from elaboration 1 to elaboration 2.

<u>Update</u>	<u>Details</u>
----------------------	-----------------------

Use Case Diagram	<ul style="list-style-type: none"> • Changed to 3 separate use case diagrams. • Included narrative description of the diagrams.
Use Case Descriptions	<ul style="list-style-type: none"> • Fully dressed the 3 remaining core use cases. (Update Job; Update Part; Create RepairPart). • Briefly described all remaining maintenance use cases
Sequence Diagrams	<ul style="list-style-type: none"> • Updates to the 3 use cases from iteration 2 – added a controller. • Added sequence diagrams for newly described use case descriptions for 3 core use cases (See above).
Domain and Data Models	<ul style="list-style-type: none"> • Removed loan unit from the ERD and Class Diagram in alignment with the change in scope.
Business Process Modelling	<ul style="list-style-type: none"> • Edits made to the 3 activity diagrams from elaboration 2 – added swimlanes. • Added activity diagrams for the 3 newly added core use cases.

Updates to documentation from elaboration 2 to construction 1.

<u>Update</u>	<u>Details</u>
Use Case Diagram	<ul style="list-style-type: none"> • Removed Customer from use case diagrams
Sequence Diagrams	<ul style="list-style-type: none"> • Updated sequence diagrams with minor wording and format edits
Business Process Modelling	<ul style="list-style-type: none"> • Updated activity diagrams with minor wording changes
Actors and goals	<ul style="list-style-type: none"> • Removal of customer
System feature list	<ul style="list-style-type: none"> • Removal of map feature as the system is now internal

Note: The major scope change is the removal of the customer from the actors, with the decision to make the Repairs Hub system completely internal.

Updates to documentation from construction 1 to construction 2.

<u>Update</u>	<u>Details</u>
Entity Relationship Diagram	<ul style="list-style-type: none"> • Removed LoanUnitID as a foreign key in repair.

Description of System Boundary

The system boundary can be described as all the events that can be triggered by an actor. With this in mind, we can say that our system boundary is all use cases that constitute the repairs application being developed for SolarWay Suppliers.

The system boundary is all the internal components of the information system, including the interfaces, the business logic for the use cases, and the database. The 2 actors are outside

the system and interact with the interfaces to use the system. Their input goes into the system and the system processes the requests and provides a function and/or output.

The system boundary is the scope of the system, it is everything that will be included in achieving the required functionalities of the system.

Overview of Actors and Goals

<u>Actors</u>	<u>Goals</u>
Technician	<ul style="list-style-type: none">- View repairs to be completed.- Update parts.- Orders parts- Assign parts to repair.- Log work hours on job tickets.- Update details of repairs.
System Administrator	<ul style="list-style-type: none">- Checks unit in or repair.- Updates repair statuses.- Receives updates from manufacturers and updates repair statuses.- Orders parts.- Maintains customer, staff, and manufacturer records.- Maintains unit details.- Assigns job to technicians where required

Use-Case Set

Core and Maintenance Use Cases

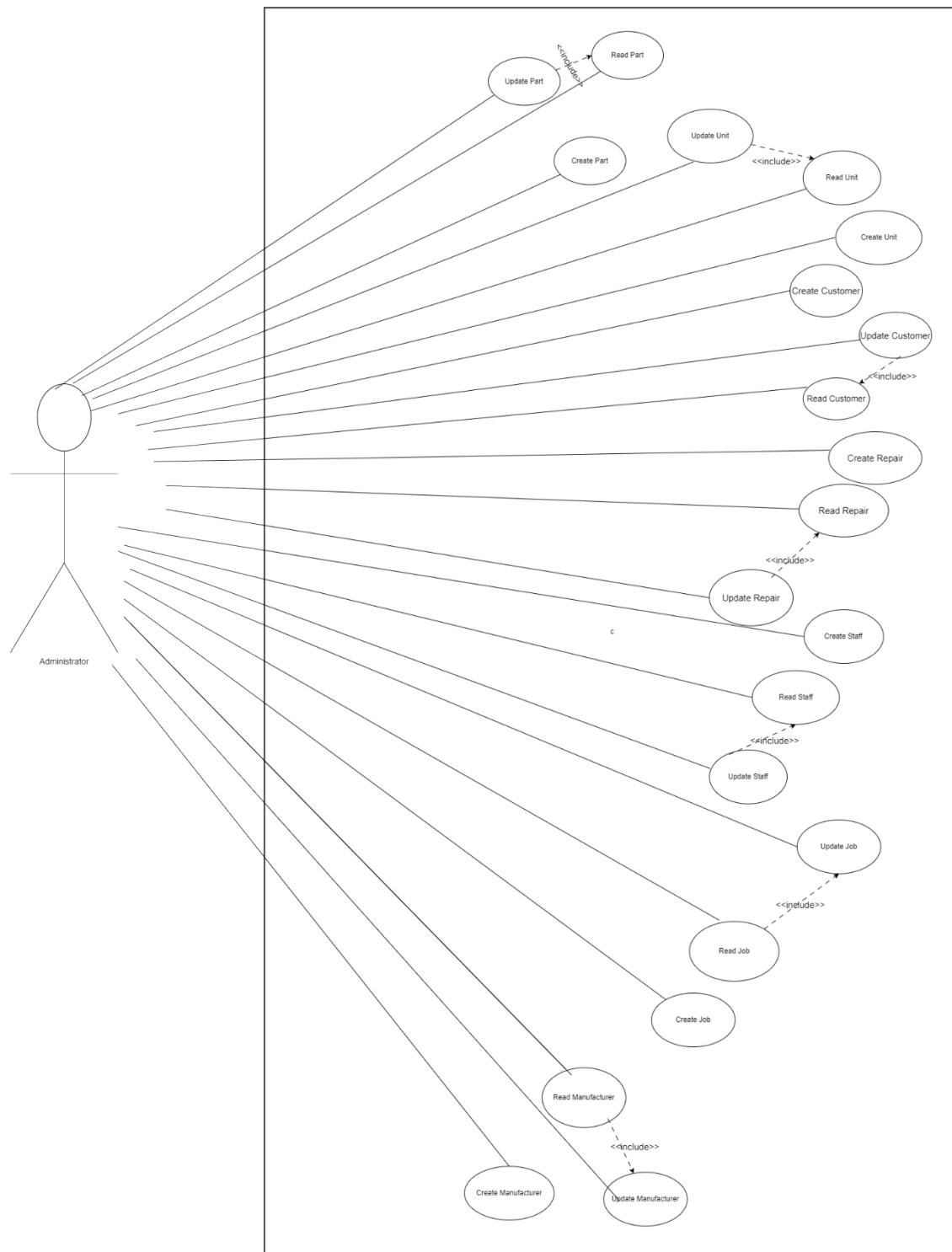
Core use cases are in red, maintenance use cases are in blue.

Entity	Use Cases
Repair	Create Repair
	Update Repair
	View Repair
Customer	Create Customer
	Update Customer
	View Customer
Part	Create Part
	Update Part
	View Part
RepairPart	Create RepairPart
	Update RepairPart
	View RepairPart
Unit	Create Unit
	Update Unit
	View Unit
Staff	Create Staff
	Update Staff
	View Staff
Job	Create Job
	Update Job
	View Job
Manufacturer	Create Manufacturer
	Update Manufacturer
	View Manufacturer

Use Case Diagram

The use case diagram has been split into 3 (one for each actor) for easy viewing

Administrator Use Case Diagram



Technician Use Case Diagram



Narrative Explanation for Use Case Diagrams

The use case diagram has been split into 2 separate parts, each part representing one of the actors involved in the SolarWay Suppliers system. These are the: administrator, and technician.

The administrator is the employee(s) at the front desk of the repairs department at SolarWay Suppliers. They oversee the entire department from the front and act as a connection between technicians and the customers. Their primary job is to manage (create, read, update) the repairs. They can manage accounts in the system, both staff and customers. The administrator can also manage units stored in the system – they correspond with sales and management (not applicable to this system) to determine the units stocked. They also have the ability to manage jobs, assigning them to technicians and logging hours where needed. Finally, the administrator manages the manufacturers in the system, having the ability to add, update their information, or simply view their information.

The technicians are the employees responsible for repairing the units brought in. They have several applicable use cases in the system. Firstly, they can read repairs currently in the system and update them to keep statuses real time through the repair pathway. They also can manage parts in the system, and parts associated with repairs, as this is their expertise in the repairs department. The technicians can also manage jobs – jobs are the lifeblood of a technician as it defines their work and how many hours they spend on each repair. Lastly the technician can view their staff profile on the system.

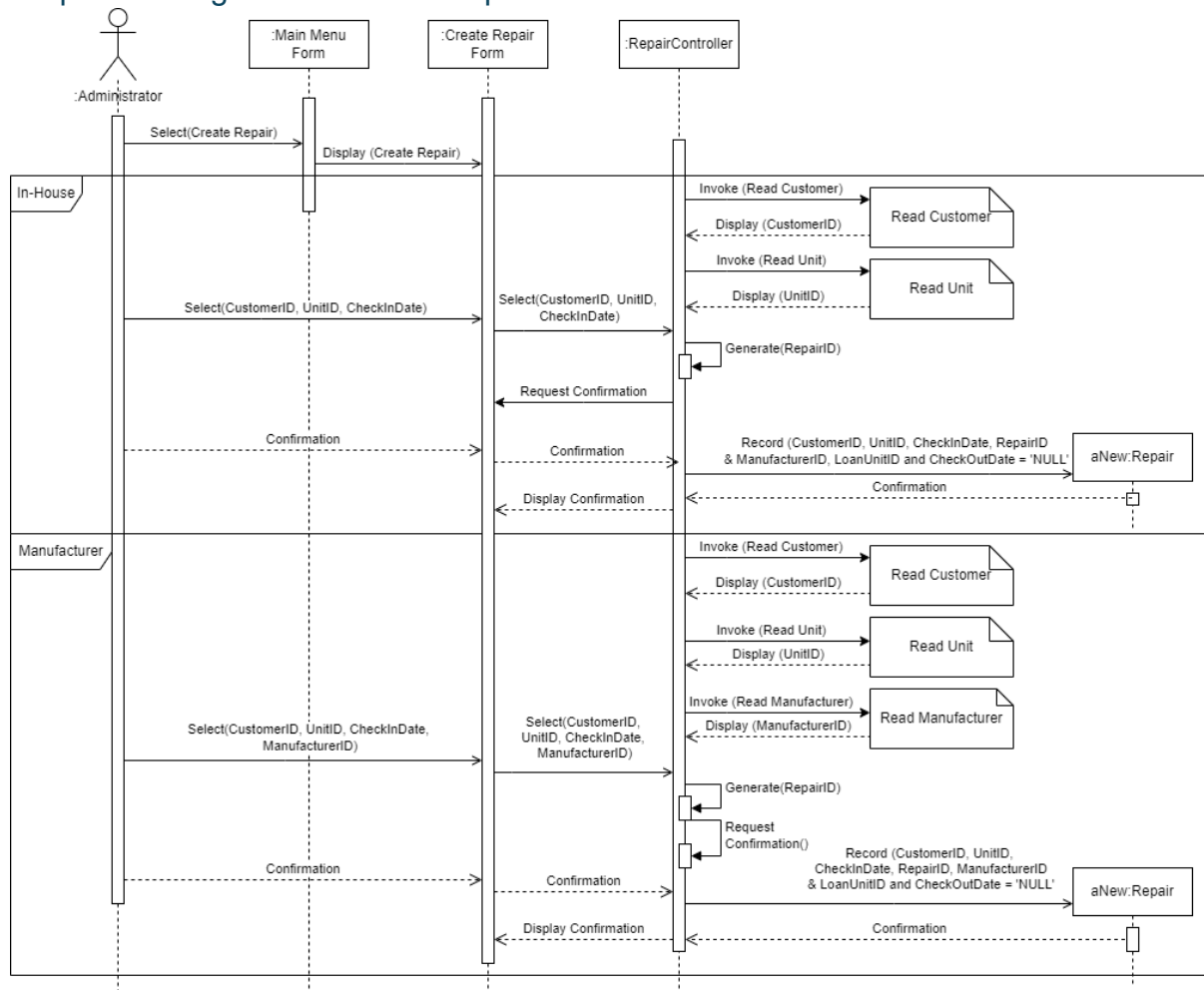
Fully Dressed Use Cases

Create Repair

Use Case Name	Create Repair	
Triggering Event	The administrator selects the option to add a new repair.	
Actor(s)	Administrator	
Related Use Cases	Read Customer, Read Unit, Read Manufacturer	
Pre-conditions	<ul style="list-style-type: none">Customer, Unit and (if applicable) Manufacturer must exist in their respective tables.	
Post-conditions	<ul style="list-style-type: none">New repair must be created in the Repairs Table using attributes RepairID, CustomerID, UnitID, CheckInDate, CheckOutDate = NULL, and ManufacturerID (set to NULL if in-house repair).	
Flow of Activities	Frame 1 – In-house Repair	
	Administrator	System
	1. Requests to create a new in-house repair.	1.1 Prompts the user to select and enter data, including the following: CustomerID (Invokes Read Customer), UnitID (Invokes Read Unit), and the check in date.
	2. Enters and selects the repair information required.	2.1 Generates RepairID 2.2 Requests confirmation from user
	3. Confirms the repair	3.1 Records repair information including CustomerID, UnitID, CheckInDate, RepairID, and sets

	ManufacturerID, LoanUnitID, and CheckOutDate as null values. 3.2 Confirms repair creation.
Frame 2 – Manufacturer Repair	
Administrator	System
1. Requests to create a new manufacturer repair. 2. Enters and selects the repairs information required. 3. Confirms the repair.	1.1 Prompts the user to select and enter information, including the following: CustomerID (Invokes Read Customer), UnitID (Invokes Read Unit), ManufacturerID (Invokes Read Manufacturer), and the check in date. 2.1 Generates RepairID 2.2 Requests confirmation from user 3.1 Records repair information including CustomerID, UnitID, CheckInDate, RepairID, ManufacturerID and sets LoanUnitID, and CheckOutDate as null values. 3.2 Confirms repair creation.
Extensions	No extensions (alternate flows) are applicable to this use case.

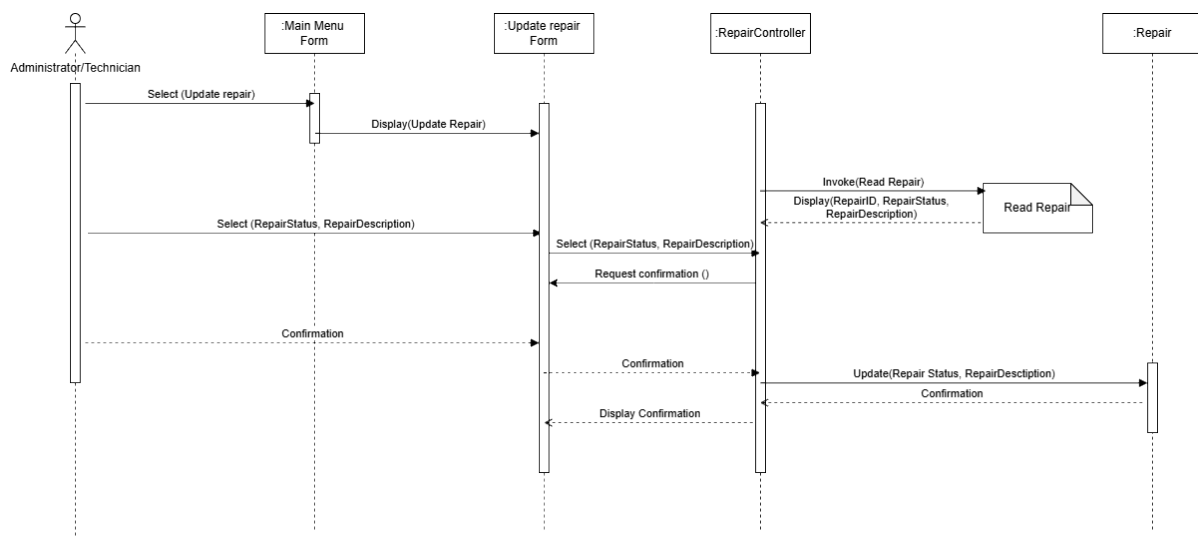
Sequence Diagram for Create Repair



Update Repair

Use Case Name	Update Repair	
Triggering Event	The technician/administrator selects the option to update repair information in the system.	
Actor(s)	Administrator/Technician	
Related Use Cases	Read Repair	
Pre-conditions	<ul style="list-style-type: none"> There are ongoing repairs in the system. The customer's repair information exists in the system. 	
Post-conditions	<ul style="list-style-type: none"> The repair information in the system is updated like RepairID, UnitID, CheckInDate, CheckOutDate, RepairStatus. 	
Flow of Activities	Technician/administrator	System
	1. Selects the option to update repair information in the system. 2.1 Selects the repair from the list. 2.2 Selects new repair status and enters description 3. Confirms the status change.	1.1 The system displays a list of ongoing repairs with details such as RepairID (invoking Read Repair Use Case) – with search functionality. 2.1 Prompts for confirmation 3.1 Records the updated RepairStatus and description. 3.2 Generates an email to send to the customer with the updates to their repair unit.
Extensions	No extensions (alternate flows) are applicable to this use case.	

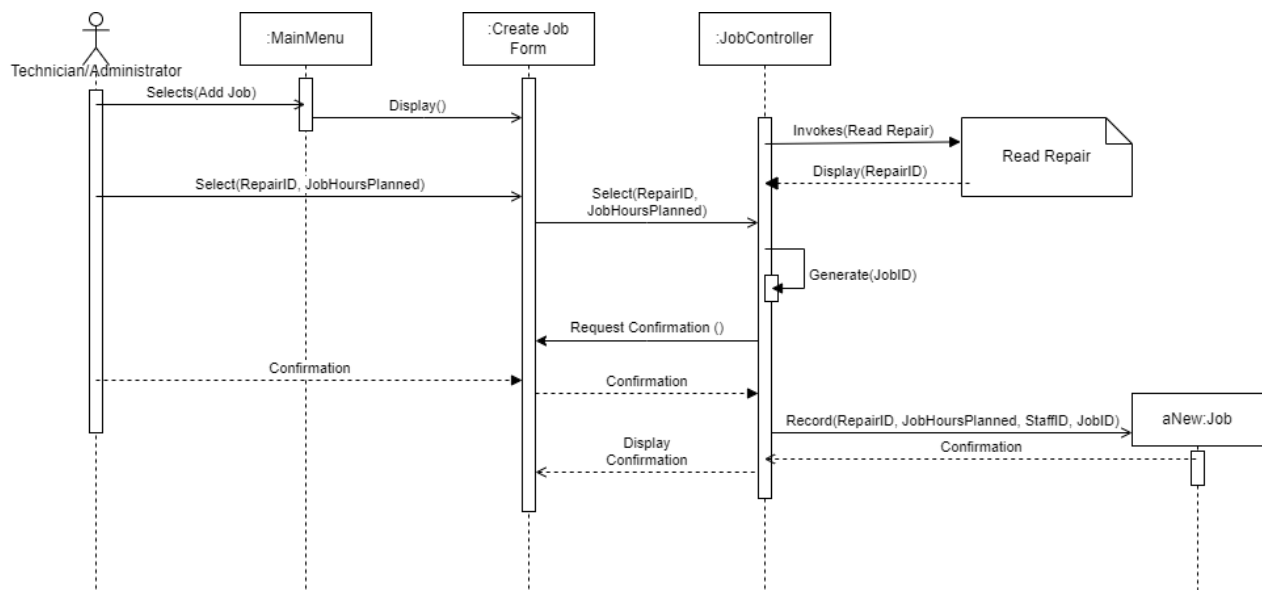
Sequence Diagram for Update Repair



Create Job

Use case name	Create Job.	
Triggering Event	The technician/administrator selects the option to add a job ticket for a repair.	
Actor(s)	Technician/Administrator	
Related Use Cases	Read Repair	
Pre-Conditions:	<ul style="list-style-type: none"> The chosen repair, and the technician must exist in their respective tables. 	
Post Conditions:	<ul style="list-style-type: none"> New job must be created in the Job Table using attributes JobID, staffID, RepairID, and JobHoursPlanned 	
Flow of Activities	Technician/Administrator	System
	<ol style="list-style-type: none"> The technician selects option to add their job for a repair. Selects the repair and enters the number of hours planned for it. Confirms job. 	<ol style="list-style-type: none"> 1.1 Prompts the user (technician) to select the repair they would like to add job hours for and enter the number of hours worked, (invokes Read Repair to display RepairID). 2.1 Generates a JobID. 2.2 Requests confirmation from user. 3.1 Records number of hours for that job, including: StaffID, RepairID and JobHoursPlanned (JobHoursWorked = 0) 3.2 Confirms Job creation
Extensions	No extensions (alternate flows) are applicable to this use case.	

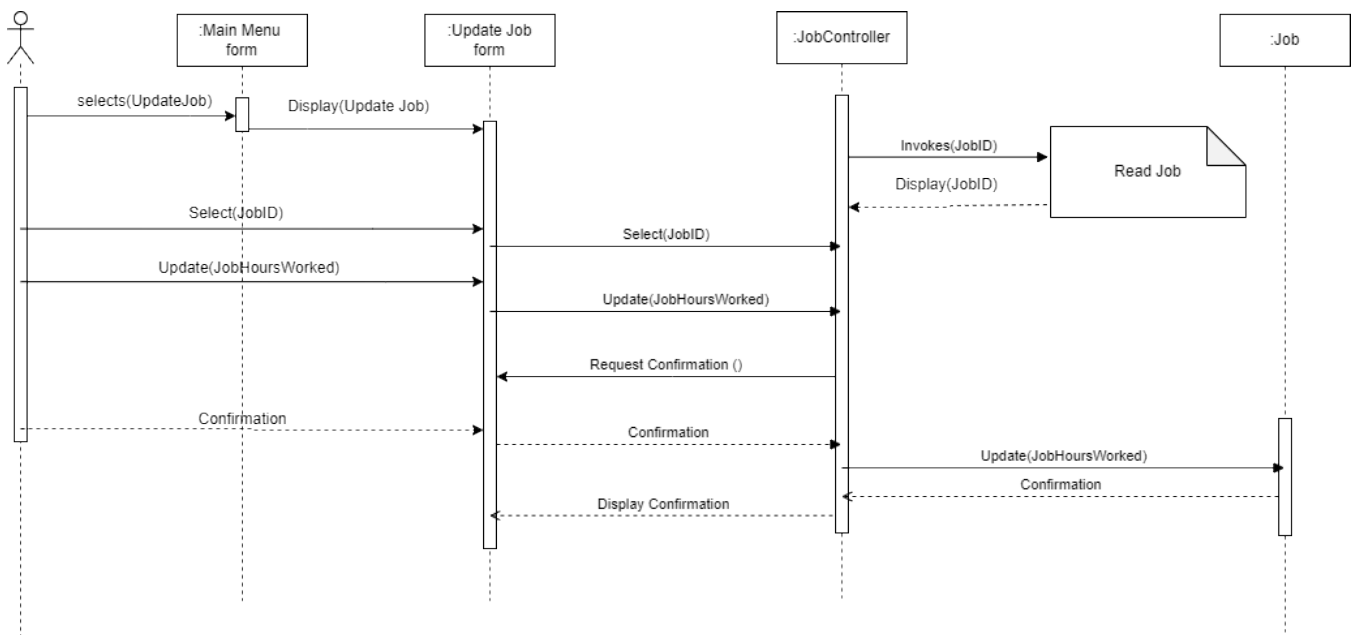
Sequence Diagram for Create Job



Update Job

Use case name	Update Job.	
Triggering Event	The technician/administrator selects the option to update job.	
Actor(s)	Technician/administrator	
Related Use Cases	Read job	
Pre-Conditions:	The job and the technician must exist in their respective tables.	
Post Conditions:	<ul style="list-style-type: none"> Job must be updated with the following attributes of JobHoursWorked. 	
Flow of Activities	Technician/administrator	System
	1. The user selects option to update their job for a repair.	1.1 Prompts the user to select the repair they would like to update job hours for and enter the number of hours worked (invokes Read Job).
	2. Selects the job and updates the number of hours for it.	2.1 Requests confirmation from user.
	3. Confirms job.	3.1 Records number of hours for JobHoursWorked
		3.2 Confirms Job updated
Extensions	No extensions (alternate flows) are applicable to this use case.	

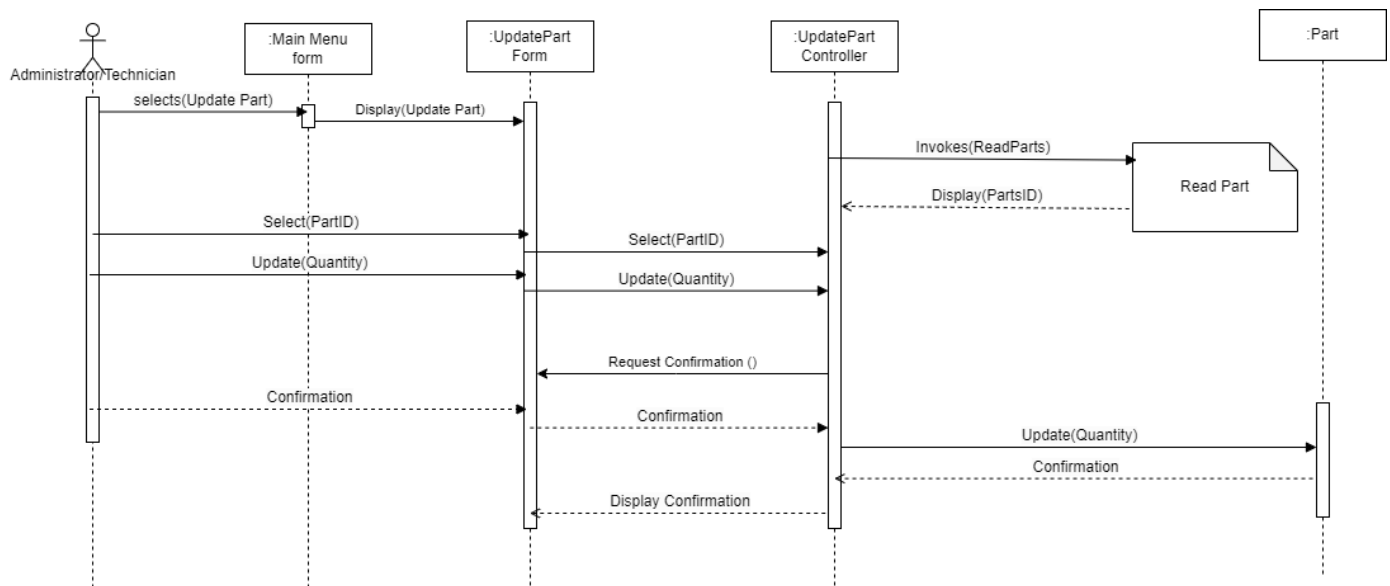
Sequence Diagram for Update Job



Update Part

Use case name	Update part.	
Triggering Event	The administrator/technician selects the option to update part.	
Actor(s)	Administrator/Technician	
Related Use Cases	Read part	
Pre-Conditions:	<ul style="list-style-type: none"> The part must exist in the Part table. 	
Post Conditions:	<ul style="list-style-type: none"> Part must be updated with the new quantity. 	
Flow of Activities	Administrator/Technician	System
	1. The user selects option to update part	1.1 Prompts the user to select the part they would like to update. (invokes Read Part).
	2. Selects the part and updates the quantity	2.1 Requests confirmation from user.
Extensions	3. Confirms part.	3.1 Records the updated quantity in the Parts table. 3.2 Confirms parts updated
	No extensions (alternate flows) are applicable to this use case.	

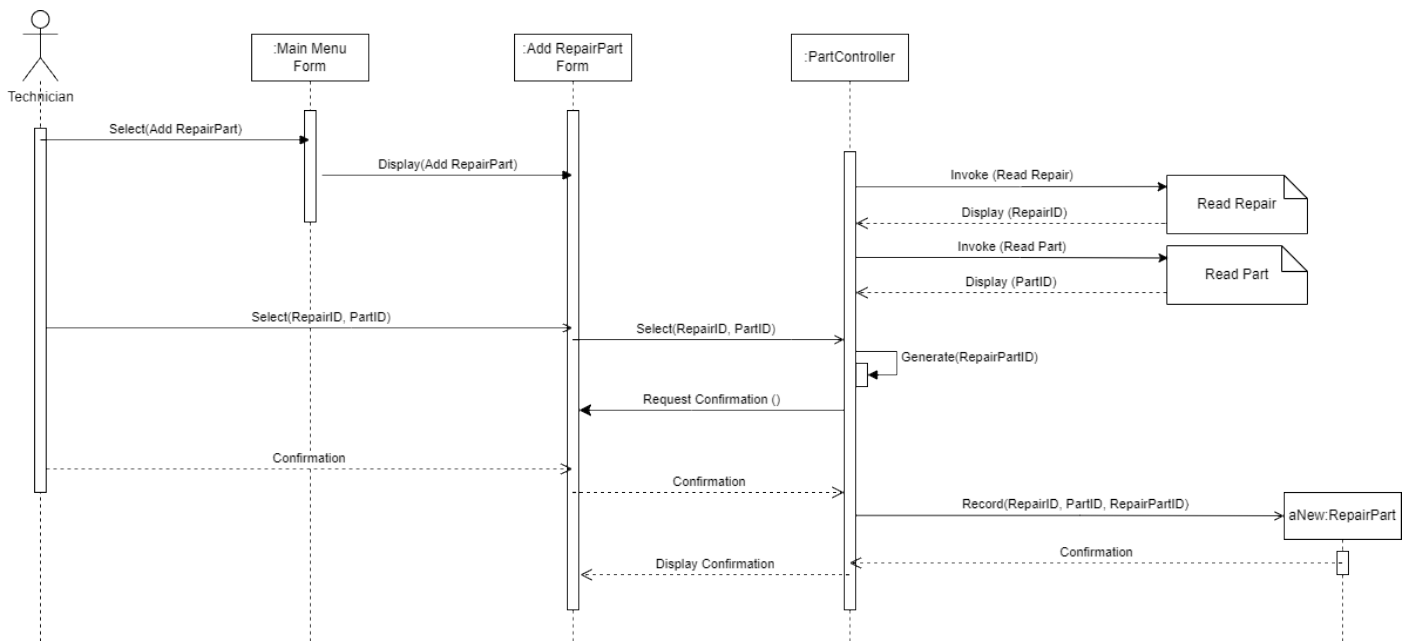
Sequence Diagram for Update Part



Create RepairPart

Use Case Name	Create RepairPart	
Triggering Event	The technician selects the option to add a new part for a repair.	
Actor(s)	Technician	
Related Use Cases	Read Repair, Read Part	
Pre-conditions	<ul style="list-style-type: none"> Repair, and Part must exist in their respective tables. 	
Post-conditions	<ul style="list-style-type: none"> New RepairPart must be created in the RepairPart Table using attributes RepairPartID, RepairID, and PartID, 	
Flow of Activities	Technician	System
	1. Requests to add a new part to a repair. 2. Selects the respective repair and part. 3. Confirms the part addition.	1.1 Prompts the user to select and enter information, including the following: RepairID (Invokes Read Repair) and PartID (Invokes Read Part). 2.1 Generates RepairPartID 2.2 Requests confirmation from user 3.1 Records repair-part information including RepairPartID, RepairID, and PartID. 3.2 Confirms repair creation.
Extensions	No extensions (alternate flows) are applicable to this use case.	

Sequence Diagram for Create RepairPart



Maintenance Use Cases

Read Repair Use Case

This use case is executed by the technician, administrator to view details of an existing repair. The user selects the option to view repair information in the system. The system displays a list of all ongoing and completed repairs (invoking the Read Repair Use Case). The user selects a repair from the list, and the system retrieves and displays detailed information about the selected repair.

Create Customer Use Case

The user selects the option to create a new customer account. The system prompts the user to enter the applicable details, including email, name, contact information, and address. After entering the required information, the user clicks create account. The system confirms the creation of the new account by displaying a confirmation message, and the customer is saved in the system.

Update Customer Use Case

This use case is executed by the admin to update a customer's information in the system. The user selects the option to update account information. The system displays the customer's current details and prompts them to make any required changes. After making the updates, the user submits the changes. The system records the updates and displays a confirmation message, ensuring that the customer's information is successfully updated in the system.

Read Customer Use Case

This use case is executed by the administrator to view the details of a customer.: The user selects the option to view customer information. The system displays a list of all customers (invoking the Read Customer Use Case). The user selects a customer from the list, and the system retrieves and displays detailed information about the selected customer.

Create Part Use Case

This use case is executed by the administrator or technician to add a new part to the system. The user selects the option to create a new part. The system prompts the user to enter the details of the part, including the part name, description, quantity, and any relevant specifications. After entering the required information, the user saves the record. The system generates a PartID, saves the part, and confirms the creation of the new part by displaying a confirmation message.

Read Part Use Case

This use case is executed by the administrator or technician to view the details of a part in the system. The user selects the option to view part information. The system displays a list of all parts (invoking the Read Part Use Case). The user selects a part from the list, and the system retrieves and displays detailed information about the selected part.

Update RepairPart Use Case

This use case is executed by the technician to update the details of parts assigned to a repair. The user selects the option to update a repair part. The system displays a list of repairs (invoking the Read Repair Use Case) and prompts the user to select a repair. After selecting the repair, the system shows the parts currently assigned to that repair (Invoking Read RepairPart). The user updates the relevant details. The system records the updates and displays a confirmation message, ensuring that the repair part information is successfully updated.

Read RepairPart Use Case

This use case is executed by the technician to view the parts assigned to a specific repair. The user selects the option to view repair parts. The system displays a list of repairs (invoking the Read Repair Use Case) and prompts the user to select a repair. After selecting the repair, the system retrieves and displays detailed information about the parts assigned to that repair (Invoking Read RepairPart).

Create Unit Use Case

This use case is executed by the administrator to add a new unit to the system. The user selects the option to create a new unit. The system prompts the user to enter details about the unit, including the brand, name, size, and any other relevant specifications. The system then generates a UnitID for the unit. After entering the required information, the user saves the record. The system confirms the creation of the new unit by displaying a confirmation message, ensuring that the unit is now registered in the system.

Update Unit Use Case

This use case is executed by the administrator to update the information of an existing unit in the system. The user selects the option to update a unit. The system displays a list of existing units (invoking the Read Unit Use Case) and prompts the user to select the unit to be updated. After selecting the unit, the user updates the applicable details. The system records the updates and displays a confirmation message, ensuring that the unit information is successfully updated in the system.

Read Unit Use Cases

This use case is executed by the administrator to view the details of an existing unit in the system. The user selects the option to view unit information. The user selects a UnitID from

the list (generated by invoking Read Unit), and the system retrieves and displays detailed information about the selected unit.

Create Staff Use Case

This use case is executed by the administrator to add a new staff member to the system. The administrator selects the option to create a new staff profile. The system prompts the administrator to enter details about the staff member, including name, address, department, and contact details. The system then generates a StaffID and confirms the creation of the new staff record by displaying a confirmation message, ensuring that the staff member is now registered in the system.

Update Staff Use Case

This use case is executed by the administrator to update the information of an existing staff member in the system. The administrator selects the option to update a staff record. The system displays a list of existing staff members with their StaffID's (invoking the Read Staff Use Case) and prompts the administrator to select the staff member to be updated. After selecting the staff member, the administrator updates the relevant details. The system records the updates and displays a confirmation message, ensuring that the staff member's information is successfully updated in the system.

Read Staff Use Case

This use case is executed by a staff member to view the details of an existing staff member in the system. The user selects the option to view staff information. The system displays a list of all staff members with their StaffID's (Invoking Read Staff). The user selects a staff member from the list, and the system retrieves and displays detailed information about the selected staff member.

Read Job Use Case

This use case is executed by the technician or administrator to view details of an existing job in the system. The user selects the option to view job information. The system displays a list of all jobs (Invoking Read Job) and prompts the user to select a specific job from the list. After selecting a job, the system retrieves and displays detailed information about the selected job.

Create Manufacturer Use Case

This use case is executed by the administrator to add a new manufacturer to the system. The administrator selects the option to create a new manufacturer. The system prompts the administrator to enter details about the manufacturer. A ManufacturerID is created by the system. After entering the required information, the administrator saves the record. The system confirms the creation of the new manufacturer by displaying a confirmation message, ensuring that the manufacturer is now registered in the system.

Update Manufacturer Use Case

This use case is executed by the administrator to update the information of an existing manufacturer in the system. The administrator selects the option to update a manufacturer. The system displays a list of existing manufacturers (invoking the Read Manufacturer Use Case) and prompts the administrator to select the manufacturer to be updated. After selecting the manufacturer, the administrator updates the relevant details. The system records the updates and displays a confirmation message, ensuring that the manufacturer's information is successfully updated in the system.

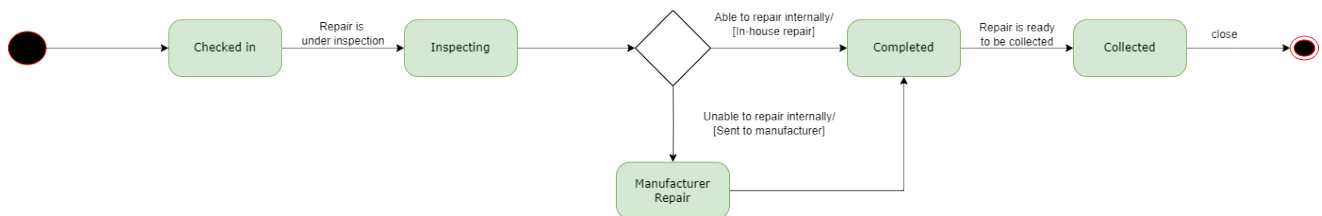
Read Manufacturer Use Case

This use case is executed by the administrator to view the details of an existing manufacturer in the system. The administrator selects the option to view manufacturer information. The administrator selects a manufacturer from the list (generated by invoking Read Manufacturer), and the system retrieves and displays detailed information about the selected manufacturer, ManufacturerID, ManufacturerName, address, and contact information.

State Machine Diagram

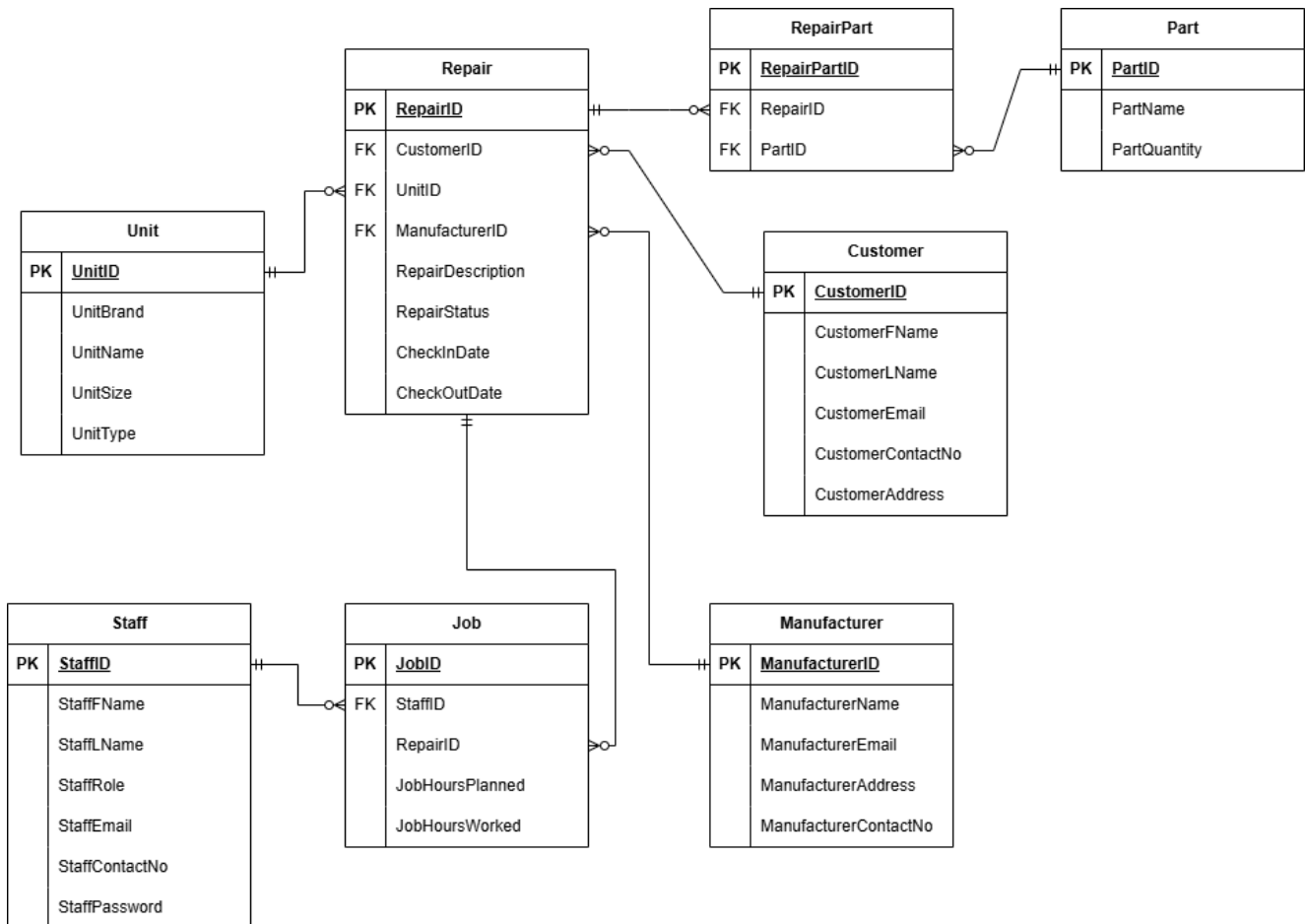
Repair Entity

As the repair entity is the only entity in the database that undergoes state changes, it is the only state machine diagram portrayed.

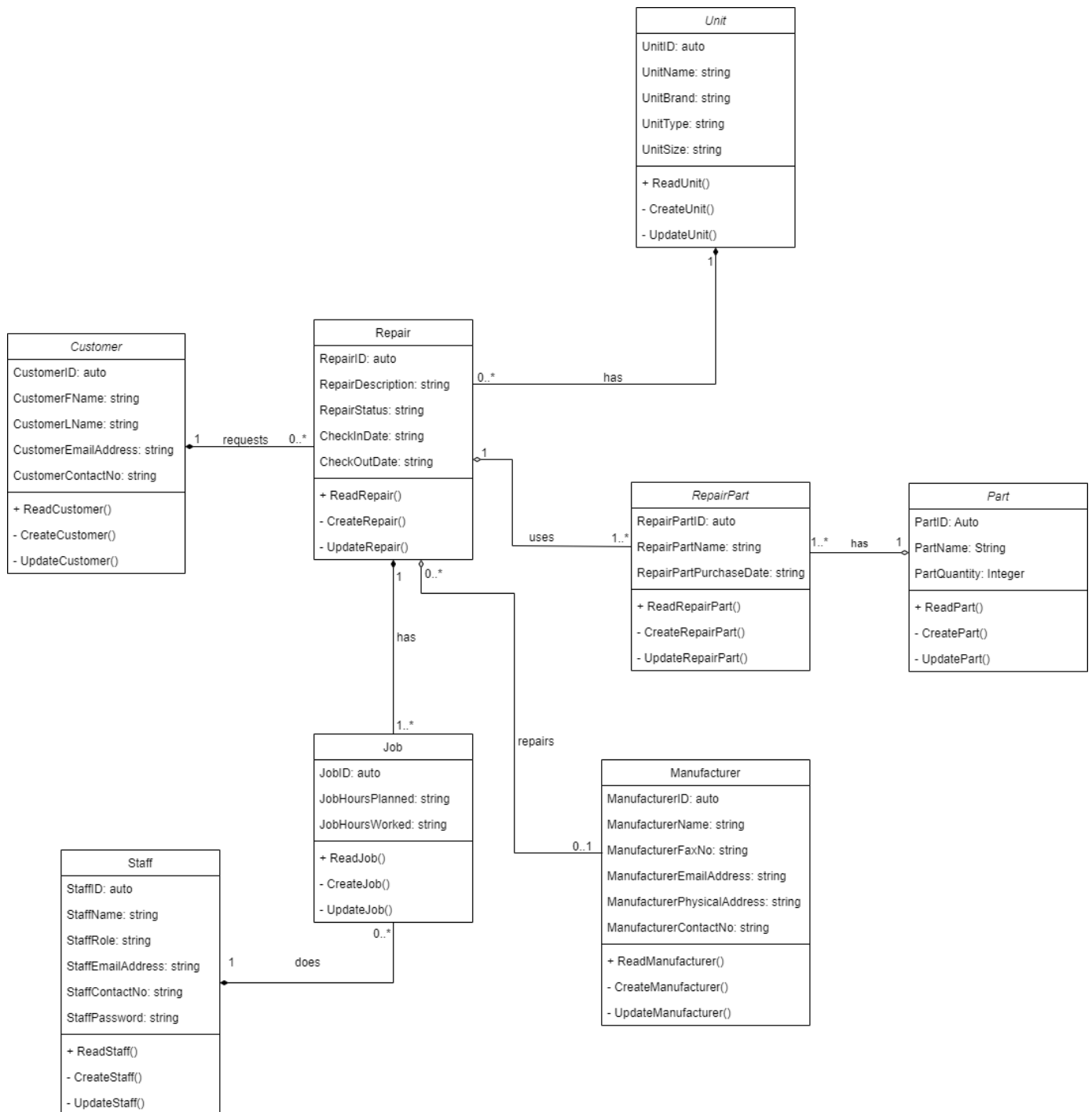


Domain and Data Models

Entity Relationship Diagram (ERD)



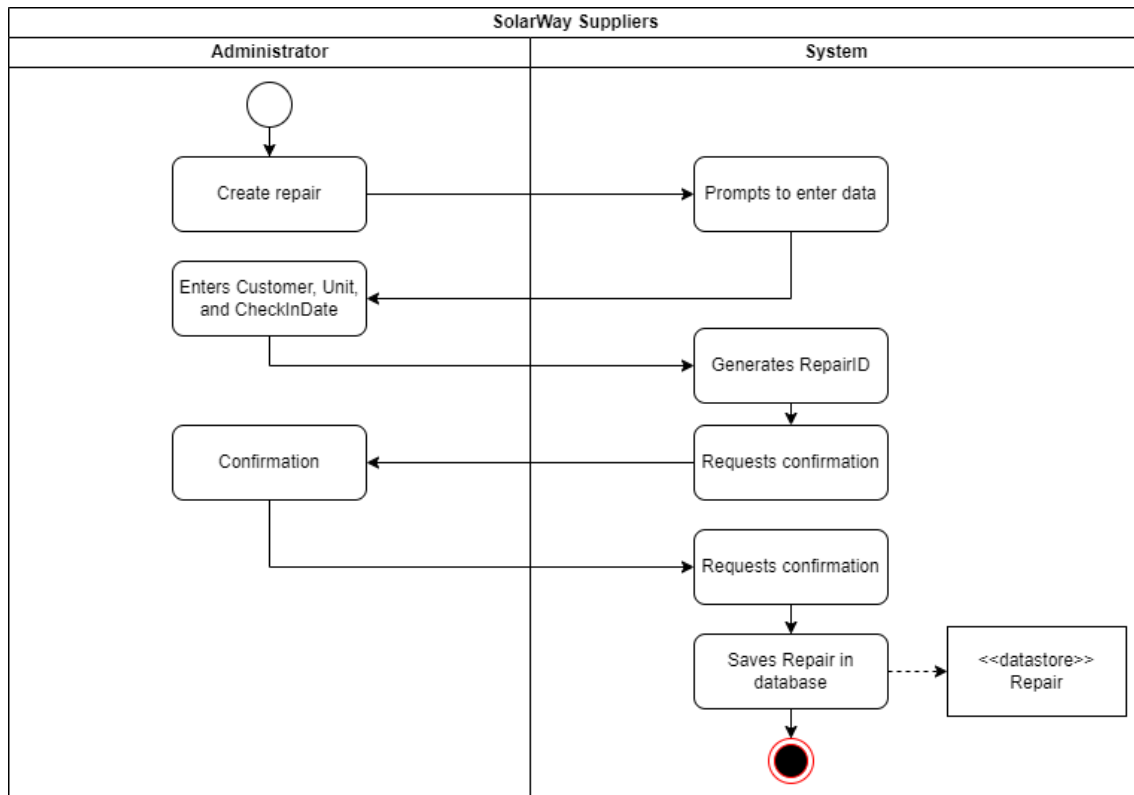
Class Diagram



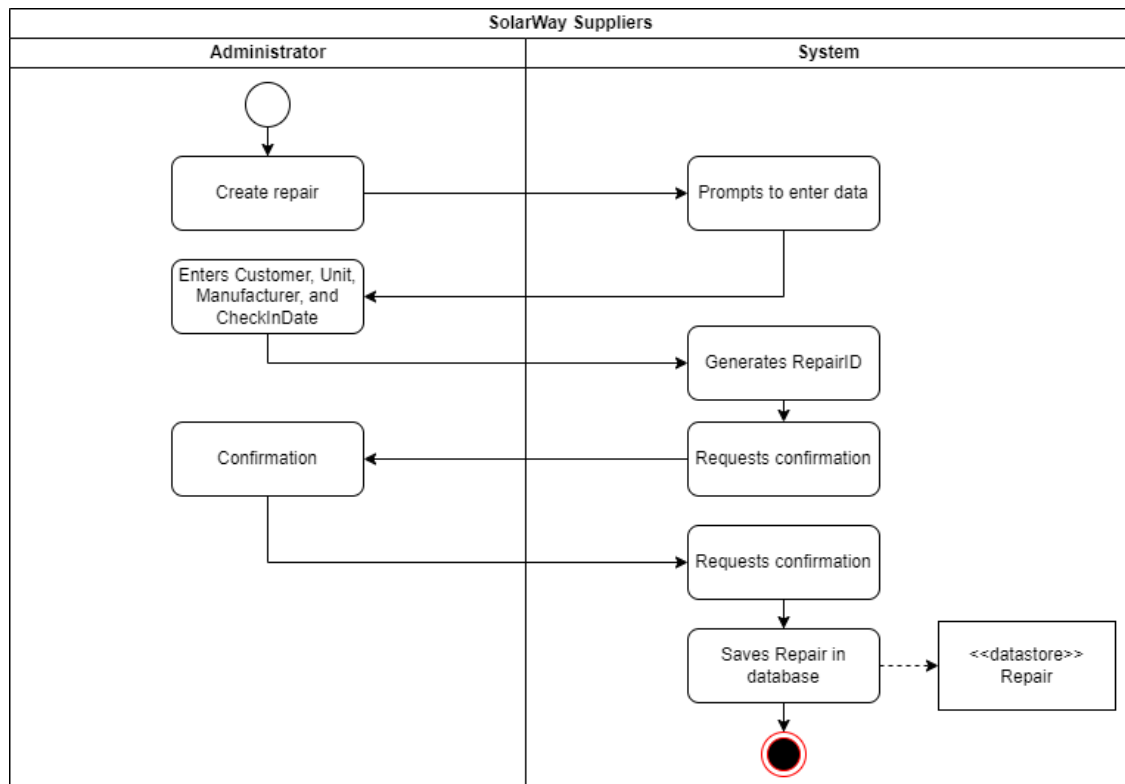
Business Process Modelling (Activity Diagrams)

Create Repair

In-house



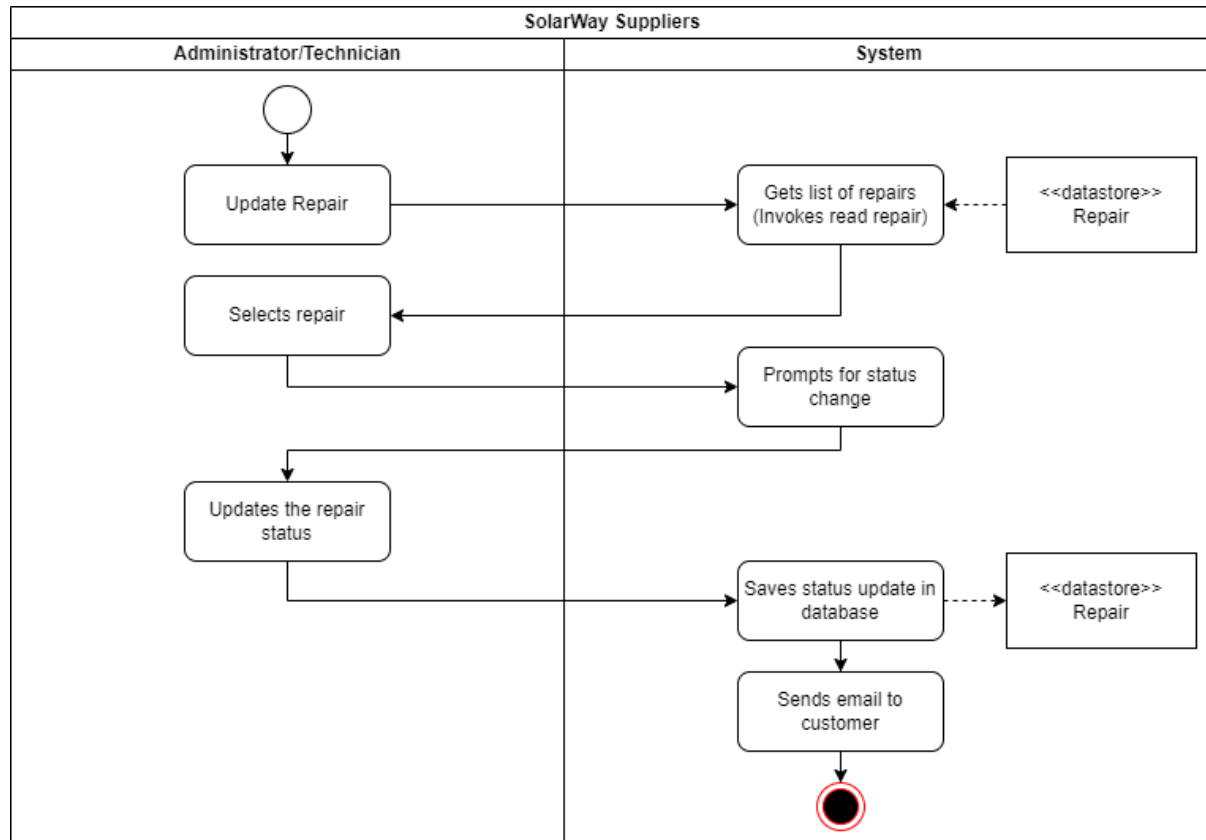
Manufacturer



Explanation

The create repair activity diagram models the flow of the use case as activities that are undertaken. There are 2 diagrams depending on the 2 use case flows, either in-house or manufacturer repair. The flow follows the use case description – the user selects the option to add a repair, they then enter the required details, followed by confirming the repair, it is then saved in the database.

Update Repair

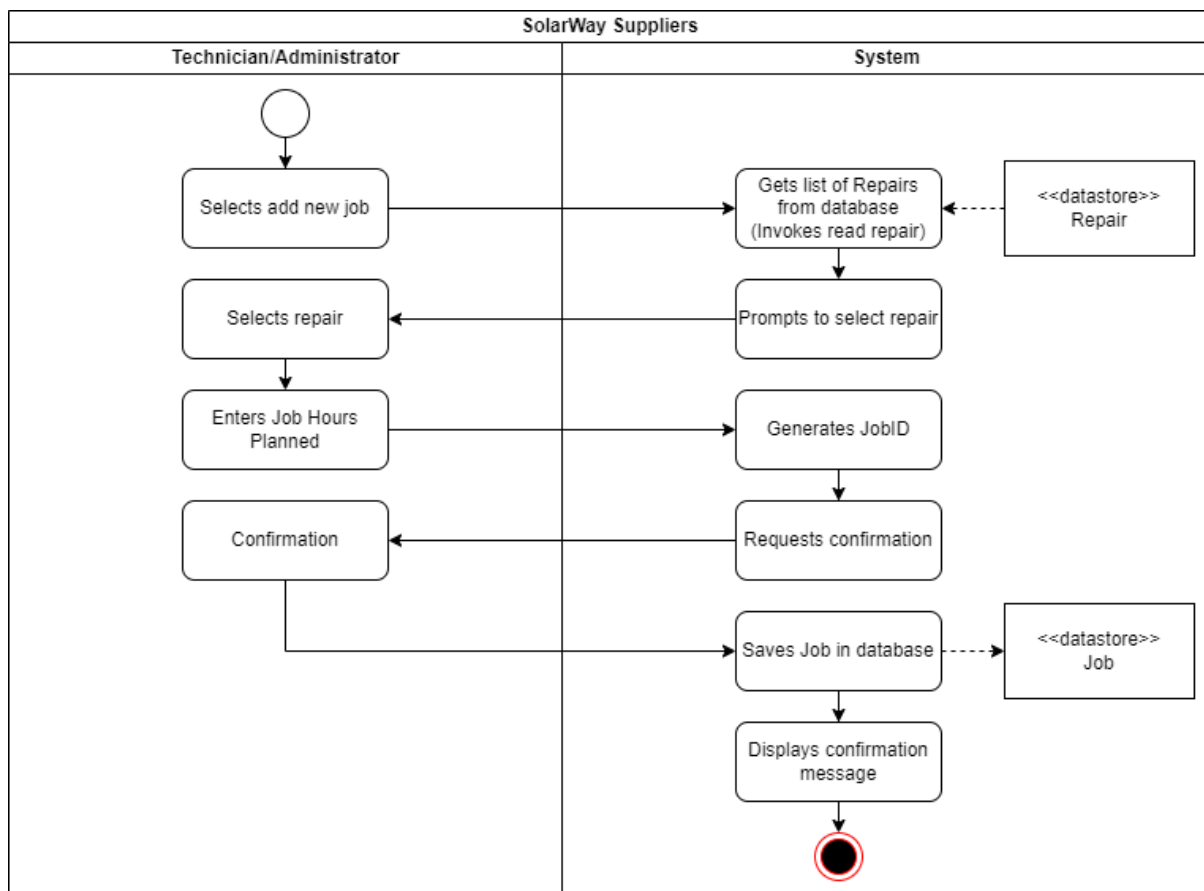


Explanation

This activity diagram models the flow of the update repair use case. The user updates the status of the repair to ensure the life cycle of the repair is accurately tracked.

When updating the status, the user selects a repair from the list, which is taken from the database, they then enter the new status and update the repair, finally, an email is sent to the customer reflecting the status change.

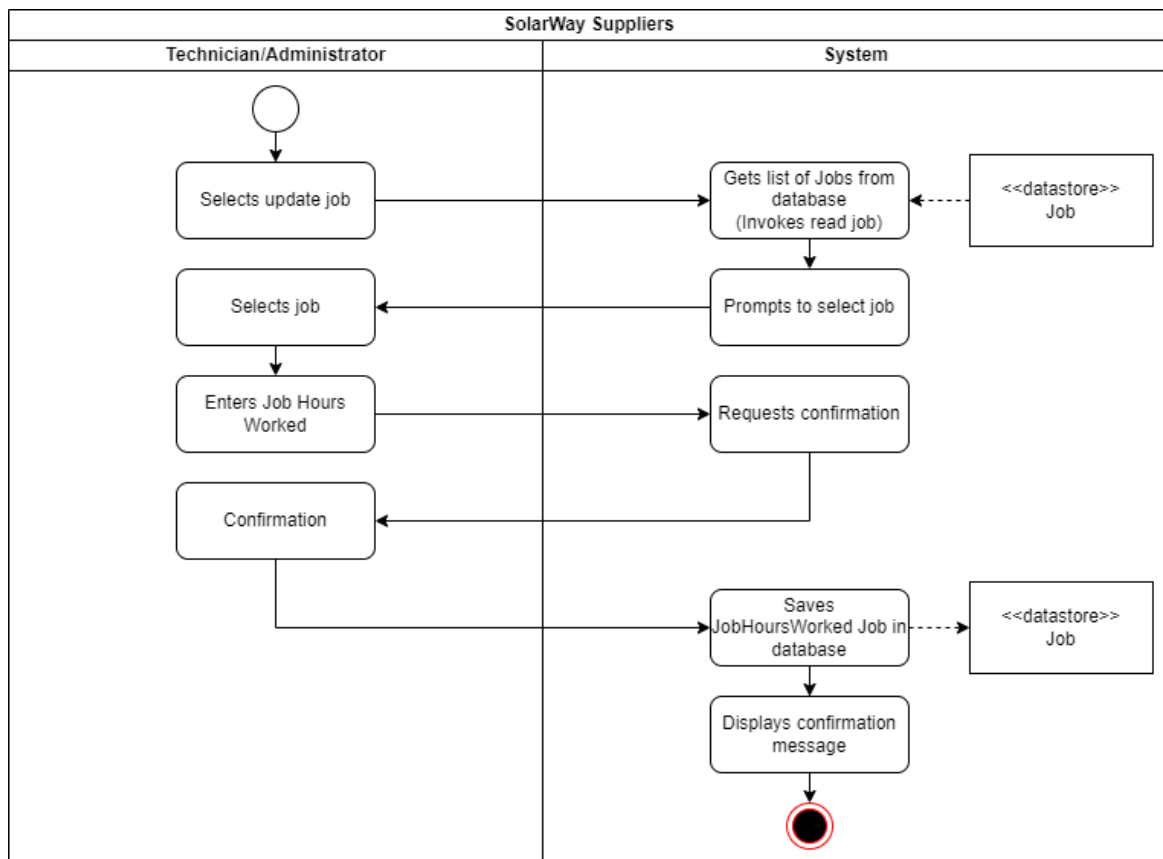
Create Job



Explanation

This activity diagram models the create job use case. The flow is simple here and models the use case description and sequence diagram. The user selects the option to create a job ticket, they then select the repair they are working on, which is taken from the active repairs in the database, and the number of hours they have worked. Following a confirmation, the job is saved in the database.

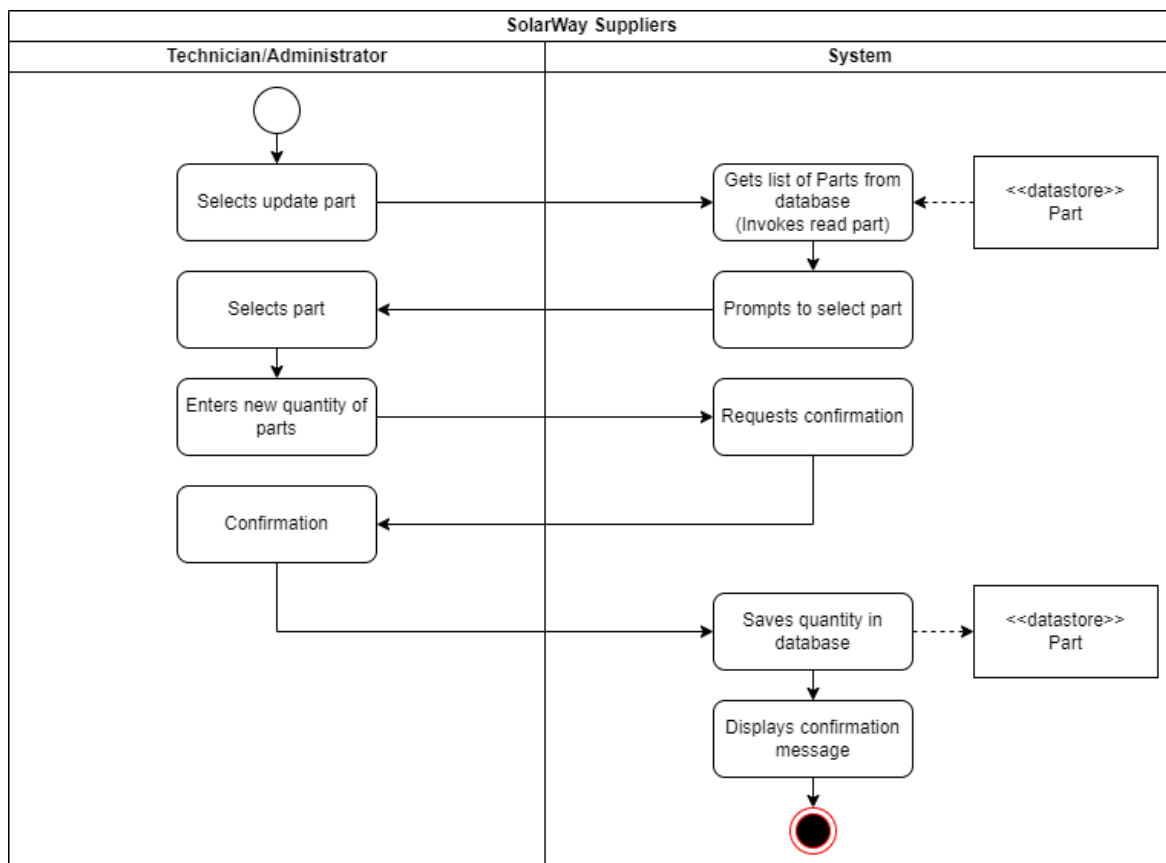
Update Job



Explanation

This activity diagram models the update job use case. Here, the technician or administrator adds the actual hours worked to a job. Firstly, they select a job from the existing jobs taken from the database, they then enter the hours worked and save it, the system finally adds this update to the system database.

Update Part

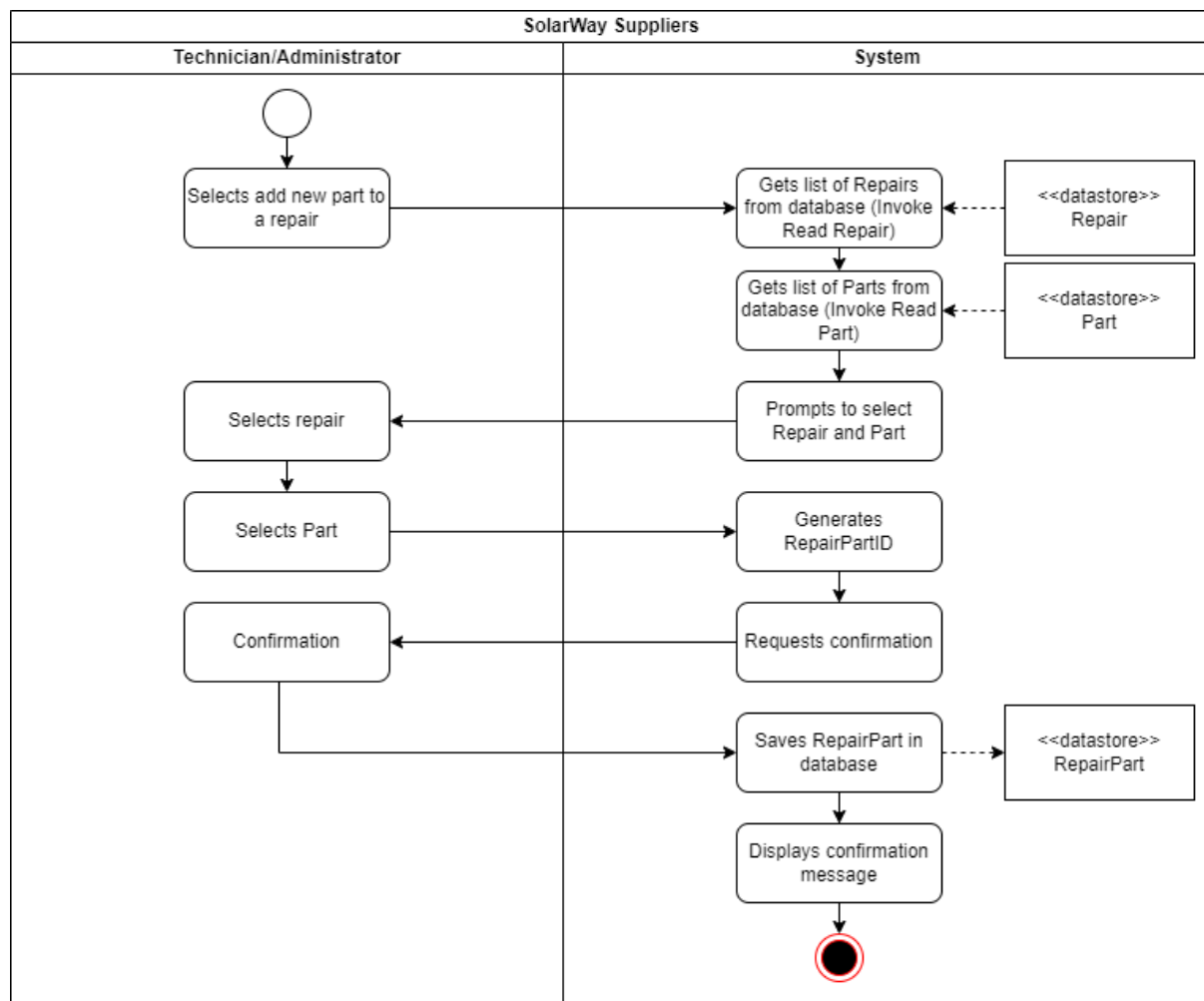


Explanation

This activity diagram models the update part use case. This use case is run when the quantity of a part needs to be updated when it is used, a stock count is taken, or a new order of parts arrive.

The administrator or technician first select the part to update from a list of parts in the database provided by the system. They then enter the new quantity of the part, and after confirming, the system updates the quantity attribute in the database.

Create RepairPart



Explanation

This activity diagram follows the logic of the create RepairPart use case. This use case is run when a technician needs to add a part used to a repair.

The logic is as follows: the technician selects a part and repair from a list provided by the system, which is gathered from the database by invoking the read repair and read part use cases. The technician then confirms the part and repair, after which the system saves it in the RepairPart table in the database.

System Feature List

There are several features that are not captured by use cases that the system will incorporate:

- **Navigation Bar:** The system will have a navigation bar for the different types of users, this will allow users easy navigation of the application.
- **Security Features:** The system will have a rigorous password protection feature to ensure that no unauthorized entities access the sensitive information on the system.
- **Input Validation:** The system will incorporate input validation; this is to ensure that user input errors are minimized.