
Introduction to Software Engineering

Coursework 1 Report

Heriot-Watt Dubai Campus

Student details

Zondwayo Mtine

H00373945

Zm2016@hw.ac.uk

Table of Contents

T1: Assumptions and Expectations:.....	1
T2: Functional and Non-functional Requirements:	2
T3: Use Case Model and Specifications:	3
T4: Traceability Matrix:.....	7
T5: Class Diagram:	8
T6: Sequence Diagram:.....	9
T7: Activity Diagram(s):	10
T8: State Machine Diagram:	12
T9: Scenario Test Cases:	13



T1: Assumptions and Expectations:

Assumption 1

The VHS system is assumed to have a reliable and efficient system for managing and updating vehicle and rental records.

Assumption 2

The VHS system is assumed to have a user-friendly interface for customers and garage technicians.

Assumption 3

The VHS is assumed to be capable of archiving completed rental records daily without manual intervention.

Assumption 4

The VHS system is assumed to have a system for managing and updating the records of garage technicians.

Assumption 5

The VHS system is assumed to have a system for managing and updating the status of vehicles.

Assumption 6

The VHS system is assumed to have a system for managing and updating the records of the touch screen devices in the reception and inspection areas.

Expectation 1

The VHS system is to have a secure and efficient system for processing and managing payments.

Expectation 2

The VHS software is to have receive regular updates for performance improvements and security enhancements.

Expectation 3

Adequate support is to be available for users' facing issues with the VHS interface.

Expectation 4

The VHS is to adhere to applicable data protection regulations in handling customer information.

Expectation 5

Garage technicians are to receive clear and timely notifications for vehicle returns and inspections.

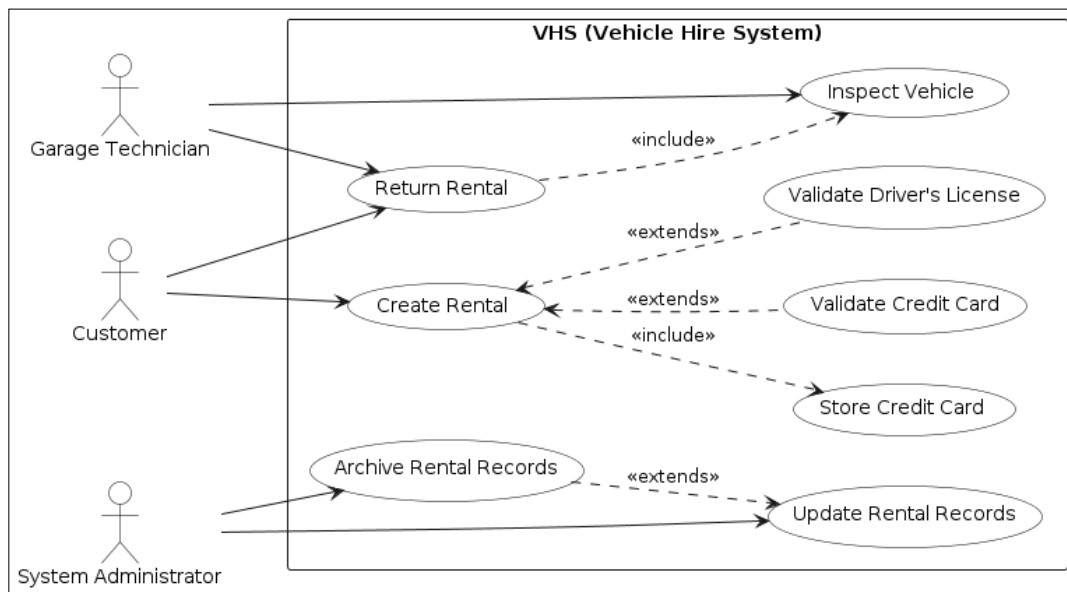
Expectation 6

Transactions and validations are to be processed within reasonable time frames.

T2: Functional and Non-functional Requirements:

<u>Functional Requirements</u>	<u>Non-Functional Requirements</u>
1.The system shall validate the customer's driving license details using the external service provided by the Driver and Vehicle Licensing Agency (DVLA).	1.The system should have a response time of less than 3 seconds for user interactions.
2.The system will be able to allow garage technicians to update the rental record after an inspection, which includes, recording missing fuel, damage to the vehicle, repair charges, and the vehicle's up-to-date mileage.	2.The system shall be reliable and resilient, capable of handling high volumes of rental transactions without affecting or slowing down the systems capabilities.
3.The system will maintain an internal record of the pool of vehicles available at its associated site, including the vehicle's registration number, current mileage, category, colour, and status.	3.The system shall be scalable, capable of handling an increase in the number of customers, vehicles, and rental transactions over time.
4.The system shall store and manage a set of customer records internally, including their unique ID number, name, date of birth, address, mobile number, email address, driving license number, associated category, expiry date, and optionally, credit card details.	4.Backup and recovery mechanisms shall be in place to ensure data integrity and availability in case of system failures.
5.The system will allow customers to request a rental via a touch screen located within the reception area of the associated site.	5.System performance shall scale proportionally with increasing load to maintain responsiveness and efficiency.
6.The system will maintain a rental record for each rental, including a reference to the customer, a reference to the vehicle, and the start and end dates for the rental period.	6.Sensitive information shall be encrypted during transmission and storage to prevent unauthorized access.

T3: Use Case Model and Specifications:



User ID: Create Rental

ID: 01

Goal: Allow customers to initiate the rental process for a vehicle.

Primary Actor: Customer

Secondary Actor: System Administrator

Preconditions:

1. The customer is logged in
2. The vehicle is available

Postconditions:

Rental record is created and stored in the system.

Main flow:

1. Customer selects desired vehicle for rental.
2. Customer provides necessary rental information (e.g., rental duration).
3. The system verifies customer's account and reserves the selected vehicle.
4. The system notifies the System Administrator to validate and finalize the rental.
5. The system generates rental agreement and sends it to the customer for confirmation.

Alternate flow:

If the customer's account cannot be verified, the system prompts the customer to update their details.

User ID: Validate Credit Card

ID:02

Goal: Verify the validity of the customer's credit card.

Primary Actor: System

Secondary Actor: N/A

Preconditions:

Customer initiates rental process.

Postconditions:

Credit card validation status is recorded in the system.

Main flow:

1. System prompts customer to provide credit card details.
2. The system validates credit card information using a secure payment gateway.
3. The system confirms credit card validity to the customer.

Alternate flow:

If credit card validation fails, the system notifies the customer and prompts for alternative payment methods.

User ID: Validate Driver's License	User ID: Archive Rental Records
ID: 03	ID: 04
Goal: Ensure the customer possesses a valid driver's license.	Goal: Automatically archive completed rental records.
Primary Actor: System	Primary Actor: System Administrator
Secondary Actor: N/A	Secondary Actor: N/A
Preconditions: Customer initiates the rental process.	Preconditions: Rental records are marked as completed.
Postconditions: Driver's license validation status is recorded in the system.	Postconditions: Completed rental records are successfully archived.
Main flow: 1.The system directs customer to upload a scanned copy of their driver's license. 2.The system verifies the authenticity and validity of the uploaded license. 3.The system confirms license validation to the customer.	Main flow: 1.The system identifies completed rental records. 2.The system archives the completed records. 3.The system sends confirmation of archival to the System Administrator.
Alternate flow: If license validation fails, system notifies the customer and asks for valid documentation.	Alternate flow: If archiving fails for any reason, System Administrator is alerted to intervene manually.

User ID: Return Rental
ID: 05
Goal: Enable customers to return rented vehicles.
Primary Actor: Customer
Secondary Actor: Garage Technician
Preconditions: Customer has an active rental. Vehicle is returned in agreed rental duration.
Postconditions: Vehicle is marked as returned in the system.
Main flow: 1.The customer arrives at the designated return location. 2.The customer notifies the system of the return. 3.The garage technician inspects the returned vehicle for damages. 4.System updates the rental record to mark the vehicle as returned. 5.Customer receives confirmation of return.
Alternate flow: If the vehicle is damaged or has issues, Garage Technician reports it, and system updates the rental record.

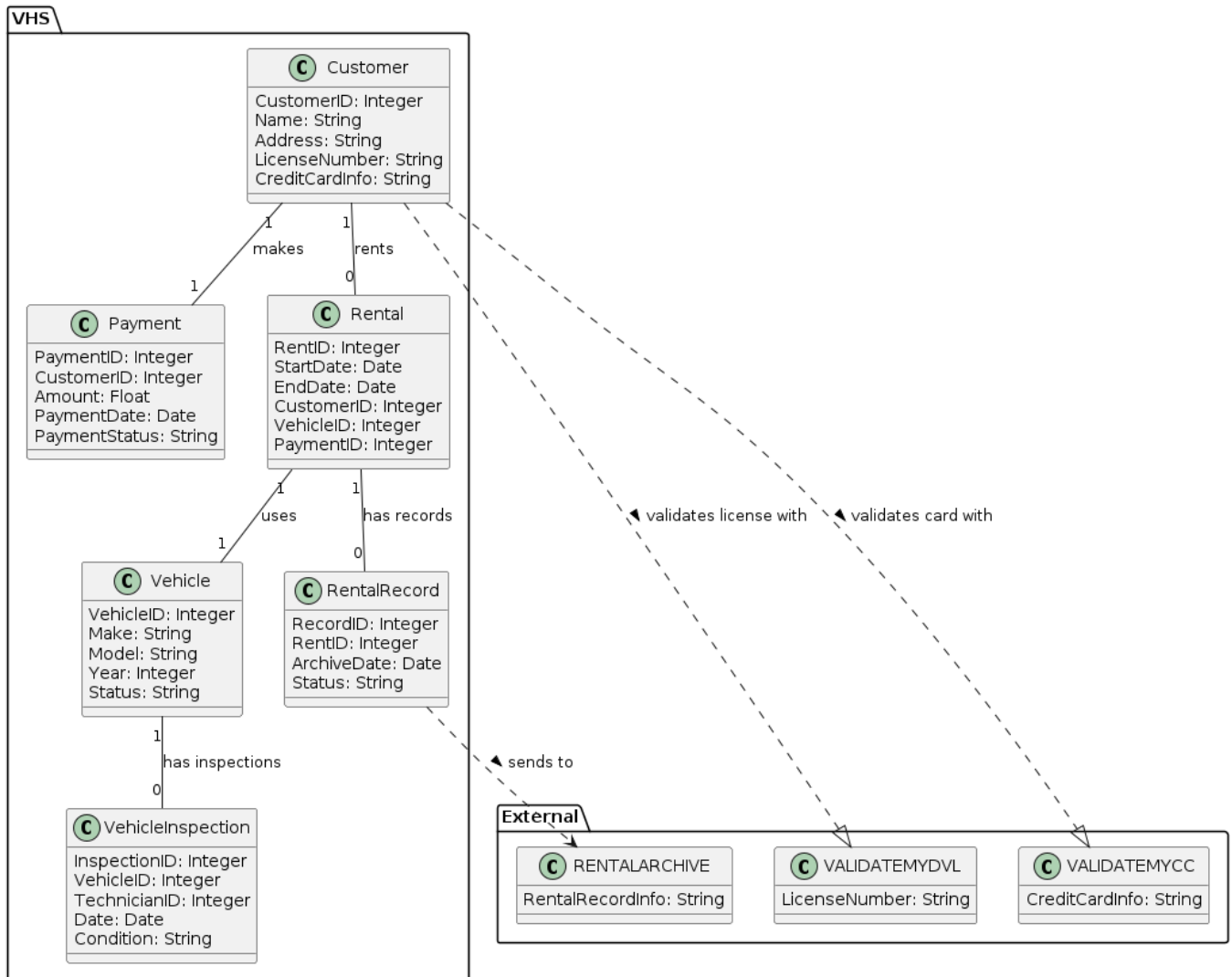
User ID: Inspect vehicle
ID: 06
Goal: Facilitate the inspection of vehicles returned by customers.
Primary Actor: Garage Technician
Secondary Actor: System
Preconditions: Vehicle is returned by the customer.
Postconditions: Inspection findings are recorded in the system.
Main flow: 1.The garage technician receives notification of vehicle return. 2.The garage technician inspects the vehicle for damages, cleanliness, and fuel level. 3.The garage Technician records inspection findings in the system.
Alternate flow: If damages are found, the garage technician reports it for further action.

User ID: Update Rental Records	User ID: Store Credit Card Details
ID: 07	ID: 08
Goal: Allow for the updating of rental records with new information.	Goal: Allow customers to securely store their credit card information for future transactions.
Primary Actor: System Administrator	Primary Actor: Customer
Secondary Actor: N/A	Secondary Actor: System
Preconditions: System administrator is logged into the system.	Preconditions: The customer is logged into the system. The customer initiates a rental process or payment transaction.
Postconditions: Rental record is updated with new information.	Postconditions: The credit card details are securely stored in the system.
Main flow: 1.The system Administrator accesses rental record requiring updates. 2.The system Administrator modifies rental information (e.g., rental duration). 3.The system updates the rental record with the new information.	Main flow: 1.Customer selects the option to store credit card details for future use. 2.The system directs customer to enter credit card information. 3.The customer provides credit card details. 4.The system securely stores the credit card information in an encrypted format. 5.The system confirms successful storage of credit card details to the customer.
Alternate flow: If the rental record cannot be updated, system administrator is notified for further investigation.	Alternate flow: If the system encounters an error during the storage process, it notifies the customer and informs them to try again later.

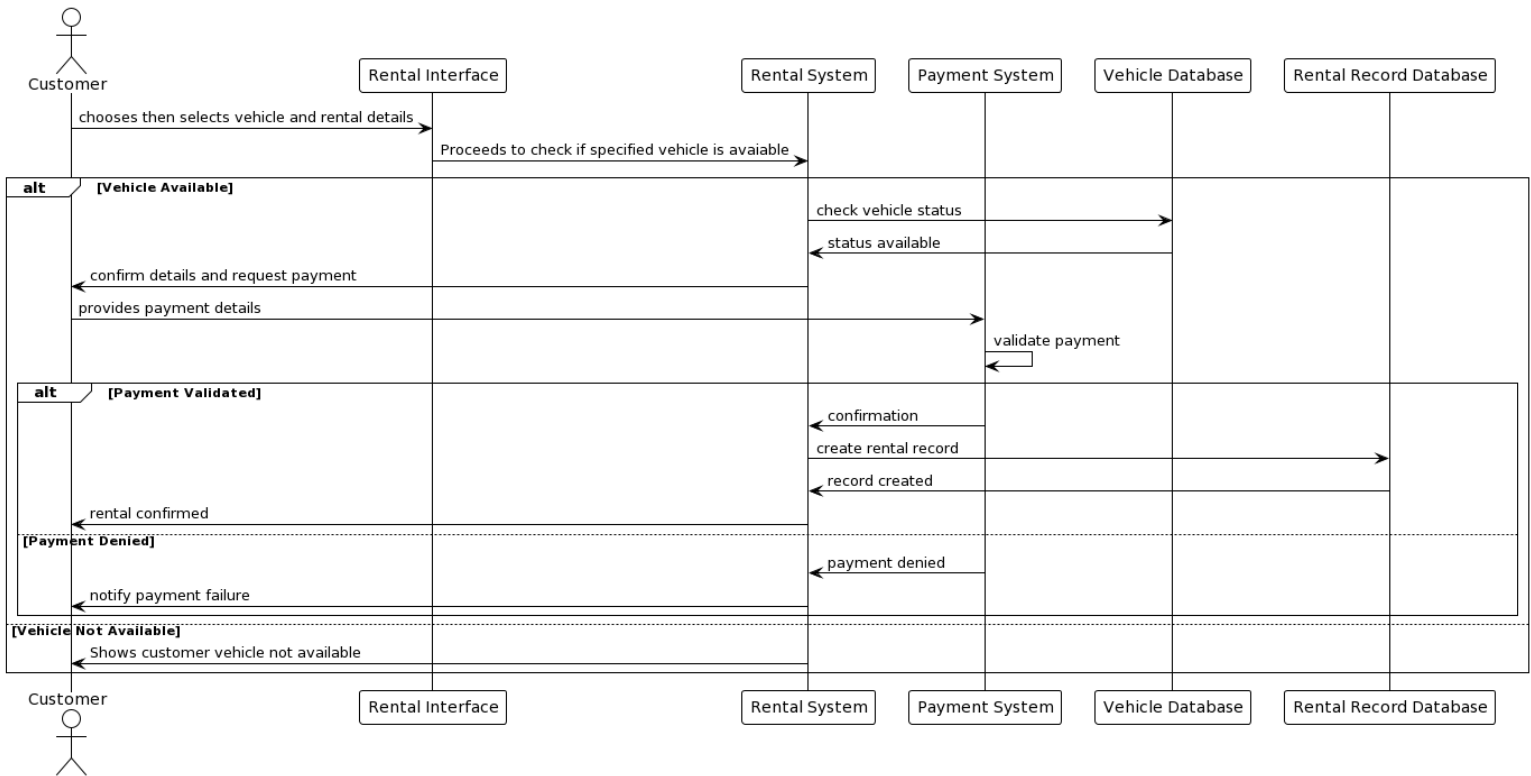
T4: Traceability Matrix:

Functional Requirement	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8
FR 1	X		X					
FR2						X	X	
FR3					X			
FR4	X							X
FR5	X							
FR6	X				X		X	

T5: Class Diagram:

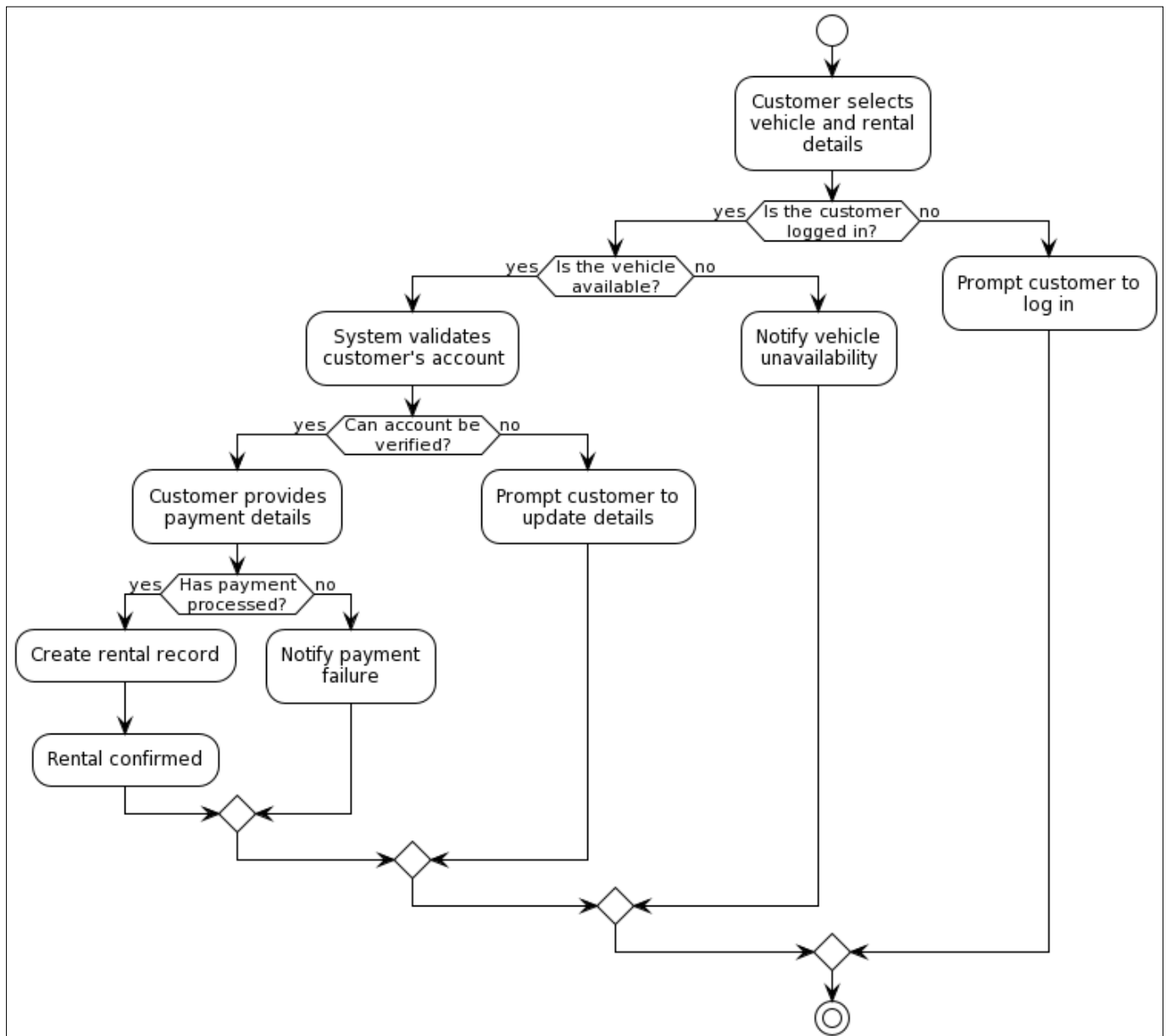


T6: Sequence Diagram:

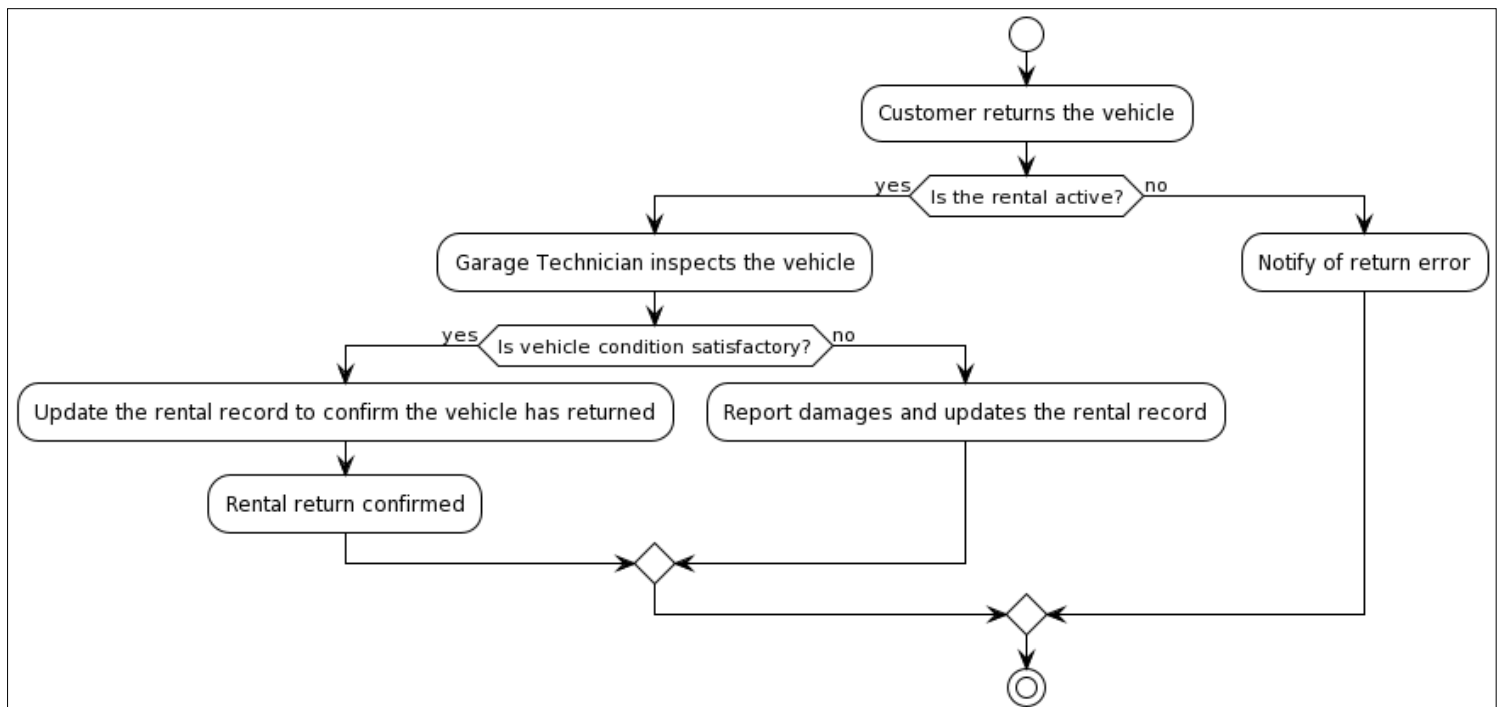


T7: Activity Diagram(s):

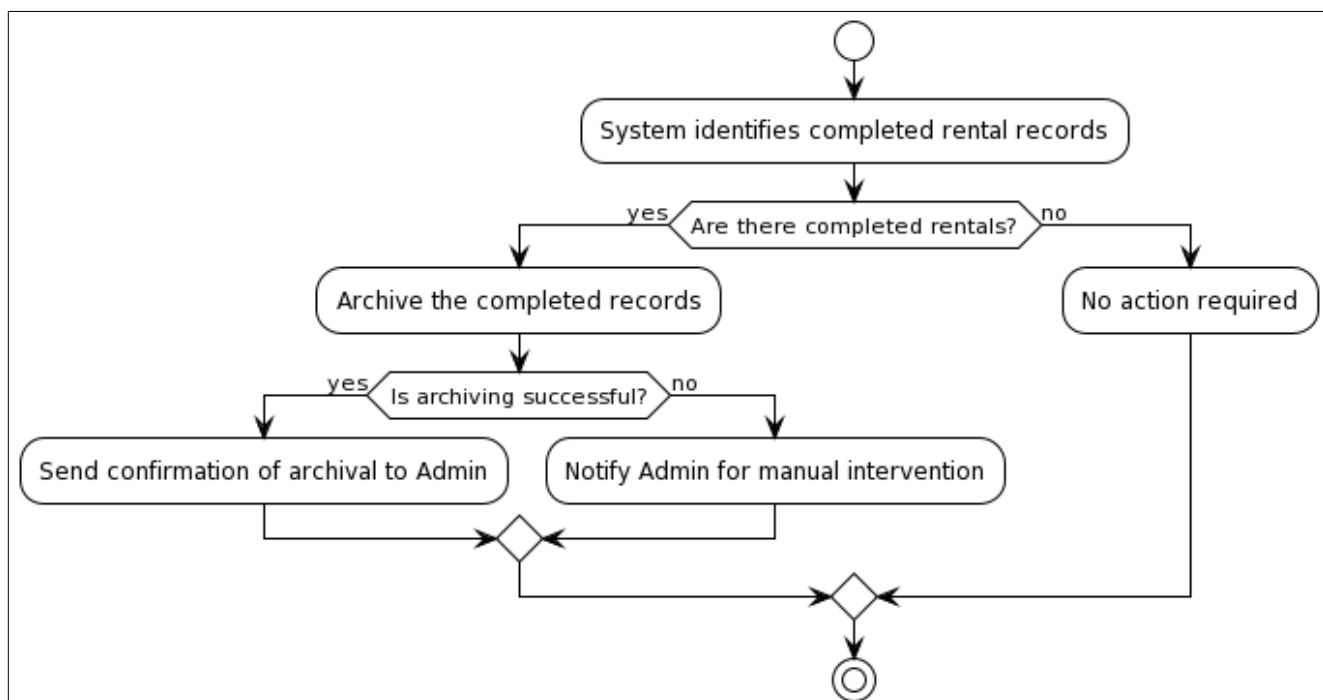
Create Rental



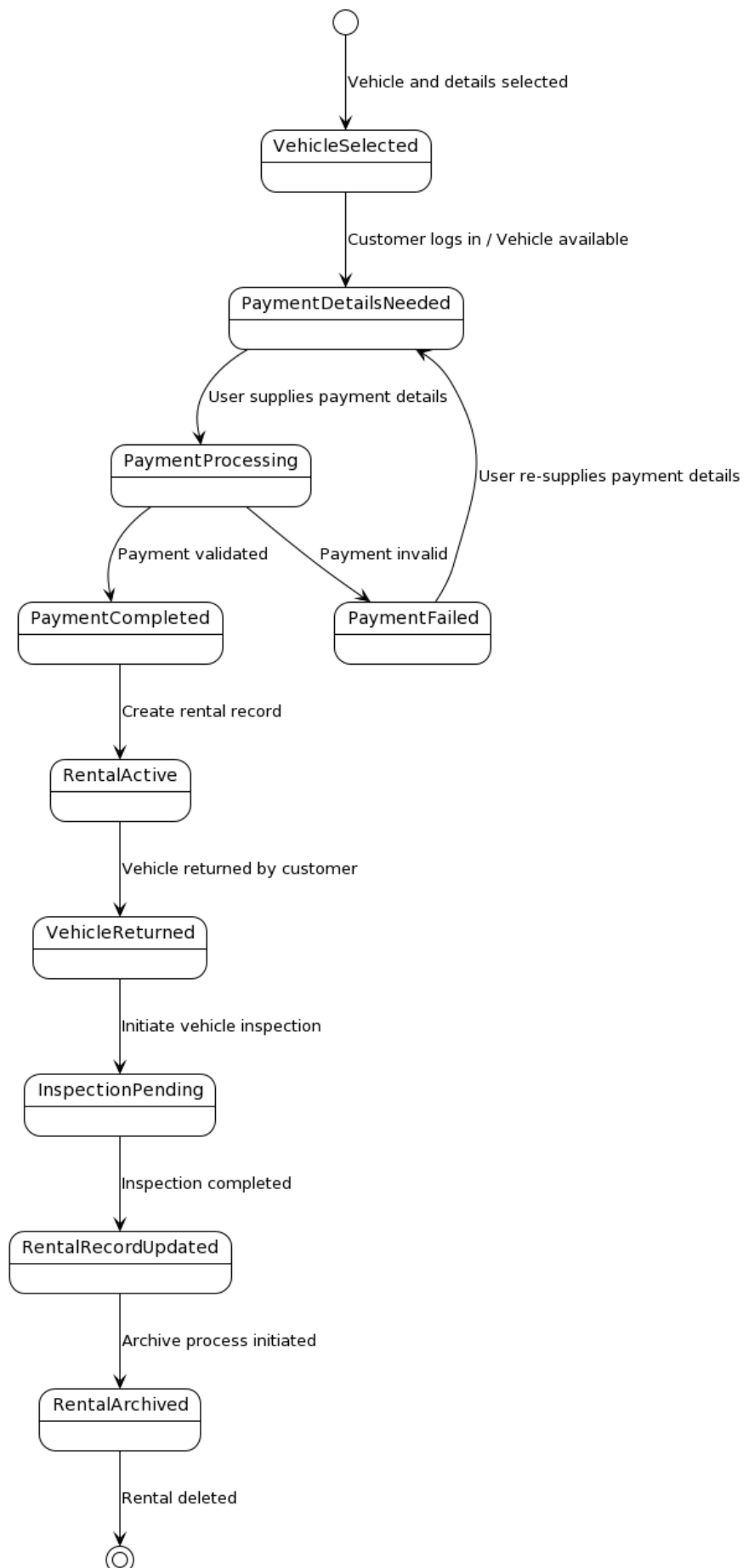
Return Rental



Archive Rental



T8: State Machine Diagram:



T9: Scenario Test Cases:

Test Case 1: Successful Rental Creation

Description: This test case covers a successful rental creation where all conditions are met.

Customer selects vehicle and rental details.

The customer is already logged in.

The vehicle is available.

The system validates the customer's account successfully.

Customer provides payment details.

Payment is successfully processed.

A rental record is created.

Rental confirmed.

Test Case 2: Vehicle Unavailability

Description: This scenario tests the system's response when the selected vehicle is not available.

Customer selects vehicle and rental details.

The customer is already logged in.

The vehicle is not available.

Notify vehicle unavailability.

Path: Customer selects vehicle and rental details -> Is the customer logged in? (yes) -> Is the vehicle available? (no) ->

Notify vehicle unavailability -> End

Test Case 3: Customer Not Logged In

Description: This test case examines the path when the customer has not logged in before attempting a rental.

Customer selects vehicle and rental details.

The customer is not logged in.

Prompt customer to log in.

Path: Customer selects vehicle and rental details -> Is the customer logged in? (no) -> Prompt customer to log in -> End

Test Case 4: Account Verification Failure and Update

Description: Tests the scenario where the customer's account cannot be verified, prompting an update.

Customer selects vehicle and rental details.

The customer is already logged in.

The vehicle is available.

The system cannot validate the customer's account.

Prompt customer to update details.

Path: Customer selects vehicle and rental details -> Is the customer logged in? (yes) -> Is the vehicle available? (yes) -> System validates customer's account -> Can account be verified? (no) -> Prompt customer to update details -> End

Test Case 5: Payment Failure and Retry

Description: This scenario simulates a payment failure and the subsequent actions taken by the customer.

Customer selects vehicle and rental details.

The customer is already logged in.

The vehicle is available.

The system validates the customer's account successfully.

Customer provides payment details.

Payment fails.

Customer re-supplies payment details (assuming the retry might succeed).

Path: Start -> Customer selects vehicle and rental details -> Is the customer logged in? (yes) -> Is the vehicle available? (yes) -> System validates customer's account -> Can account be verified? (yes) -> Customer provides payment details -> Has payment processed? (no) -> Notify payment failure -> Customer re-supplies payment details -> End (assuming this leads back to payment processing)