Requirement Analysis

Modeling Document

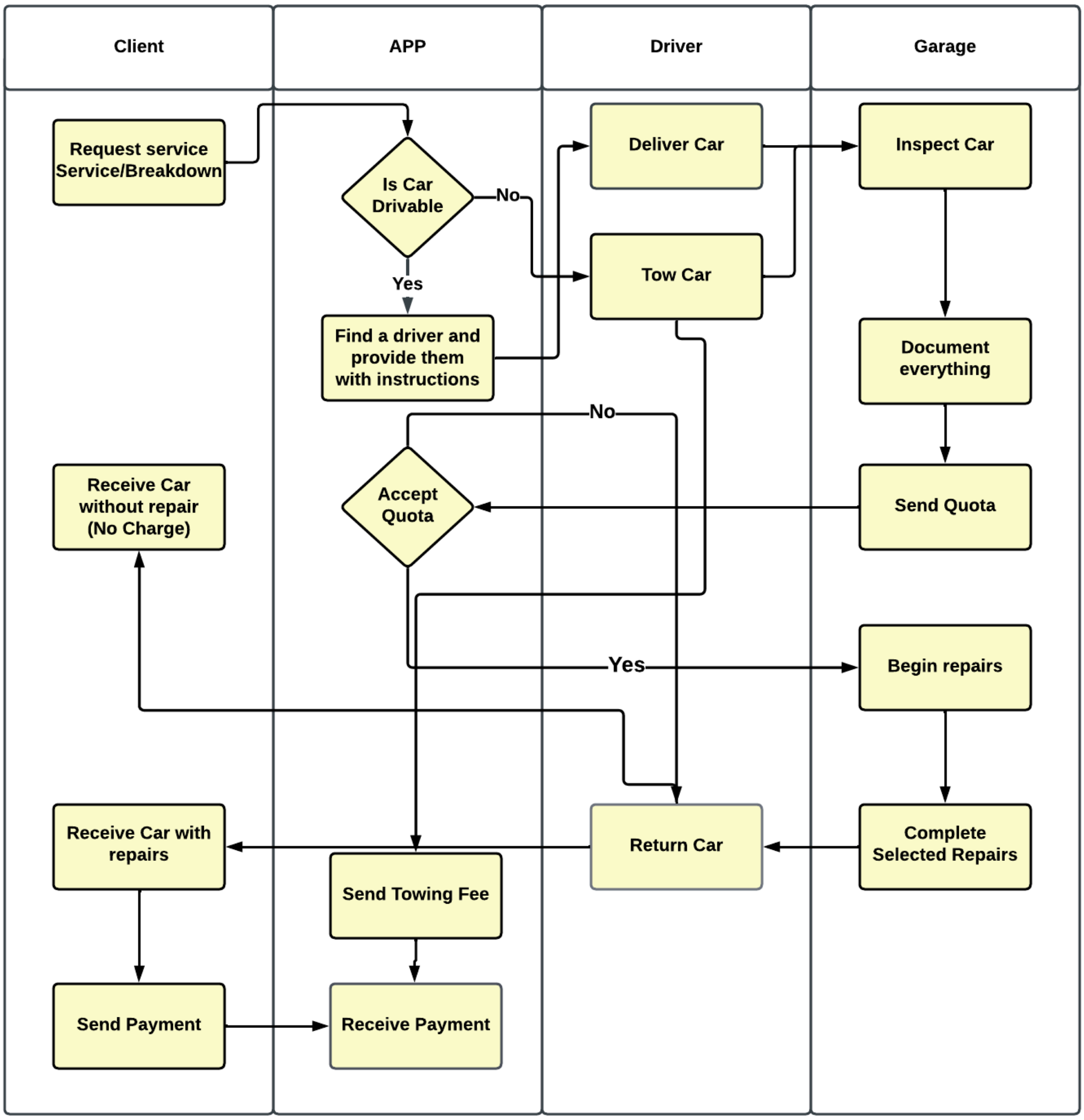
Group-1

Ahmed (Jimmy) Abdalla
Aditya Singh Gaharwar
Rhea Sera Rodrigues
Reuben Joslin Coutinho
Reshad Mohsini

Table of Contents

Modeling

Swimlane Diagram



The swimlane diagram visualizes the process flow of our cloud-based automotive repair service. The lanes are divided into four actors: Client, APP, Driver, and Garage. Here's how the process unfolds:

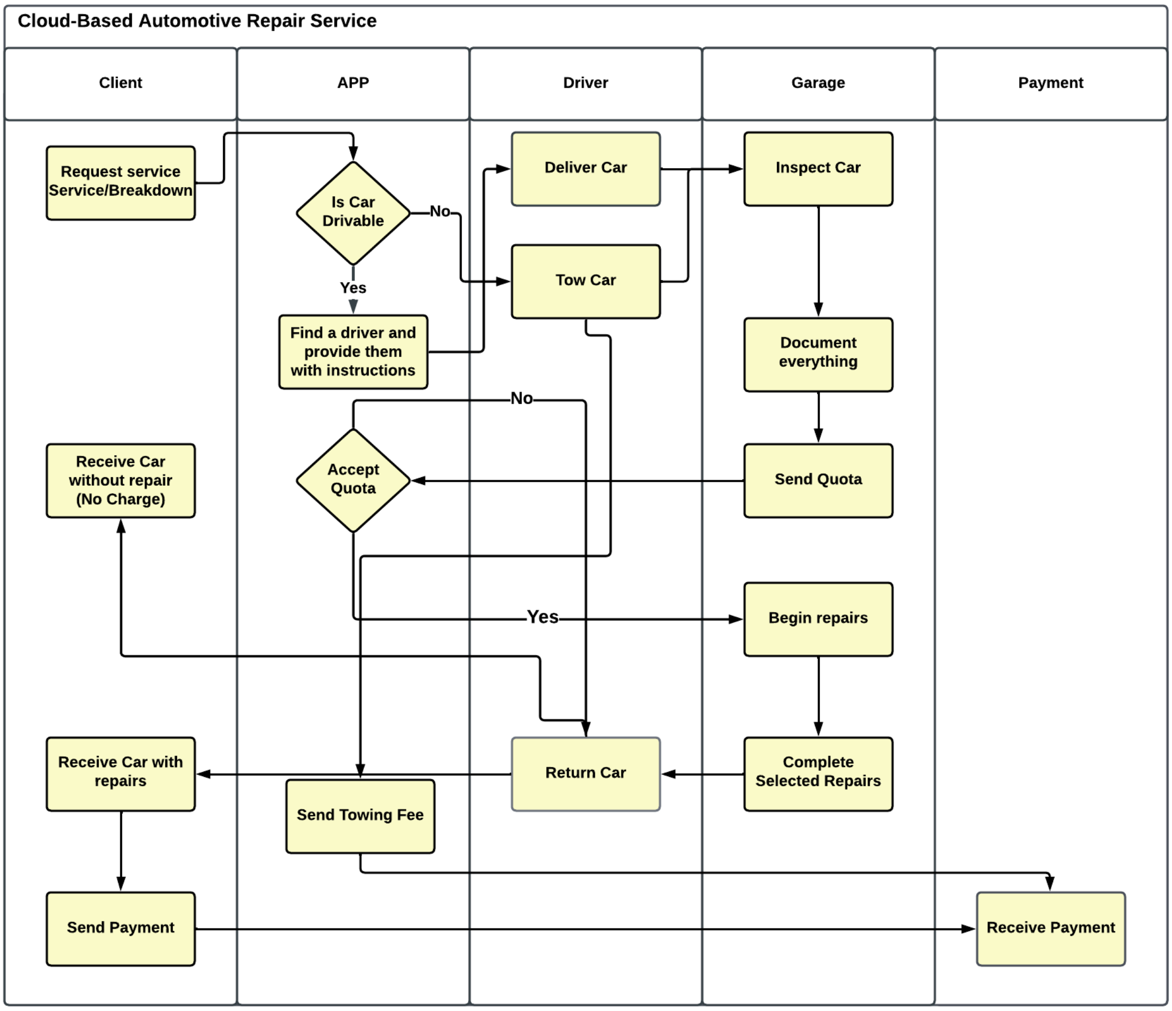
1. **Client:** The process begins with the client requesting service due to a breakdown or for routine maintenance.
2. **APP:** The application determines whether the car is drivable.

    - If yes, the app finds a driver and provides them with instructions on car pickup and garage destination.
    - If not, the car needs to be towed.
3. **Driver:**
    - If the car is drivable, the driver delivers the car to the garage.
    - If the car is not drivable, the driver tows the car to the garage.
4. **Garage:** Upon receiving the car, the garage inspects it, documents everything, and creates a quota for the necessary service or repairs. This quota is then sent back to the APP.
5. **APP:** The APP relays the quote to the client who decides whether to accept it.
    - If the quote is rejected, the car is returned to the client without repair and no charge.
    - If the quote is partially or fully accepted, the repairs begin.
6. **Garage:** Once the repairs are completed, the garage returns the car to the driver.
7. **Driver:** The driver delivers the car back to the client.
8. **Client:** Finally, the client receives the car. If repairs were made, the client sends payment to the APP, which then processes the payment.

In scenarios where the car isn't driveable and needs to be towed, there's an additional step where the client is responsible for a towing fee, highlighting that this service incurs an extra cost regardless of whether the repair quote is accepted.

This swimlane diagram effectively illustrates the step-by-step interaction between the client, the app, the driver, and the garage, providing a clear visual representation of the service process from the initial request through to the completion of repairs and payment. It shows the decision-making points, such as the assessment of the car’s condition (drivable or not), acceptance of the quota, and the flow of information and tasks among different actors.

Updated Swimlane Diagram



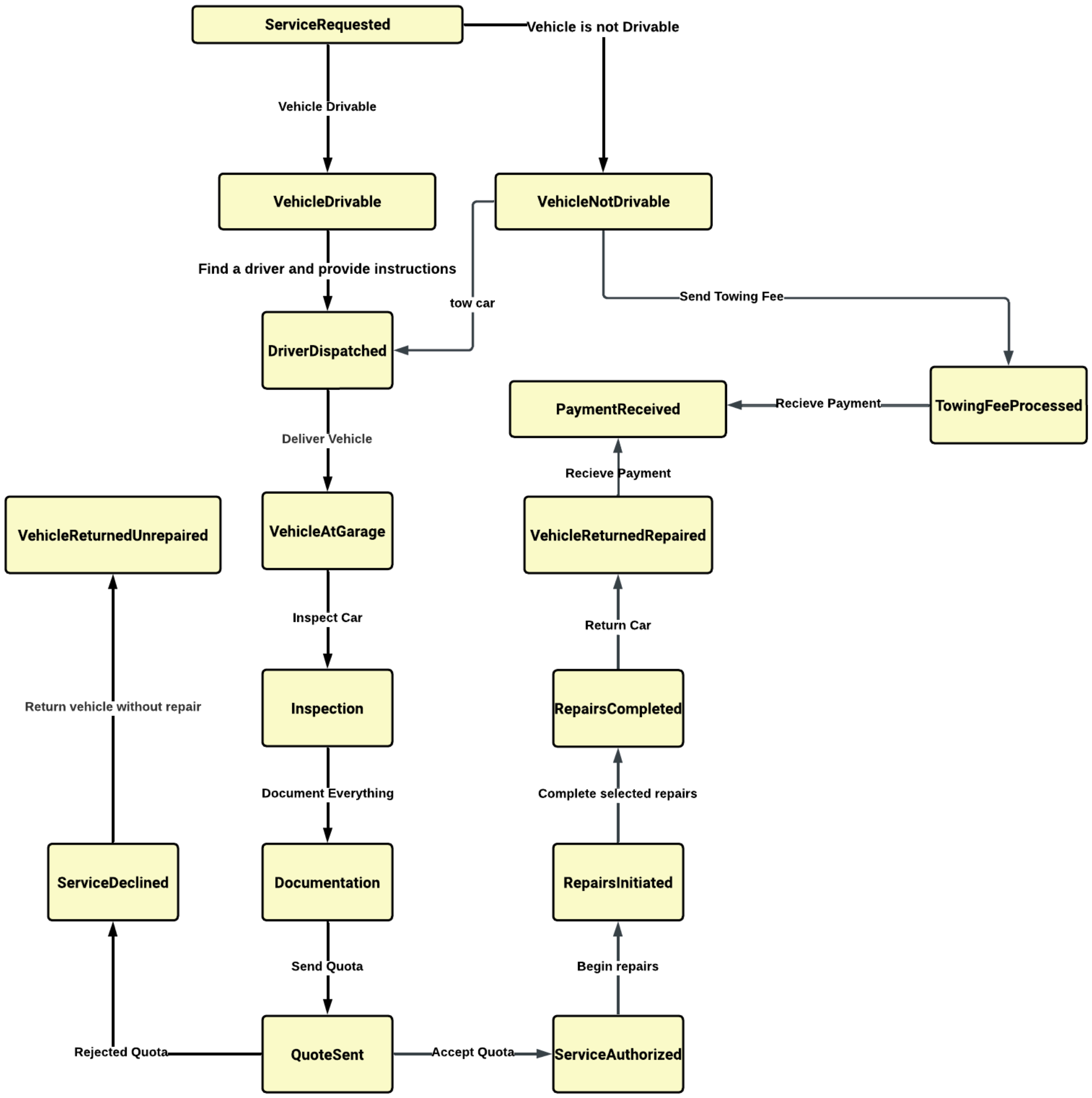
Based on stakeholder feedback, the swimlane diagram was revised to include a dedicated "Payment" lane. This outlines the financial interactions from the operational workflow, allowing for a focused examination of the monetary exchanges and improving oversight of financial controls within the system.

1. **Client**: The process begins when the client requests either service or breakdown assistance for their car.
2. **APP**: The application determines if the car is drivable.
   - If the car is drivable, the app finds a driver and provides them with instructions.
   - If the car is not drivable, it triggers the towing process.
3. **Driver**:
   - If the car is drivable, the driver delivers the car to the garage.
   - If the car is not drivable, the driver tows the car to the garage.

4. **Garage**:
    - Once the car arrives at the garage, it is inspected.
    - Everything about the condition of the car is documented.
    - The garage sends a quota to the client for the necessary services.
    - If the client accepts the quota, the garage begins repairs. If not, the process of returning the car is initiated.
5. **Client**:
    - If the client accepts the quota, they will eventually receive the car with repairs.
    - If the client declines the quota, they receive the car without repairs and no charge is applied, unless the car was towed.
6. **Payment** (Newly added lane):
    - If towing is required, the client sends a towing fee.
    - After repairs are completed and the car is returned, the client sends payment for the services.

The additional "Payment" lane in the updated diagram helps to separate and organize the financial transactions from the service delivery aspects of the process, allowing stakeholders to more easily track the flow of money separately from the operational steps. This clear division enhances the readability of the diagram and helps in identifying any financial-specific steps or requirements, such as invoicing and payment processing.

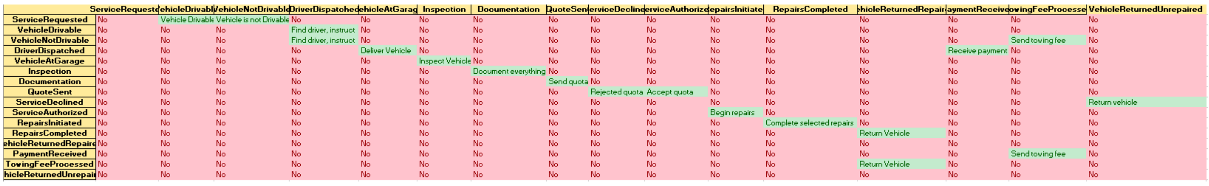
State Transition Diagram

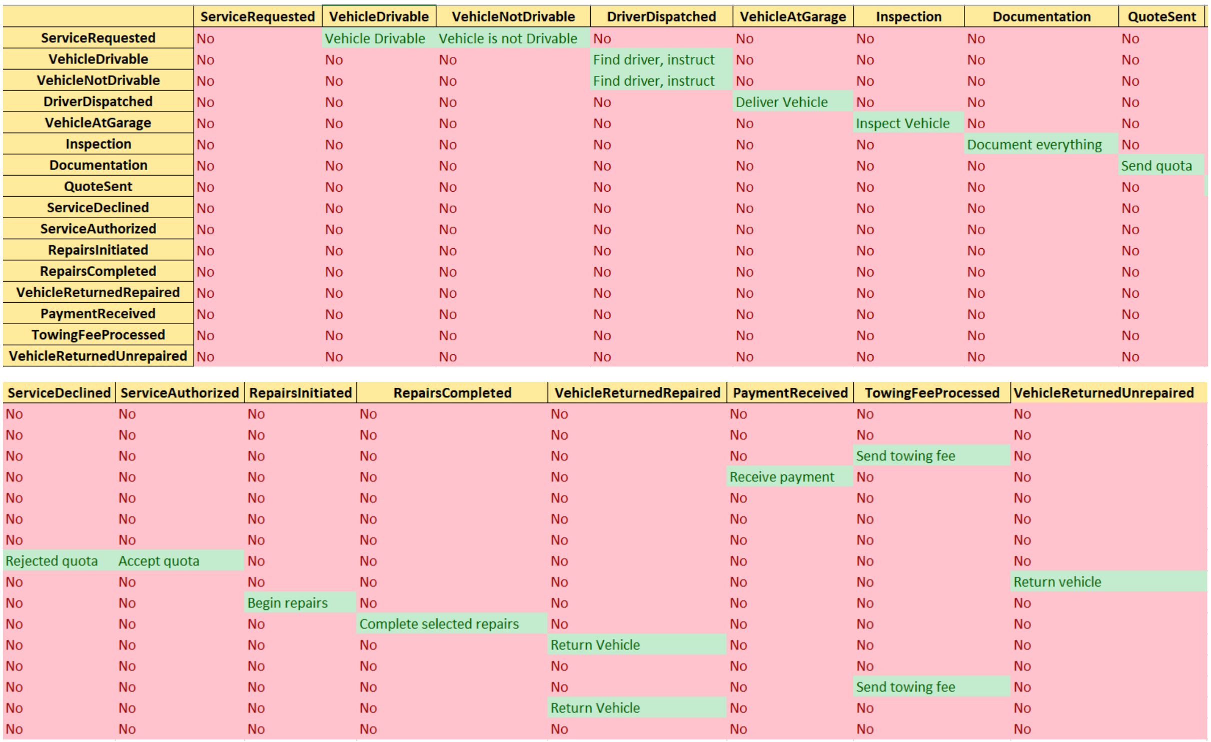


1. **ServiceRequested**: The process begins when a client requests a service or reports a vehicle breakdown through the application.
2. **VehicleDrivable** / **VehicleNotDrivable**: The application then determines if the vehicle can be driven to the garage.
   - If the vehicle is drivable, the application proceeds to dispatch a driver.
   - If not, the towing process is initiated, leading to a "TowingFeeProcessed" state before proceeding.
3. **DriverDispatched**: A driver is dispatched to the client's location to either.
   - Drive the vehicle to the garage, leading to the "VehicleAtGarage" state.
   - Tow the vehicle if it's not drivable, leading to "PaymentReceived".

4. **VehicleAtGarage**: Once at the garage, the vehicle undergoes:
   - An **inspection**, leading to documentation of its condition.
   - A quotation for repair work, moving to the "QuoteSent" state.
5. **QuoteSent**: After the client receives the quote:
   - If the service is declined, the vehicle is returned unprepared, leading to "VehicleReturnedUnrepaired".
6. **Service Authorized**: If the client authorizes the service based on the quote provided, this leads to "Repairs Initiated".
7. **RepairsInitiated**: The garage begins the repair process, leading to:
   - "RepairsCompleted": The vehicle is successfully repaired and returned to the client ("VehicleReturnedRepaired").
8. **RepairsCompleted**: After the repairs have been initiated and all necessary work has been carried out on the vehicle, the state transitions to:
   - **VehicleReturnedRepaired**: The vehicle is successfully repaired and ready to be returned to the client.
9. **PaymentReceived**: This state is reached after the towing fee has been processed (if there was a towing fee) or once the vehicle repair is completed and the client has been billed.
   - **TowingFeeProcessed**: If the vehicle was not drivable and required towing, the fee is processed and the system transitions to this state awaiting payment.

State Transition Table



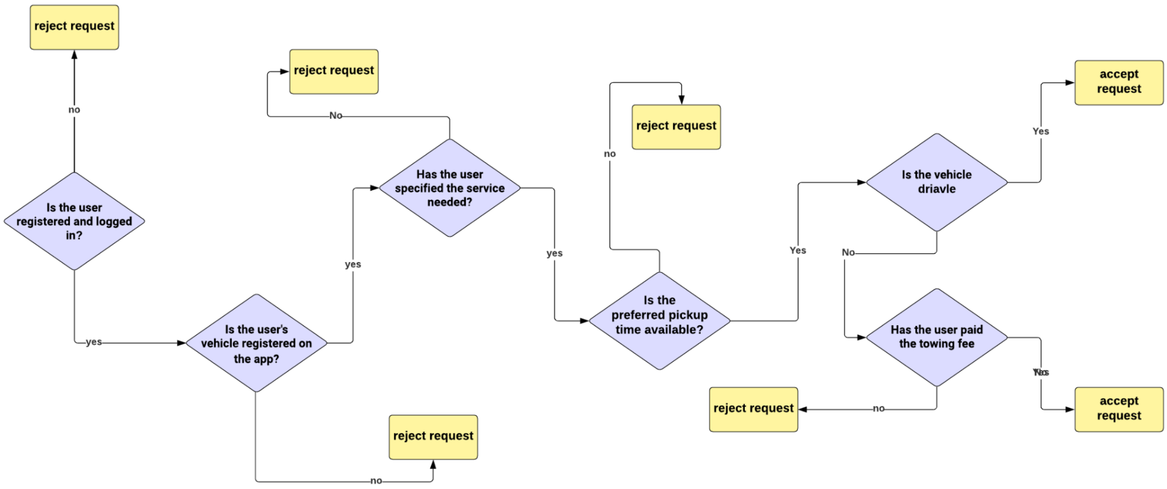


	ServiceRequested	VehicleDrivable	VehicleNotDrivable	DriverDispatched	VehicleAtGarage	Inspection	Documentation	QuoteSent	ServiceDeclined	ServiceAuthorized	RepairsInitiated	RepairsCompleted	VehicleReturnedRepaired	PaymentReceived	TowingFeeProcessed	VehicleReturnedUnrepaired
ServiceRequested	No	Vehicle Drivable	Vehicle is not Drivable	No	No	No	No	No	No	No	No	No	No	No	No	No
VehicleDrivable	No	No	No	Find driver, instruct	No	No	No	No	No	No	No	No	No	No	No	No
VehicleNotDrivable	No	No	No	Find driver, instruct	No	No	No	No	No	No	No	No	No	No	Send towing fee	No
DriverDispatched	No	No	No	No	Deliver Vehicle	No	No	No	No	No	No	No	No	Receive payment	No	No
VehicleAtGarage	No	No	No	No	No	Inspect Vehicle	No	No	No	No	No	No	No	No	No	No
Inspection	No	No	No	No	No	No	Document everything	No	No	No	No	No	No	No	No	No
Documentation	No	No	No	No	No	No	No	Send quota	No	No	No	No	No	No	No	No
QuoteSent	No	No	No	No	No	No	No	No	Rejected quota	Accept quota	No	No	No	No	No	No
ServiceDeclined	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Return vehicle
ServiceAuthorized	No	No	No	No	No	No	No	No	No	No	Begin repairs	No	No	No	No	No
RepairsInitiated	No	No	No	No	No	No	No	No	No	No	No	Complete selected repairs	No	No	No	No
RepairsCompleted	No	No	No	No	No	No	No	No	No	No	No	No	Return Vehicle	No	No	No
VehicleReturnedRepaired	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
PaymentReceived	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Send towing fee	No
TowingFeeProcessed	No	No	No	No	No	No	No	No	No	No	No	No	Return Vehicle	No	No	No
VehicleReturnedUnrepaired	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No

The state transition table here is to help visually map out and understand the state transition diagram more clearly, and help document and validate all possible states a service request can transition through and the events that trigger these transitions.

- "**No**" Entries: In the table, a "No" signifies that a transition between the states is not allowed or not possible. For instance, you cannot go directly from "ServiceRequested" to "VehicleReturnedRepaired" without the intermediate steps.
- Other **Green Entries**: Entries other than "No" define the particular event or action that triggers the state transition. For example, if the state is "QuoteSent" and the client decides to proceed with the service, indicating acceptance of the quote, this triggers a transition to the "ServiceAuthorized" state.

Decision Tree


reject request
no
Is the user registered and logged in?
yes
Is the user's vehicle registered on the app?
no
reject request
yes
Has the user specified the service needed?
No
reject request
yes
Is the preferred pickup time available?
no
reject request
Yes
Is the vehicle driavle
Yes
accept request
No
Has the user paid the towing fee
Yes
accept request
no
reject request

The decision tree for processing service requests within the cloud-based automotive repair service system consists of a series of checks that lead to an action to either accept or reject the request. The process is as follows:

1. **User Status Check**: The system first verifies if the user is registered and logged in. If this condition is not met, the request is rejected.

2. **Vehicle Registration Check**: For a logged-in user, the next check is whether the user's vehicle is registered on the app. If not registered, the request is rejected.

3. **Service Specification Check**: For a registered vehicle, the system then determines if the user has specified the required service. The absence of this specification leads to the rejection of the request.

4. **Pickup Time Availability Check**: Upon service specification, the system checks the availability of the preferred pickup time. Unavailability of the time slot results in the request being rejected.

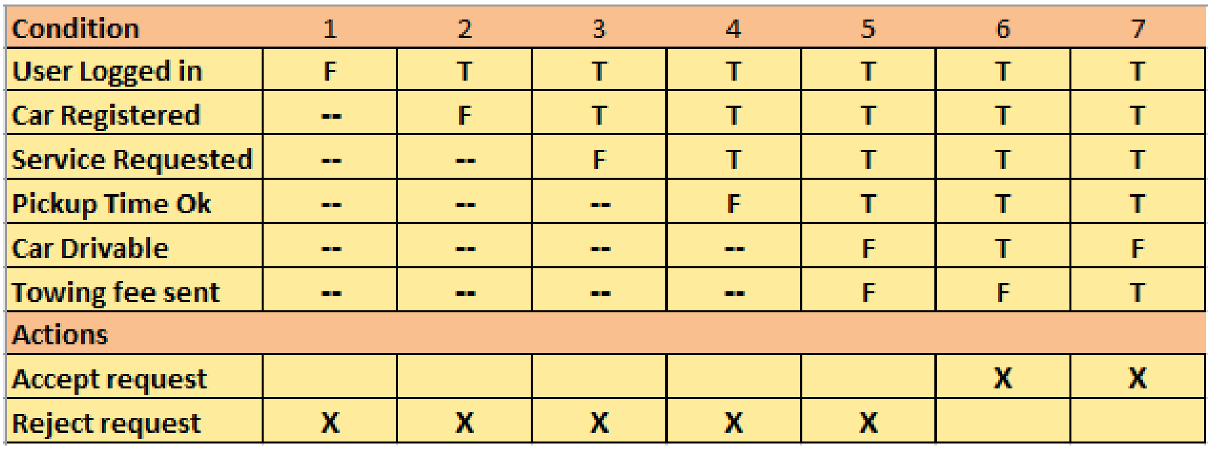
5. **Vehicle Drivability Check**: If the preferred pickup time is available, the system checks the drivability of the vehicle. If the vehicle is drivable, the system proceeds with the request.

6. **Towing Fee Payment Check**: If the vehicle is not drivable, the system requires confirmation of towing fee payment. Failure to confirm payment results in the rejection of the request.

7. **Request Acceptance**: The request is accepted only if all conditions are positively met, including the payment of the towing fee when applicable.

This decision tree ensures that all necessary conditions are reviewed before a service request is processed, optimizing the flow of operations and ensuring that system actions are consistent with user inputs and statuses.

Decision Table



Condition	1	2	3	4	5	6	7
User Logged in	F	T	T	T	T	T	T
Car Registered	--	F	T	T	T	T	T
Service Requested	--	--	F	T	T	T	T
Pickup Time Ok	--	--	--	F	T	T	T
Car Drivable	--	--	--	--	F	T	F
Towing fee sent	--	--	--	--	F	F	T
Actions							
Accept request						X	X
Reject request	X	X	X	X	X		

The decision table represents the logic for handling service requests in a cloud-based automotive repair service system, detailing the conditions that lead to either accepting or rejecting a request.

Conditions:

- **User Logged in**: The user must be logged into the system.
- **Car Registered**: The user's car must be registered within the system.
- **Service Requested**: The user must have indicated the service they require.
- **Pickup Time OK**: The user's preferred pickup time must be available.
- **Car Drivable**: It must be determined if the car is drivable.
- **Towing fee sent**: If the car is not drivable, the user must have paid the towing fee.

Actions:

- **Accept request**: The service request is approved and processed.
- **Reject request**: The service request is not approved and is terminated.

Rules:

- **Rule 1**: If the user is not logged in, the request is rejected.
- **Rule 2**: If the user is logged in but the car is not registered, the request is rejected.
- **Rule 3**: If the car is registered but no service is requested, the request is rejected.
- **Rule 4**: If a service is requested but the pickup time is not okay, the request is rejected.
- **Rule 5**: If the pickup time is okay but the car is not drivable and the towing fee has not been sent, the request is rejected.
- **Rule 6**: If the car is drivable the towing fee has been sent, and all other conditions are met, the request is accepted.
- **Rule 7**: If all conditions are true, the request is accepted.

In this table, "**T**" represents a true condition, "**F**" represents a false condition, and "–" indicates that the condition does not affect the outcome. The "**X**" under Actions indicates which action is taken based on the conditions. This table ensures that all possible scenarios are considered, reducing the risk of missing requirements and ensuring systematic processing of service requests.

References:

1. Wiegers, K., & Beatty, J. (2013). Software Requirements. Microsoft Press.

2. Ludewig, J. (2003). Models in software engineering–an introduction. *Software and Systems Modeling*, *2*, 5-14.

3. Lamsweerde, A. van. *Requirements engineering: from system goals to UML models to software specifications*. John Wiley & Sons, Ltd, 2009.