VIETNAMESE GERMAN UNIVERISTY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEER

Compulsory Elective 1

CAPTURING FULL VEHICLE HISTORY

Instructor: Dr. Le Lam Son

Student 1:	Nguyen Dai Minh	10421037
Student 2:	Nguyen Khoi Nguyen	10421097
Student 3:	Nguyen Ho Tan Dat	10421071
Student 4:	Luu Minh Khang	10421024

February 24, 2025

Content

1 Introduction **2**
1.1 Definition . . . 2
1.2 Current issues and Challenges . . . 2
1.3 Why Blockchain is necessary ? . . . 2

2 Two business processes related to capturing the full history of vehicles **3**
2.1 Overhaul Reparation of Vehicles . . . 3
2.1.1 Processes Descriptions . . . 3
2.1.2 Actors Involved . . . 3
2.1.3 Business Process Steps . . . 3
2.1.4 BPMN Diagrams . . . 5
2.2 Periodical Technical Inspection of Vehicles . . . 8
2.2.1 Case Descriptions . . . 8
2.2.2 Actors Involved . . . 8
2.2.3 Business Process Steps . . . 9
2.2.4 BPMN Diagrams . . . 10

3 Blockchain Design **14**
3.1 On-Chain Design . . . 14
3.1.1 Roles of On-Chain Component . . . 14
3.1.2 On-Chain block attributes . . . 15
3.2 Off-Chain Design . . . 15
3.2.1 Roles of Off-Chain Component . . . 15
3.2.2 Entity-Relationship Diagrams . . . 17
3.2.3 Off-Chain database table . . . 19

1 Introduction

1.1 Definition

Capturing the full history of a vehicle involves recording all relevant details throughout its lifecycle, including ownership, maintenance, accidents, inspections, repairs, and modifications. This information provides transparency and helps potential buyers, insurance companies, and regulatory authorities verify the vehicle's condition, ensuring its safety, legal compliance, and resale value.

1.2 Current issues and Challenges

There are 3 main difficulties that we currently face in capturing the whole vehicle history nowadays:

- Fragmented Data Sources: Vehicle history segmentations are often handled or captured by various systems. We can easily name several examples, such as service centers, insurance companies, regulatory agencies, and vehicle owners. These companies' and agencies' data warehouses are not always linked. These gaps lead to difficulty in summarizing data into one full record. Moreover, the pipeline between these data sources to connect together can be very inconsistent since there are plenty of differences between company regulations and configuration.
- Lack of Transparency and Trust: Vehicle history data such as accidents, repairs, or modifications of vehicles can be easily manipulated or modified, if these data do not be stored in secure storage. Moreover, this problem can also be caused during exchanges between companies by middle-man attacks to fool the receiver about the correct status of the vehicle. We can conclude that this is a critical problem about the correctness of the vehicle's true condition.
- Difficulties in Ownership Transfer: Inconsistent records make ownership transfer very time-consuming and hard, especially when we consider that vehicles have complex services or accident histories.
- Limited Access to Information: Hard accessibility for people to access and view vehicle history data is what we are facing since these data are currently not always easily accessible. Especially for potential buyers or third parties like insurers who need comprehensive records for risk assessment.

1.3 Why Blockchain is necessary ?

Blockchain technology brings to users many ideal solutions for solving challenges by giving advanced strategies

- Immutability: Vehicle data is recorded on a blockchain. This not only remains effectively unchanged data but also prevents tampering or fraudulent activities.

- Transparency and Accessibility: Blockchain creates a transparent, decentralizing ledger where vehicle history can be securely shared and accessed by all authorized parties.
- Integration of Data Sources: By linking all vehicle-related data (service records, accidents, ownership, inspections) on a single platform, blockchain provides a comprehensive and accurate history.
- Increased Trust: Blockchain technology has modern validating methods. By these methods, it will ensure that all data recorded about vehicle history is accurate and verifiable. This will directly benefit buyers, sellers, insurers, and regulators.

2 Two business processes related to capturing the full history of vehicles

2.1 Overhaul Reparation of Vehicles

2.1.1 Processes Descriptions

This process involves a detailed inspection, a comprehensive repair plan, and the use of new vehicle parts. Blockchain technology is used to record every detailed aspect of the repair, ensuring transparency, traceability, and a permanent record of all work done. This ensures that the vehicle's overhaul history is fully documented and easily accessible, helping future buyers or repair centers evaluate the vehicle's condition.

2.1.2 Actors Involved

The actors involved in this business process include:

- Vehicle Owner: Initiates the overhaul and approves the repair process.
- Repair Center: Conducts the repair and updates the blockchain with detailed service records.
- Blockchain Network: Ensures the integrity of the repair data, records the transaction, and stores the history immutably. The network is updated by the employee of the Vehicle Information Department, which might be an authorized department that has the rights to modify the car data.

2.1.3 Business Process Steps

Here are the detailed steps of this business process:

- Step 1 [Client Side]: The client initiates a request for a major overhaul service at an authorized repair center, specifying the nature of the overhaul (e.g., engine, transmission, or full vehicle restoration). This can also include specifying specific issues or goals for the repair (e.g., improving fuel efficiency or restoring a classic car).

- Step 2 [Repair Center]: The Service Center will receive and analyze the request. If the customer's request is invalid, the Service Center will announce back and end the process, if not, the process carries on with a parallel stage.

- Step 3 [Blockchain Side]: The blockchain creates a new smart contract for the overhaul service and then update to the on-chain database, which includes key details such as:

  - Vehicle information (VIN, license plate, model, year)
  - Client information (owner name, contact, and address)
  - Scope of the overhaul (what components are being repaired or replaced)
  - Estimated timeline and costs (for transparency)
  - Service center details (repair center name, technicians involved)

- Step 4 [Repair Center, Client Side]: Concurrently to the step 3, the Repair Center performs an initial assessment to evaluate the vehicle's condition and the parts needing replacement or repair. They prepare an estimate, which includes labor costs, part costs, and expected repair timeline. The client reviews and approves the overhaul plan before work begins.

- Step 5 [Blockchain Side]: The blockchain verifies and validates the overhaul agreement between the repair center and the client, ensuring all necessary conditions (warranty status, insurance, payment terms) are met. The smart contract is updated with the agreed-upon details and conditions. In case the terms are not met, the Smart Contract is terminated and the process ends. In both case, the information is then updated to the on-chain database.

- Step 6 [Repair Center]: The Repair Center begins the overhaul, which could involve disassembling parts of the vehicle, replacing critical components, and testing parts to restore the vehicle's condition to optimal functionality. The repair center updates the progress in the smart contract in real time, including details about parts used, repairs completed, and any changes in scope. The detailed logs of reparation is also updated to the off-chain database.

- Step 7 [Blockchain Side]: As the overhaul progresses, the blockchain continuously records real-time updates, ensuring full traceability. Every part installed, replaced, or repaired is logged, along with the date and specific work done. If new issues are discovered during the repair (e.g., additional damage or wear), the blockchain logs these findings and any change to the initial scope. The logs are updated to the on-chain database continuously.

- Step 8 [Blockchain Side]: The network checks if the overhaul terms are met. If all conditions are satisfied, if notifies the service center for further client inspection. On the other hand, the overhaul process carries on with the next part in the agreement.

- Step 9 [Client Side, Repair Center]: The client inspects the vehicle once the overhaul is completed and verifies that all work meets expectations. The repair center also provides

post-repair testing results and performance evaluations. The overhaul process is also finalized and updated to the off-chain database.

- Step 10 [Blockchain Side]: Once the overhaul is complete and the client approves, the blockchain updates the smart contract with final details, including the total cost, all parts used, and the final condition of the vehicle. The system broadcasts the updated record to all nodes for network-wide verification.

- Step 9 [Blockchain Side]: A new node is created in the blockchain to archive the entire overhaul process. This includes all service records, parts used, work completed, and any changes made to the original plan. This provides future owners, insurers, and service centers with a comprehensive history of the vehicle's major repairs. The information is also updated to the smart contract database.

2.1.4 BPMN Diagrams

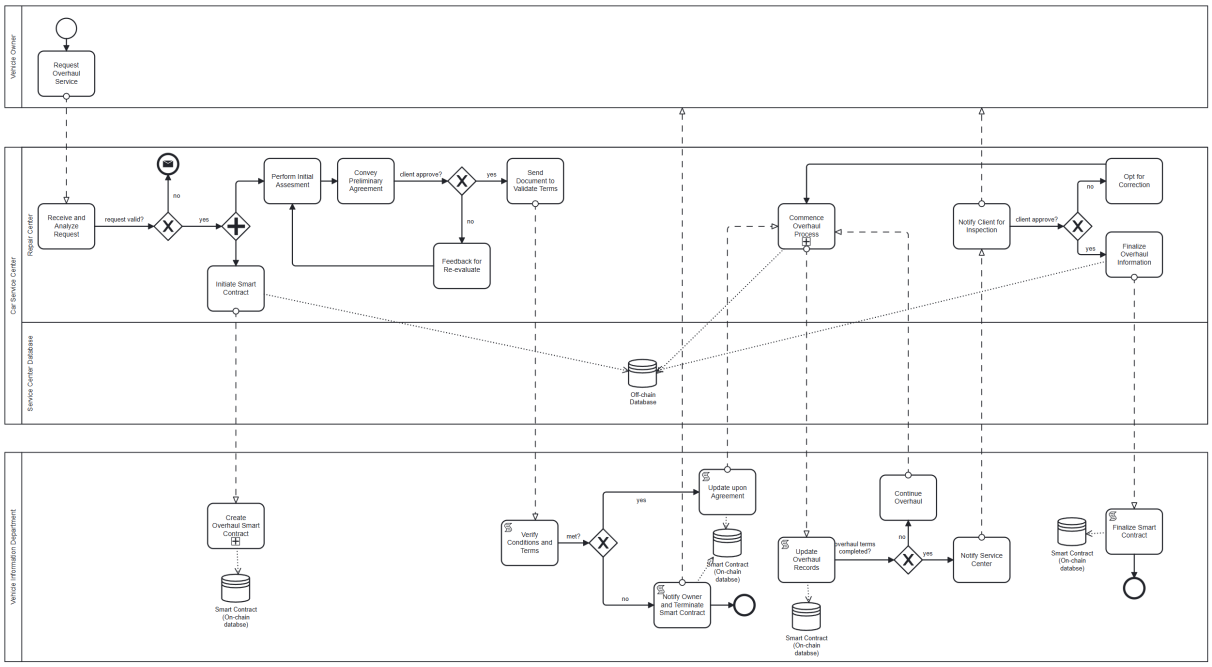

Figure 1: Overhaul Process BPMN Diagram

Including three main participants: Vehicle Owner, Car Service Center, and Vehicle Information Department. Key step in this process:

Vehicle Owner

- Request Overhaul Service: Submits a request to initiate the process.

Car Service Center

- Receive and Analyze Request: Evaluate the validity of the service request. Check the validation of the request (if Yes: Proceed to the next step/if No: End the process).

- Initiate Smart Contract: Starts formal documentation.
- Perform Initial Assessment: Conducts an initial technical evaluation.
- Convey Preliminary Agreement: Sends an agreement for approval. Check client approves (if Yes: Proceed to validate terms/ if No: Re-evaluate the terms).
- Send Document to Validate Terms: Formalizes the agreement.
- Commence Overhaul Process: Begins the overhaul process.
- Notify Client for Inspection: Invites the client for post-overhaul approval. Check client approves (if Yes: Finalize the contract/ if No: Opt for corrections).
- Finalize Smart Contract: Completes the agreement.

Vehicle Information Department

- Create Overhaul Smart Contract: Establishes the official agreement.
- Verify Conditions and Terms: Ensures contract compliance. Check condition met(if Yes: If Yes → Update records/ if No: Notify owner and terminate the contract).
- Update Overhaul Records: Maintains official records.
- Overhaul Completion Check if Overhaul terms are completed (if Yes: Notify the Service Center/if No: Continue the process).
- Notify Service Center: Confirms overhaul completion.
- Finalize Smart Contract: Marks the end of the process.

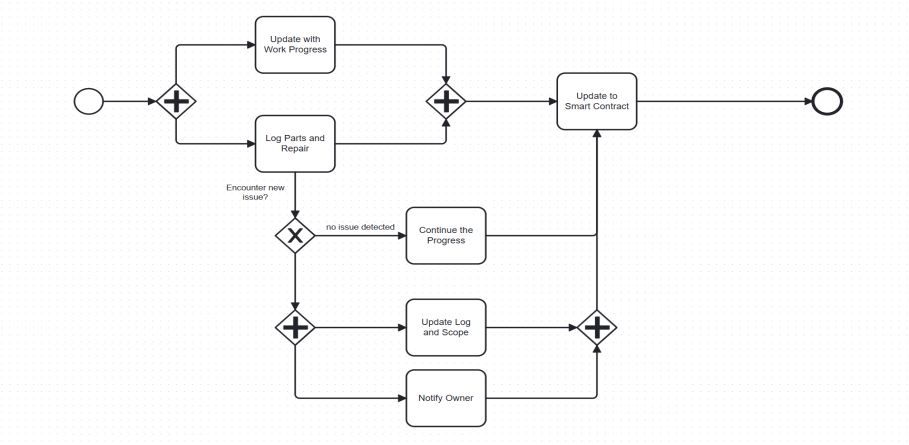

Figure 2: Commence Overhaul subprocess

This subprocess shows detailed breakdown of the Commence Overhaul Process phase. It includes action related to login progress, repairs, issue detection, and updating records. Workflows:

1. **Start Event:**
    - The process begin in initial state
2. **Parallel Tasks:**
    - Update with Work Progress: Document the current status of work completed.
    - Log Parts and Repair: Record parts used and repairs status during the overhaul.
3. **Decision: Encounter New Issue?**
    - A determine check is created to define whether any new issues are identified during the overhaul
    - if No Issue Detected: Continue the Progress, moving forward with planned repairs.
    - if New Issue Detected: Trigger corrective actions to handle the new issue –> Update records to reflect changes in scope and notify the owner about the newly encountered issue and changes in the work plan.
4. **Merge Workflow:**
    - After resolving the decision point (new issue or no issue), the line merge back into a single flow to continue the process.
5. **Update to Smart Contract:**
    - The final step that updating the smart contract to reflect all progress, repairs, and scope changes.
6. **End Event:**
    - The completion of the overhaul activities for subprocess.

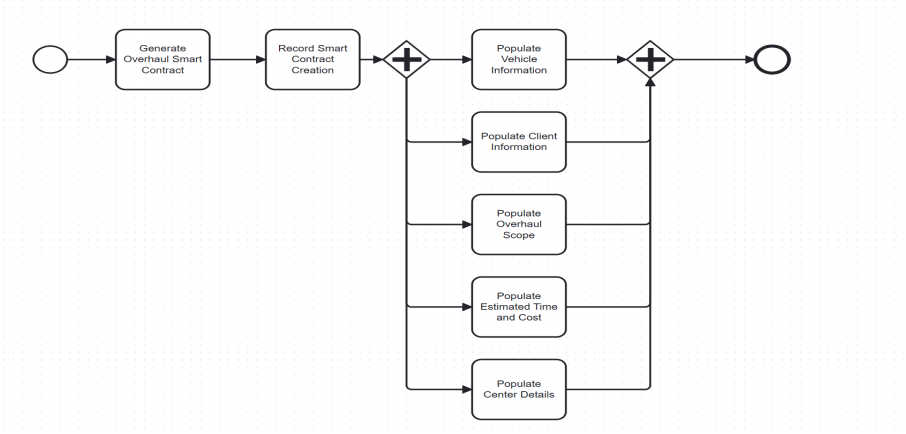

Figure 3: Create Contract subprocess

1. **Start Event:**

- The subprocess begins when the need for creating an overhaul smart contract is issue.

2. **Generate Overhaul Smart Contract:**
   - The system or employee responsible for creating the draft smart contract repair.
3. **Record Smart Contract Creation:**
   - The generated smart contract is recorded for checking and documentation purposes.
4. **Parallel Gateway:**
   - Populate Vehicle Information: Add specific details about the vehicle.
   - Populate Client Information: Include client-specific details.
   - Populate Overhaul Scope: Define the scope of work, listing repairs, replacements, and services to be performed.
   - Populate Estimated Time and Cost: Estimate the time and cost required for the overhaul process.
   - Populate Center Details: Add details about the service center handling the overhaul.
5. **Merge Gateway:**
   - After all tasks are completed, the workflow merge into a single path.
6. **End Event:**
   - The completion of the smart contract for subprocess.

2.2 Periodical Technical Inspection of Vehicles

2.2.1 Case Descriptions

Every vehicle, before it can operate on the road or after a specific period of operating, needs to be examined by an inspection center to ensure safe driving. The time is based on each government's regulations. Periodical Technical Inspection of Vehicles is the process that tracks the regular, mandatory technical inspections that vehicles undergo to ensure safety and regulatory compliance. Inspections may cover safety features, emissions, and roadworthiness. Blockchain records the results of each inspection, including passed or failed components, repairs needed, and future inspection schedules. This ensures that the vehicle's inspection history is transparent and cannot be altered.

2.2.2 Actors Involved

The actors involved in this process include:

- Vehicle Owner: Requests and takes the vehicle for inspection. They will receive the inspection results after the process.

- Inspection Center: Conducts the inspection (pre-assessment and full inspection), records the findings, and updates the status of the vehicle.
- Storage Center: Storing the non-critical data that are not tied directly to the final inspection, such as several non-critical personal data, the error logs, or technician logs. Moreover, acts as the middle-man for communication between the blockchain system and other actors.
- Blockchain Network: Validates and stores the inspection results securely, keeps data transparent, and ensures they are easily accessible for future reference.

2.2.3 Business Process Steps

Here are the detailed steps of this process:

- Step 1 [Client Side]: The vehicle owner requests a periodical technical inspection at an authorized inspection center, according to the schedule required by law or manufacturer recommendations (e.g., annually or bi-annually). The owner provides vehicle details (e.g., VIN, license plate, mileage) to initiate the inspection process. By just using their ID, they can query the data that they are storing before their data and their vehicle detail data.
- Step 2 [Blockchain Side, Storage Center]: A new smart contract has been created to track the vehicle's technical inspection process. The initial details which are retrieved from the Storage Center are recorded in the blockchain including: Vehicle information (VIN, license plate, model, year) Owner information (owner name, contact details) Inspection type (standard inspection, emissions test, safety check, etc.) Inspection due date (based on the vehicle's registration and required schedule) Inspection center details (location, inspection technician)
- Step 3 [Client side, Inspection Center]: The vehicle is brought to the inspection center, where the technician performs a pre-assessment of the vehicle's condition to identify any immediate issues. This can include checking lights, brakes, tires, and emissions. The owner may be informed if any problems are detected prior to the official inspection.
- Step 4 [Inspection Side]: The inspection center performs the required technical inspection, which may include: Emissions testing to ensure the vehicle meets environmental standards. Brake checks for safety. Tire and suspension checks for roadworthiness. Lights, horn, and safety features checks. Engine and exhaust system inspection for performance and safety. The technician records the inspection results, noting any failed components or required repairs.
- Step 5 [Blockchain Side]: The blockchain logs the detailed results of the inspection, including: Passed/failed status for each inspection category (e.g., emissions, brakes). Detailed findings (e.g., issues with brake pads, emissions failure). Parts that need replacement or repairs (if applicable). Next inspection date (based on the outcome of this inspection).

The blockchain ensures that all inspection data is stored immutably and is time-stamped for accuracy.

- Step 6 [Client Side]: The vehicle owner receives the inspection report, which includes a summary of the inspection results, required repairs (if any), and the next scheduled inspection date. The owner can also receive recommendations for any preventive maintenance.
- Step 7 [Blockchain Side]: The blockchain broadcasts the completed inspection details to all nodes in the network, ensuring the inspection history is shared and recorded across all blockchain participants. This includes a record of any failed components and repairs that were carried out, if applicable.
- Step 8 [Client Side, Inspection Center]: If the vehicle fails any part of the inspection (e.g., emissions, brakes), the owner takes the vehicle for repairs. Upon completion of the repairs, the inspection center can update the smart contract with the repaired components and re-test the vehicle if necessary.
- Step 9 [Blockchain Side]: The blockchain creates a new node to archive the vehicle's complete inspection history, including passed and failed inspections, repairs, and service recommendations. This node provides a long-term record that can be referenced during future inspections, repairs, or resale.

2.2.4 BPMN Diagrams

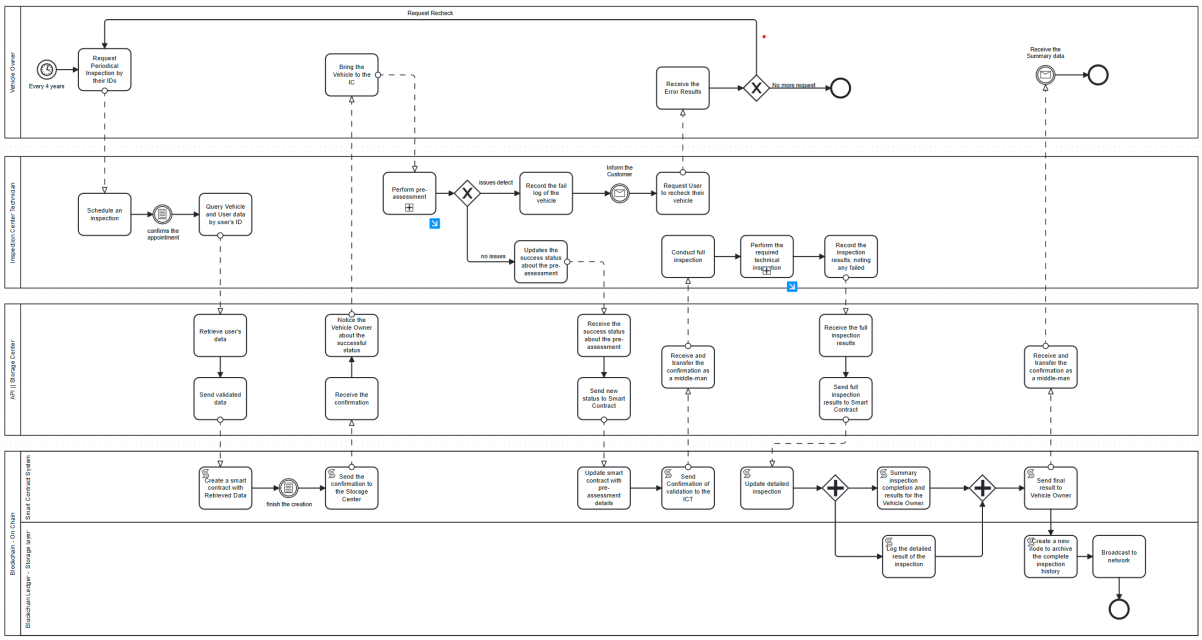

Figure 4: Periodical Inspection Process BPMN Diagram

Vehicle Owner

- Request Periodical Inspection: Submits a request to initiate the process, required every four years.

- Bring Vehicle to Inspection Center: Ensures the vehicle is available at the Inspection Center (IC) for the scheduled inspection.
- Receive Inspection Results: Obtains feedback on the inspection, including any recheck requirements.

Inspection Center/Technician

- Schedule an Inspection: Schedules and confirms the inspection with the owner.
- Query Vehicle and User Details: Retrieves relevant data for the inspection using vehicle and owner IDs.
- Perform Pre-Assessment: Perform an initial assessment of the vehicle to detect any issues. Check for Issues (if Yes: Records the fail log, informs the vehicle owner, and requests recheck/ if No: Updates the progress and proceeds with the process).
- Conduct Full Technical Inspection: Performs a comprehensive inspection if the vehicle undergoes repairs and is returned for rechecking.
- Record Inspection Results: Documents the inspection outcomes, including any repairs or updates.

API/Storage Center

- Retrieve Data: Accesses and provides necessary vehicle and user details for inspection and contract updates.
- Update Smart Contract: Records inspection details in the smart contract to ensure transparent and secure documentation.
- Send Confirmation: Notifies the Inspection Center and Vehicle Owner about the inspection status and results.

Blockchain On-Chain/Storage System

- Create Smart Contract: Generates a tamper-proof record of the inspection using blockchain.
- Log Detailed Inspection Results: Archives all inspection results and logs them securely in the blockchain.
- Broadcast to Network: Shares the complete inspection history with the network to ensure immutability and traceability.
- Finalize Process: Marks the completion of the inspection process and archives the final data.

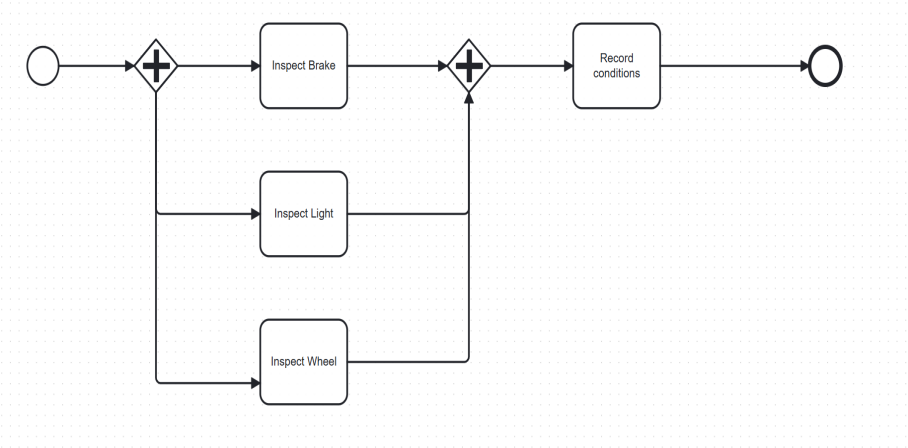

Figure 5: Perform Pre-assessment subprocess

This subprocess provides a detailed breakdown of the vehicle pre-assessment phase, focusing on key safety inspections and record-keeping. The workflows are as follows:

1. **Start Event:**
   - The process begin in initial state.
2. **Parallel Tasks:**
   - Inspect Brake: Conduct a thorough check of the braking system to ensure its functionality and safety compliance.
   - Inspect Light: Evaluate the vehicle's lighting system, including headlights, indicators, and brake lights, to confirm proper operation.
   - Inspect Wheel: Assess the condition of wheels, including tires and rims, to ensure they meet safety and operational standards.
3. **Merge Workflow:**
   - After completing all parallel inspections, the workflows merge into a single path.
4. **Record Conditions:**
   - Log the findings from the brake, light, and wheel inspections, creating a report of the vehicle's condition.
5. **End Event:**
   - The subprocess concludes with the recording of all inspected conditions, preparing for the next phase.

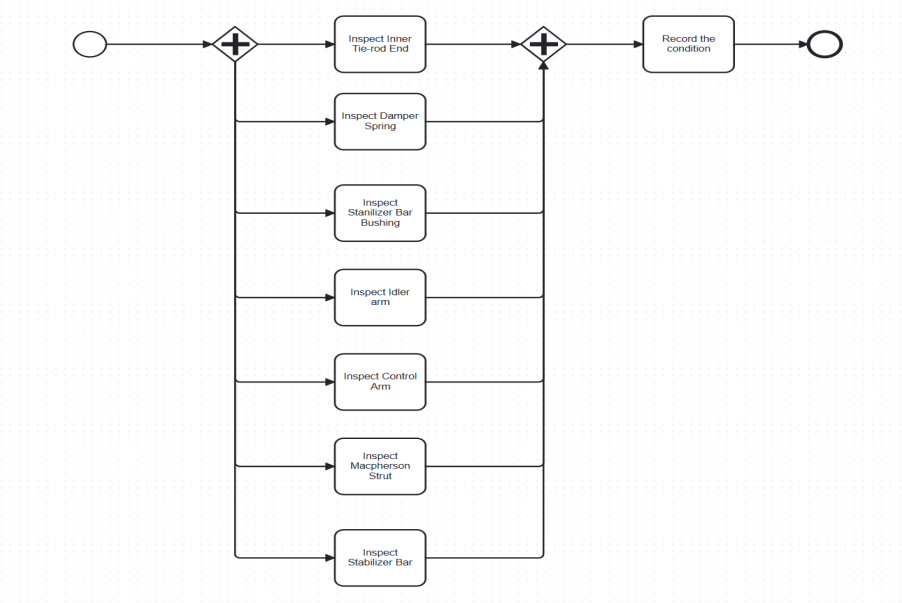

Figure 6: Perform Required Technical Assessment subprocess

This subprocess shows the detailed assessment of critical vehicle components, ensuring all are in proper condition before proceeding to the next stage. The workflows are as follows:

1. **Start Event:**
   - The process begin in initial state.
2. **Parallel Tasks:**
   - Inspect Inner Tie-rod End: Assess the inner tie-rod end for wear and tear, ensuring optimal steering performance.
   - Inspect Damper Spring: Check the condition and performance of the damper spring to ensure proper suspension functionality.
   - Inspect Stabilizer Bar Bushing: Examine the stabilizer bar bushing for damage or excessive wear, crucial for vehicle stability.
   - Inspect Idler Arm: Evaluate the idler arm for alignment and structural integrity in the steering system.
   - Inspect Control Arm: Inspect the control arm for damages or deformities, ensuring proper handling and suspension alignment.
   - Inspect MacPherson Strut: Review the MacPherson strut for any issues affecting the vehicle's suspension and ride quality.
   - Inspect Stabilizer Bar: Examine the stabilizer bar for cracks or damage, ensuring proper functionality in reducing body roll.
3. **Merge Workflow:**
   - After completing all parallel inspections, the workflows merge into a single path.

4. **Record Conditions:**
    - Document the findings for each inspected component, providing a consolidated condition report.
5. **End Event:**
    - The subprocess concludes after all conditions are recorded, ready for subsequent actions.

3 Blockchain Design

The blockchain system is implemented to ensure secure, transparent, and the recording of critical inspection processes and vehicle data. This system implements smart contracts and a distributed ledger to provide enhanced data integrity, process automation, and traceability for vehicle inspection activities.

3.1 On-Chain Design

3.1.1 Roles of On-Chain Component

The On-Chain components of the system perform the following essential roles:

1. **Data Security and Integrity:**
    - Critical inspection data, vehicle details and repair informations are stored on the blockchain, ensuring tamper-proof and immutable records
2. **Automation through Smart Contracts:**
    - Automates inspection workflows, such as validating pre-assessments, storing results, and updating inspection history.
3. **Transparency and Traceability:**
    - Provides transparent records accessible to authorized stakeholders, ensuring accountability and compliance with inspection standards.
4. **Decentralized Access:**
    - Enables authorized users to access real-time inspection records directly from the blockchain.
5. **Historical Recordkeeping:**
    - Maintains an exhaustive, immutable history of all inspections, repairs, and results for future reference and compliance checks.

3.1.2 On-Chain block attributes

The following attributes are stored securely on-chain within the Blockchain Ledger and managed by the Smart Contract System:

Attribute	Data Type	Description
OwnerID	String	Unique Identifier of the Owner.
VehicleID	String	Unique Identifier of the Vehicle.
OverhaulID	String	Unique Identifier of the Overhaul.
InspectionID	String	Unique Identifier of the Inspection.
InspectionCenterID	String	Unique Identifier of the Inspection Center.
RepairCenterID	String	Unique Identifier of the Repair Center.
TechnicianID	String	Unique Identifier of the Technician.
Status	String	Status of the Overhaul, Repair, or Inspection.
Hash	String	Unique identifier of the block (calculated hash).
TimeStamp	DateTime	Time the record was added or updated on the blockchain.
PreviousHash	String	Hash of the previous block, ensuring chain integrity.
PartID	String	Unique Identifier of the Overhaul Part (if applicable).

Table 1: On-Chain Attribute Table

3.2 Off-Chain Design

The off-chain component complements the on-chain system by managing operations that do not require blockchain immutability or distributed consensus. The off-chain system interacts with the on-chain smart contract to validate and update critical data while handling complex computations and temporary data storage.

3.2.1 Roles of Off-Chain Component

The roles of the off-chain component include the following:

1. **Data Processing:**

- Handles intensive computations, such as analytics for inspection trends or predictions for upcoming failures.
- Reduces the computational load on the blockchain network by offloading complex tasks.

2. **Temporary Data Storage:**
   - Store date such as real-time inspection logs, technician notes during inspections.
   - Ensures the data is available for immediate use but dont overload the immutable blockchain ledger.
3. **Interaction with External Systems:**
   - Store date such as real-time inspection logs, technician notes during inspections.
   - Ensures the data is available for immediate use but dont overload the immutable blockchain ledger.
4. **Interaction with External Systems:**
   - Interfaces with third-party systems such as repair shops, spare parts databases.
   - Fetches and verifies external data, such as technician certifications or vehicle manufacturer repair guidelines, before updating the blockchain.
5. **Enhanced Security:**
   - Manages secure authentication for technicians and inspection center staff.
   - Protects sensitive data, such as inspection center login credentials or proprietary repair logs, which do not require blockchain immutability.
6. **Audit Support:**
   - Logs detailed activities related to inspections and repairs for auditing purposes.
   - Prepares comprehensive reports for inspection centers and vehicle owners while maintaining blockchain integration for traceability.
7. **Smart Contract Trigger:**
   - Provides inputs to trigger on-chain smart contract events, such as inspection status updates or repair requirements.
   - Ensures data integrity and synchronization with the blockchain for critical updates.
8. **User Interface Management:**
   - Powers user-facing applications such as dashboards for vehicle owners and inspection centers.
   - Ensures smooth interaction with the blockchain system for non-technical users.

3.2.2 Entity-Relationship Diagrams

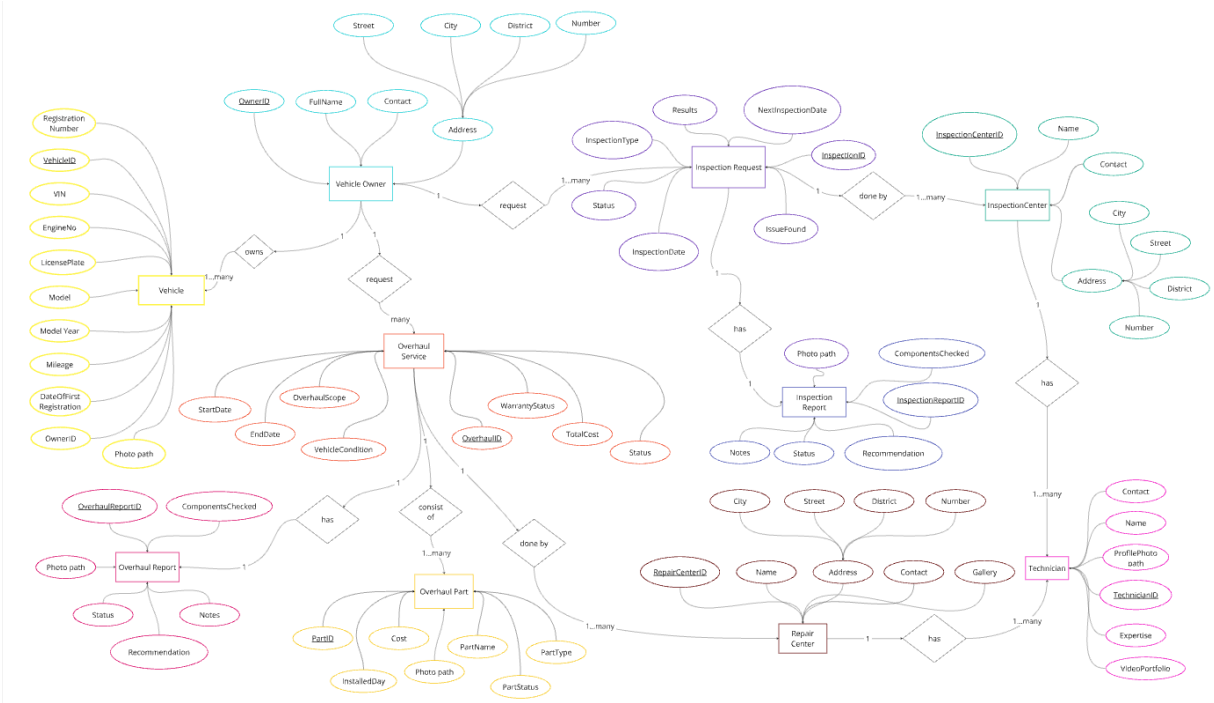

Figure 7: ER Diagrams

Relationships:

1. Vehicle and Vehicle Owner
    - Type: 1-to-many
    - Details: Each vehicle is owned by one owner, but one owner can own multiple vehicles.
    - Vehicle: Tracks ownership.
    - Vehicle Owner: Holds ownership details.
2. Vehicle Owner and Inspection Request
    - Type: 1-to-many
    - Details: A vehicle owner can request multiple inspections.
    - Vehicle Owner: Initiates inspection requests.
    - Inspection Request: Captures details of each request.
3. Inspection Request and Inspection Center
    - Type: 1-to-many.
    - Details: An inspection center performs multiple inspection requests.
    - Inspection Request: Identifies the inspection being performed.
    - Inspection Center: Executes the inspection.

4. Inspection Request and Inspection Report
    - Type: 1-to-1.
    - Details: Each inspection request generates one report.
    - Inspection Request: Tracks the request for an inspection.
    - Inspection Report: Records the findings and recommendations.
5. Vehicle Owner and Overhaul Service
    - Type: 1-to-many.
    - Details: A vehicle owner can request multiple overhaul services.
    - Vehicle Owner: Initiates overhaul requests.
    - Overhaul Service: Tracks the details of each service.
6. Overhaul Service and Overhaul Parts
    - Type: 1-to-many.
    - Details: Each overhaul service uses multiple parts.
    - Overhaul Service: Tracks the service.
    - Overhaul Part: Represents individual parts used.
7. Overhaul Service and Overhaul Report
    - Type: 1-to-1.
    - Details: Each overhaul service generates one report.
    - Overhaul Service: Summarizes the work performed.
    - Overhaul Report: Documents the service.
8. Overhaul Service and Technician
    - Type: 1-to-many.
    - Details: Overhaul services are performed by one or more technicians.
    - Overhaul Service: Tracks which technicians performed the service.
    - Technician: Executes the service.
9. Repair Center and Technician
    - Type: 1-to-many.
    - Details: A repair center employs multiple technicians.
    - Repair Center: Maintains a list of technicians.
    - Technician: Works under the repair center.

3.2.3 Off-Chain database table

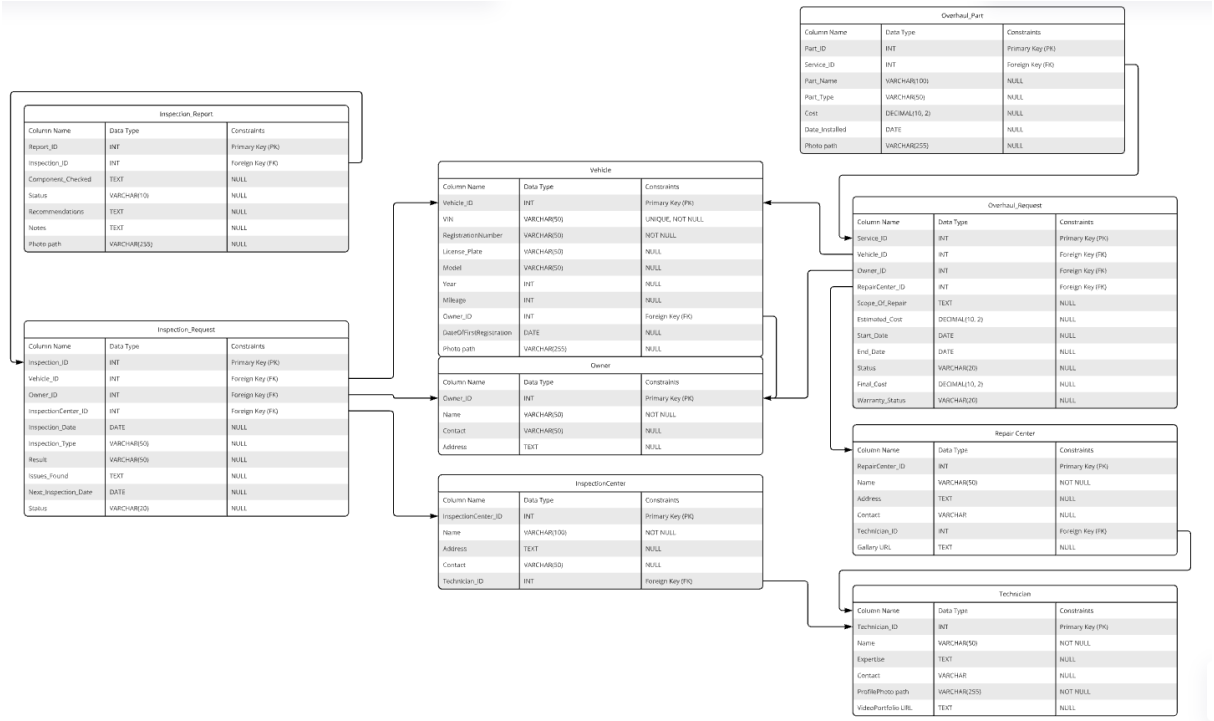

Figure 8: Off-chain database table

- **Vehicle:** Stores information about vehicles, including their details and owner relationships.
- **Owner:** Tracks details of vehicle owners.
- **Inspection_Request:** Logs inspection requests and their status.
- **Inspection_Report:** Documents the outcomes of inspections.
- **Overhaul_Request:** Manages overhaul requests, including costs and details.
- **Overhaul_Part:** Tracks parts used in the overhaul process.
- **Repair_Center:** Stores information about repair centers.
- **Technician:** Tracks technicians' profiles and skills.
- **Inspection_Center:** Logs details of inspection centers and associated technicians.