Data Types

Vehicle

Attribute	Data type	Nullable
VIN	String	Not Null
Type	String	Not Null
Manufacturer	String	Not Null
Model - Name	String	Not Null
Model - Year	String	Not Null
FuelType	String	Not Null
Color	List<String>	Not Null
Horsepower	Integer	Not Null
Description	String	Null

Customer

Attribute	Data type	Nullable
Address - Street	String	Not Null
Address - City	String	Not Null
Address - State	String	Not Null
Address - PostalCode	String	Not Null
PhoneNumber	String	Not Null
Email	String	Null

Individual

Attribute	Data type	Nullable
Name - FirstName	String	Not Null
Name - LastName	String	Not Null
SSN	String	Not Null

Business

Attribute	Data type	Nullable
TIN	String	Not Null
BusinessName	String	Not Null
PrimaryContact - Title	String	Not Null
PrimaryContact - Name - FirstName	String	Not Null
PrimaryContact - Name - LastName	String	Not Null

Sale

Attribute	Data type	Nullable
SalePrice (Derived)	Float	Not Null
SaleDate	DateTime	Not Null

Purchase

Attribute	Data type	Nullable
PurchasePrice	Float	Not Null
Condition	String	Not Null
PurchaseDate	DateTime	Not Null

Vendor

Attribute	Data type	Nullable
Name	String	Not Null
Address - Street	String	Not Null
Address - City	String	Not Null
Address - State	String	Not Null
Address - PostalCode	String	Not Null
PhoneNumber	String	Not Null

Part

Attribute	Data type	Nullable
Description	String	Not Null
VendorNumber	String	Not Null
Unit Price	Float	Not Null
Status	String	Not Null
Quantity	uint	Not Null

Part Order

Attribute	Data type	Nullable
OrderNumber	String	Not Null
TotalCost (Derived)	Float	Not Null

Employee

Attribute	Data type	Nullable
Username	String	Not Null
Password	String	Not Null

Business Logic Constraints

General Constraints

1) **Access Control**: Only privileged users such as owner, managers, inventory clerks, and salespeople can access specific functionalities. Public users can only search for vehicles.

2) **Data Integrity**: All data entered into the system must be validated for accuracy. For example - ensuring VIN uniqueness, valid dates, and valid customer identifiers.

3) **User Role Limitations**: Users can only perform actions relevant to their roles. For example - salespeople can't add vehicles; managers can only view data.

Vehicle Constraints

4) **VIN Uniqueness**: Each vehicle must have a unique Vehicle Identification Number (VIN).

5) **Model Year Validity**: The model year entered must not exceed the current year plus one and must include century digits. An example would be that "1999" is valid but "15" is not.

6) **Multiple Colors**: Vehicles can have multiple colors, and these need to be stored as a combined string or array, but displayed as a single entry in search results.

7) **Fuel Type Restrictions**: Vehicles must have a fuel type selected from a predefined list such as gas, diesel, etc.

Customer Constraints

8) **Customer Uniqueness**: Each customer must have either a unique Social Security Number (SSN) for individuals or a unique Tax Identification Number for businesses.

9) **Customer Type Handling**: The system must handle both individual and business customers, providing different input fields based on the customer type.

Seller Constraints

10) **Linking Vehicles to Sellers**: Each vehicle sold must be linked to a single seller, and each seller can sell multiple vehicles.

11) **Purchase Price Constraints**: Purchase prices for vehicles need to be manually entered by inventory clerks and they cannot be changed after the vehicle is added to inventory.

Parts Order Constraints

12) **Parts Order Status Tracking**: Each part within a parts order must track its status (ordered, received, installed), and status cannot be changed back to a previous state once updated.

13) **All Parts Must be Installed**: A vehicle cannot be sold or made searchable if it has any parts pending or not installed.

14) **Vendor Information**: Each parts order must be associated with a single vendor, and vendor details must be maintained.

Sales Constraints

15) **Sale Price Calculation**: Sale price for vehicles must be calculated as 125% of the purchase price plus 110% of any associated parts costs.

16) **Transaction Logging**: All sales transactions must log the sale date and link back to the buyer and salesperson.

Reporting Constraints

17) **Report Access**: Reports can only be accessed by the Owner and managers, ensuring sensitive information remains confidential.

18) **Highlighting Criteria**: In reports like seller history, any seller meeting specific criteria (e.g., average of five or more parts) must be highlighted visually.

19) **Drilldown Reports**: Drilldown reports must accurately reflect real-time data and adhere to sorting and filtering criteria specified in the report definitions.

Inventory Constraints

20) **Time in Inventory Calculation**: The average time a vehicle remains in inventory must consider the purchase and sale dates, counting both the first and last days as one day.

Owner Constraints

21) **Modifying sold cars:** The owner cannot sell or order parts for cars which have already been sold.

Task Decomposition and Abstract Code

Search Vehicles

Task Decomposition

Lock Types: Read-only on Vehicles and Part tables
Number of Locks:
- Vehicle Entity: 1
- Part Entity: 1
- Multivalued attribute “color”: 1
- Many-to-Many Vehicle to Part relationship: 1
- Total Read Locks: 4

Enabling Conditions: None
Frequency: 1000’s per day
Consistency (ACID): Not Critical, only reading vehicles
Subtasks: Displaying number of available vehicles and searching access the same table and occur around the same time, and therefore do not need to be decomposed.

Abstract Code

Search Vehicles
Display number of vehicles available
User enters some or all of *Vehicle type, Manufacturer, Year, Fuel type, Color,* and *Keyword* input fields
When ***Search*** button is pressed:
- If *Year* > currentYear + 1:
  - Display “Year cannot be that far in the future”
- Find all matching vehicles
- If matching vehicles is empty:
  - Display “Sorry, it looks like we don’t have that in stock!”
- Find and display all matching vehicles without uninstalled Parts.

When ***Vehicle Details*** button is pressed:
- Jump to **Display Vehicle Details** task and display **Vehicle Detail** Page

Authenticate Privileged User

Task Decomposition

Lock Types: Read only on PrivilegedUser table
Number of Locks: Single Read Lock on Employee

Enabling Conditions: None
Frequency: 100's per day
Consistency (ACID): High consistency is required to ensure only valid users are authenticated.
Subtasks: Mother task not needed. No decomposition needed.

Abstract Code

Authenticate Privileged User

User enters *username* and *password* input fields.
If username and password pass data validation checks:
    When ***Enter*** button is pressed:
        If username is not found or username is found but password is incorrect:
            Display error message "username and password combination not found"
        Else:
            Show *Search By VIN* input field
            Find user type by username
            Switch user type:
                Case Inventory Clerk:
                    Show ***Add Vehicle*** button
                    Show ***Create Part Order*** button
                Case Salesperson:
                    // Do Nothing
                Case Manager:
                    Show ***View Inventory*** button
                    Show ***Purchase History*** button
                    Show ***View Reports*** button
                Case Owner:
                    Show ***Add Vehicle*** button
                    Show ***Create Part Order*** button
                    Show ***View Inventory*** button
                    Show ***Purchase History*** button
                    Show ***View Reports*** button

Search Vehicles Privileged

Task Decomposition

Lock Types: Read-only on Vehicles table.
Number of Locks: Single
Enabling Conditions: Triggered by successful login.
Frequency: 1000's per day.
Consistency (ACID): High consistency to ensure that data is up to date and to prevent any errors in search results.

Subtasks: Displaying number of available vehicles and searching access the same table and occur around the same time, and therefore do not need to be decomposed.

Abstract Code

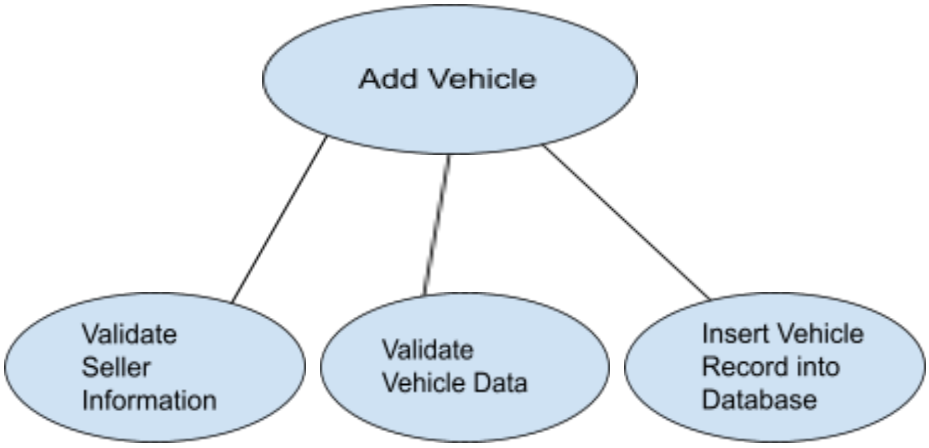
Search Vehicles Privileged
Display number of vehicles available
If Manager or IC user type:
    Display number of vehicles with Parts with status other than “installed”
User enters some or all of *Vehicle type, Manufacturer, Year, Fuel type, Color, Keyword*, and *VIN* input fields
When ***Search*** button is pressed:
    If *Year* > currentYear + 1:
        Display “Year cannot be that far in the future”

    Find all matching vehicles
    If matching vehicles is empty:
        Display “Sorry, it looks like we don’t have that in stock!”
    If user type is Manager or Owner:
        If filter input is “Sold Vehicles”:
            Find and display matching vehicles that are involved in a sale transaction
        If filter input is “Unsold Vehicles”:
            Find and display matching vehicles not involved in a sale transaction
        If filter input is “All Vehicles”:
            Find and display all matching vehicles
    Else:
        Find and display all matching vehicles without uninstalled Parts.

When ***Add Vehicle*** button is pressed:
    Jump to **Add Vehicle** task and display **Add Vehicle** Form
When ***Vehicle Details*** button is pressed:
    Jump to **Display Vehicle Details** task and display **Vehicle Detail** Page
When ***View Reports*** button is pressed:
    Show dropdown selection with buttons
- ***Seller History*** -> Jumps to **Generate Seller History** and **Seller History Report Page**
- ***Average Time in Inventory*** -> Jumps to **Calculate Inventory Time** and **Average Time in Inventory Report** Page
- ***Price Per Condition*** -> Jumps to **Calculate Price per Condition** and **Price Per Condition Report** Page
- ***Parts Statistics*** -> Jumps to **Generate Parts Statistics** and **Parts Statistics Report** Page
- ***Monthly Sales*** -> Jumps to **Generate Monthly Sales Summary** and **Monthly Sales Summary Report** Page

Add Vehicle

Task Decomposition



Lock Types: Write lock on Inventory table.
Number of Locks: Single
Enabling Conditions: Inventory clerk must be logged in.
Frequency: Around 10-20 additions per day.
Consistency (ACID): High consistency is required for data integrity.
Subtasks: Validate seller information, Validate Vehicle Data, Insert Vehicle Record into Database.

Abstract Code

Add Vehicle
Inventory clerk selects ***Add vehicl***e.
The system displays form to input vehicle details: *VIN, vehicle type, condition, purchase price, seller detail, etc.*
Validate all fields for required data and correct formats.
    If seller does not exist in database:
        Show an option to add a new seller and validate their details
On form submission:
Insert the new vehicle in the database.
Redirect the clerk to the vehicle details page, showing entered data.
If the addition is unsuccessful due to validation or database errors:
    Return to the form with appropriate error messages.

Search/Add Customer

Task Decomposition

Lock Types: Read-only on the customer table if searching for an existing customer.
Write on the customer table if adding a new customer.
Number of Locks: Single read and single write.
Enabling Conditions: The task can be performed only during a sale or purchase transaction.
Frequency: Around 50-100 customer searches or additions per day.
Consistency (ACID): Critical - customer records should be unique and accurate.
Subtasks:

1) Input SSN or Tax ID to search for a customer
2) If customer is found, then use existing record
3) If customer is not found, add new customer details
4) Associate customer with current transaction

Abstract Code

Search/Add Customer
User selects the search/add customer option.

Display form with input fields:

- *SSN* or *Tax ID*

If SSN is entered:
Validate *SSN* or *Tax ID*:
- If invalid format, display error message and stop

- If valid, proceed to search

Search database for customer using *SSN* or *Tax ID*
If customer is found:
Display customer details.
Allow the user to proceed with the next action.

Else if customer is not found:
Display “Customer not found. Add a new customer” option.

If user selects ***Add new customer***:
Display form for new customer details:
- *First name*
- *Last name* (or business name)
- *Address*
- *Phone Number*
- *SSN or Tax ID* (already entered)

Validate new customer data:
- Ensure all required fields are filled.
- Check *SSN* or *Tax ID* for uniqueness in the database.

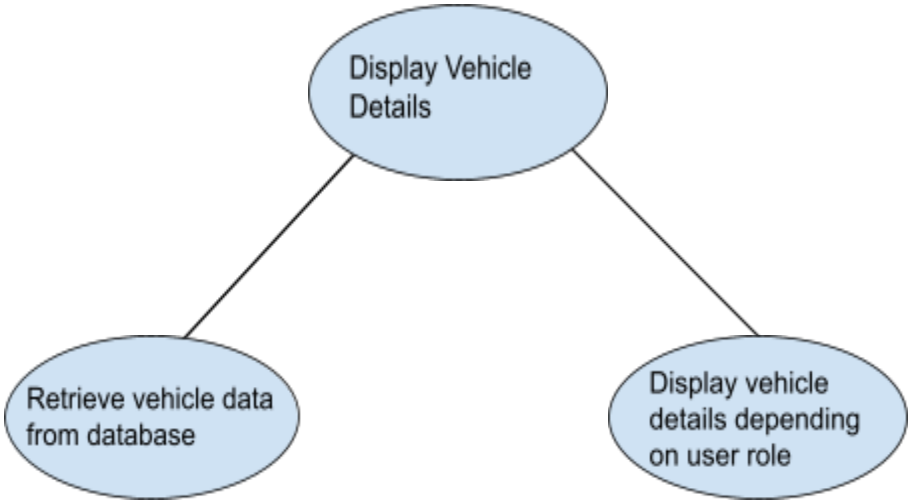
If validation fails:
Display error message and return to the form.

If validation succeeds:
Insert new customers into the database.
Display confirmation message: “Customer successfully added.”

If database error occurs during search or addition:
Display error message and return to the form for correction.

Display Vehicle Details

Task Decomposition



Lock Types: Read lock on the Vehicle and Parts tables.
Number of Locks: Single
Enabling Conditions: Users must be logged in.
Frequency: Regular use
Consistency (ACID): High consistency is required to ensure the details displayed are accurate and up to date.
Subtasks:

1) Retrieve vehicle data from the database.
2) Display vehicle details in the appropriate format depending on the user role.

Abstract Code

Display Vehicle Details
User selects “View Vehicle Details” from search results.

Retrieve vehicle ID from user input (e.g., search results or *VIN* input).

Query database for vehicle details using vehicle ID:

- VIN
- Vehicle type
- Condition
- Purchase price
- Parts status (pending/received/installed)
- Seller details (if it is relevant to user role)
- Purchase date

If vehicle is found:

Display vehicle details based on user role:

If user is Inventory Clerk:

- Display additional fields: Original purchase price, parts cost.
- Display parts list: Part number, description, vendor, part status, and ***Update Parts Status*** button.
- Enable buttons/links for ***Create Parts Order*** and ***Record Vehicle Sale***.

If user is a salesperson:

- Display vehicle details without parts data.
- Enable ***Sell Vehicle*** option if the vehicle has no pending parts.

If user is a manager:

- Display full vehicle details including purchase price, parts, seller info, sales history.
- Allow filtering by sold/unsold vehicles.

If user is an owner:

- Display all available data, including ability to edit or update vehicle details (within defined constraints)
- Ensure no ability to modify vehicles that have been sold.

If vehicle is not found:

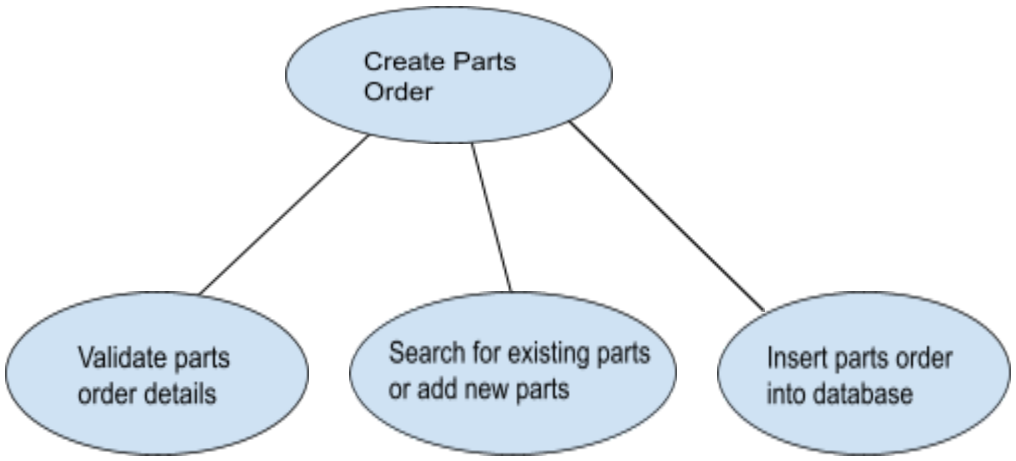
Display error message: “Vehicle not found.”

If database error occurs:

Display error message and return to previous page.

Create Parts Order

Task Decomposition



Lock Types: Write lock on Parts Order and Parts table.
Number of Locks: Two
Enabling Conditions: Inventory clerk must be logged in.
Frequency: Around 5-10 orders created per day.
Consistency (ACID): High consistency is required to maintain accurate records of parts orders.
Subtasks:

1) Validate parts order details.
2) Search for existing parts or add new parts
3) Insert parts order record into the database

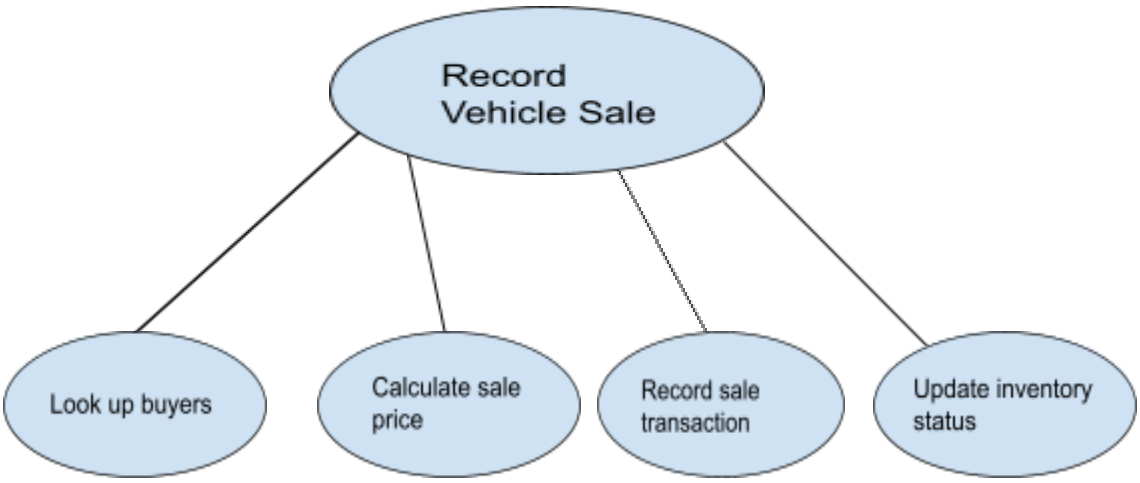
Abstract Code

Create Parts Order

If User enters *Vendor Name* input field and presses ***Search for Vendor*** button
    Jump to **Search for Vendor** task
If User presses ***Add Vendor*** button:
    Jump to **Add Vendor** task
Until User presses ***Add Parts Order*** button:
    User enters *part description, vendor part number,* and *unit cost* input fields in **Part** form
    If User presses ***Add Part*** button:
        Add part to order and clear **Part** form
Part order status is set to “ordered”

Record Vehicle Sale

Task Decomposition



Lock Types: Read-only on the Inventory table. Read-only on the Customer table. Write on the Sales Transaction table.
Number of Locks: Two (one for reading inventory/customer, one for writing sales transactions).
Enabling Conditions: Vehicle must have no pending parts and must be unsold.
Frequency: Estimated 50-100 sales transactions per week.
Consistency (ACID): Critical. This operation must ensure that sales are processed accurately, so consistency and atomicity are required. Either the vehicle is fully sold, or no transaction is recorded.
Subtasks:

1) Look up buyers (check if they are an existing customer, otherwise add customers).
2) Calculate sale price.
3) Record sale transaction.
4) Update inventory status (mark as sold).

Abstract Code

Record Vehicle Sale
Salesperson selects ***Record Vehicle Sale***

Display form to input sale details:

- *VIN of the vehicle being sold*

- *Customer ID (SSN or Tax ID)*
- *Sale price*
- *Sale date*

Validate all fields for required data and correct formats.

If customer is found:
- Confirm customer details.

Else:
- Show option to add a new customer.
    - Input new customer details (*First name, Last name, SSN or Tax ID*).
    - Validate customer details for uniqueness.

If vehicle is found and available for sale:
- Check if the sale price matches the vehicle's listed price.

If all validations are successful:

On form submission:
- Create new sale in database:
    - Store VIN, customer details, sale price, and sale date.
- Update vehicle status in inventory that it has been sold

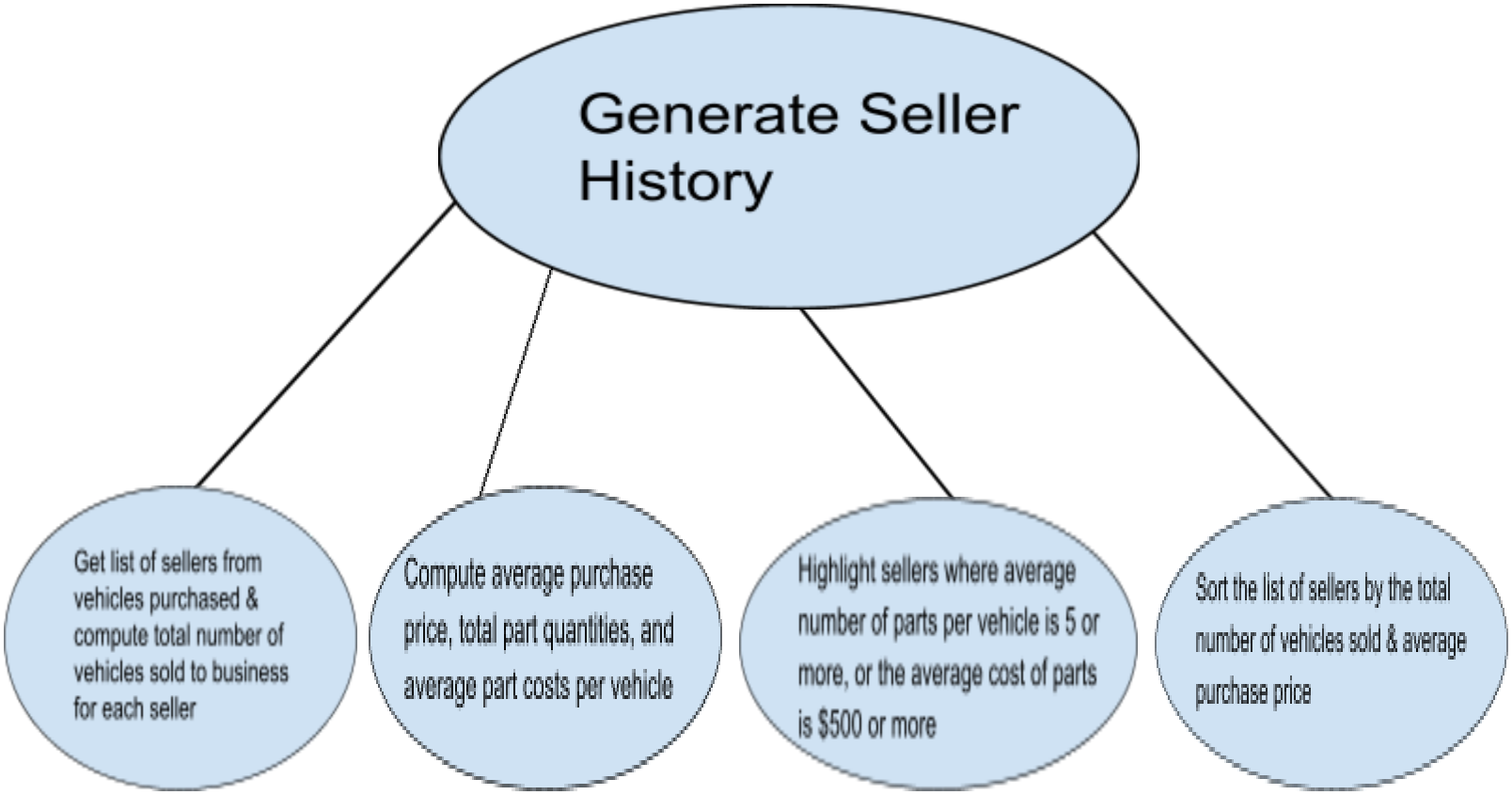
Redirect salesperson to sales confirmation page

If the sale recording is unsuccessful due to validation or database errors:

Return to the form with appropriate error messages:
- Specify which fields are invalid.
- If a new customer was added, show the confirmation.

Generate Seller History

Task Decomposition



Lock Types: Read only on Vehicle purchases, Parts Order, Parts, and Customer
Number of Locks: Four
Enabling Conditions: Manager or owner must be logged in.
Frequency: Monthly or quarterly report.
Consistency (ACID): High consistency is required for accuracy of seller details and vehicle purchase/parts data.
Subtasks:

1) Fetch the list of all sellers from whom vehicles were purchased and compute the total number of vehicles sold to the business for each seller.
2) For each seller, compute the average purchase price of vehicles, total part quantities, and average part cost per vehicle.

3) Highlight sellers where the average number of parts per vehicle is 5 or more, or the average cost of parts is $500 or more.
4) Sort the list of sellers by the total number of vehicles sold (descending) and average purchase price (ascending).

Abstract Code

Generate Seller History
Get list of all sellers
For each seller:
    If seller type is Individual:
        Get seller first and last name
    Else:
        Get business name

    Get list of vehicles sold by seller
    Calculate average vehicle purchase price
    Calculate total parts per vehicle quantity
    Calculate total parts per vehicle cost

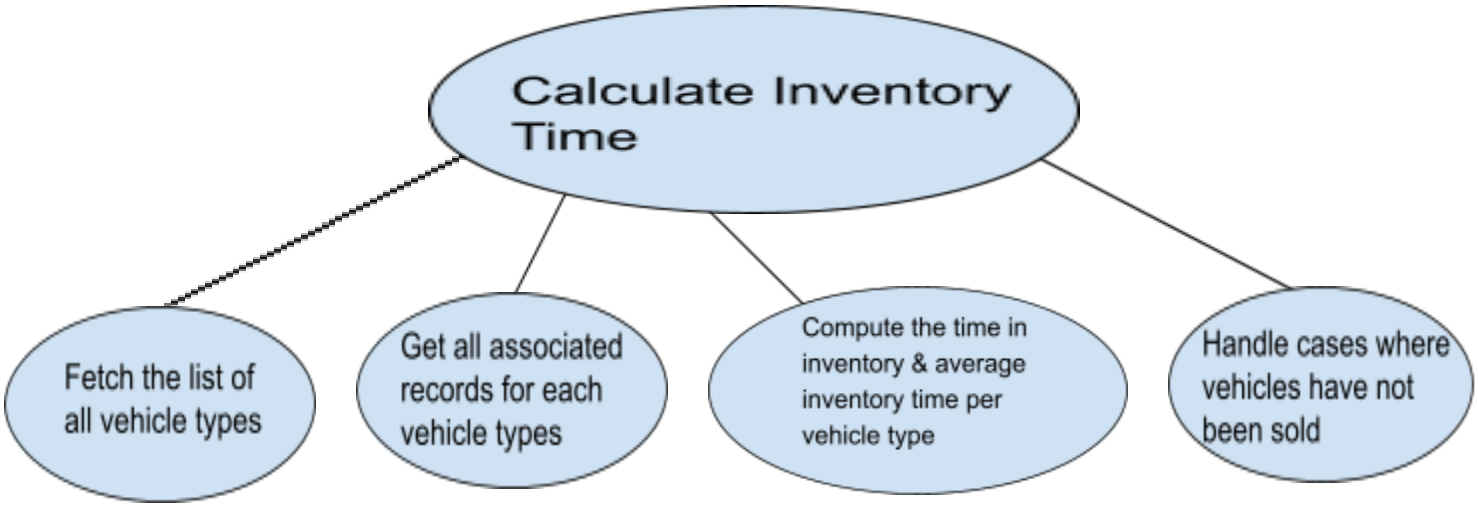
    If average total parts per vehicle quantity > 5 or total parts per vehicle cost > $500:
        Flag seller to be highlighted

Sort list of sellers by total number of vehicles sold descending, break ties with average purchase price ascending

Display sorted list of sellers, highlighting any flagged sellers with a red background

Calculate Inventory Time

Task Decomposition



Lock Types: Read only on the Vehicle and Transaction.
Number of Locks: Two
Enabling Conditions: Manager or owner must be logged in.
Frequency: Monthly or quarterly report.
Consistency (ACID): High consistency is required for the data.
Subtasks:

1) Fetch the list of all vehicle types.
2) For each vehicle type, get all associated vehicle records.
3) Compute the time in inventory and the average inventory time per vehicle type.
4) Handle cases where vehicles have not been sold.

Abstract Code

Calculate Inventory Time

```
For each vehicle type:
        Get list of all vehicles of vehicle type
        SoldFlag = False
        For each vehicle:
                If vehicle has sold date:
                        Sold Flag = True
                        If sold date is same as purchase date:
                                Inventory time = 1
                        Else:
                                Inventory time = sold date - purchase date
                        Total Inventory time += Inventory time
```

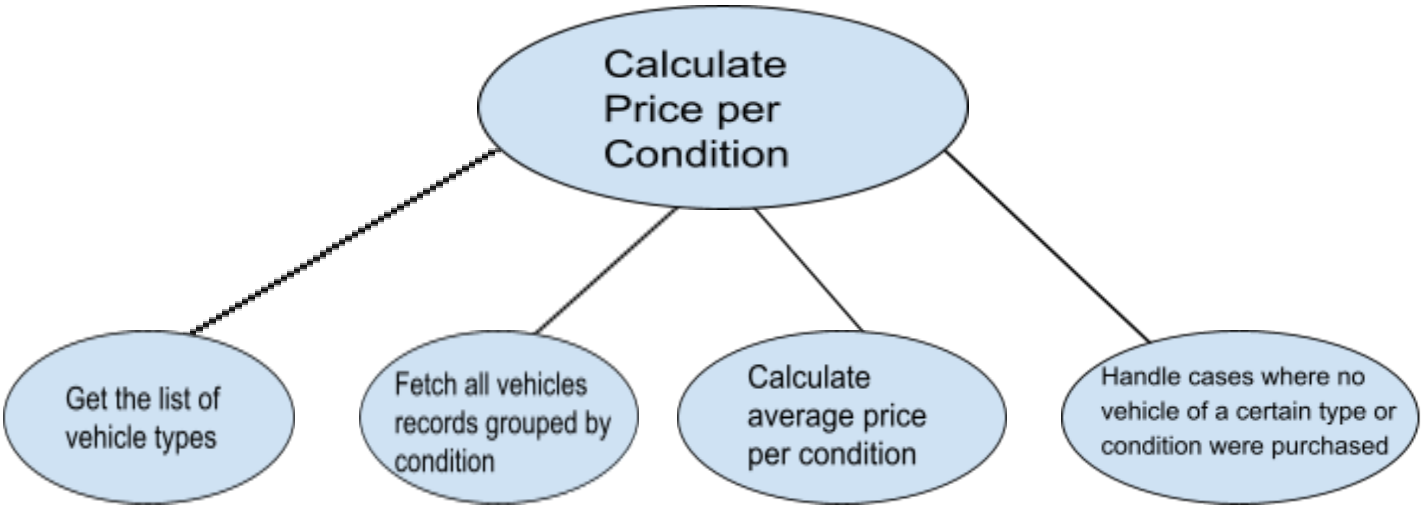
Sold Count += 1

If SoldFlag:
    Average Inventory time = Total inventory time / Sold count
Else:
    Average Inventory time = “N/A”

Display each vehicle type with its Average Inventory Time

Calculate Price per Condition

Task Decomposition



Lock Types: Read only on the Vehicle table.

Number of Locks: Single
Enabling Conditions: Manager or owner must be logged in.
Frequency: Occasionally for calculations on reports and price assessments.
Consistency (ACID): Atomicity is not as critical since it is read only.
Subtasks:

1) Get the list of vehicle types
2) For each vehicle type, fetch all vehicle records grouped by condition
3) Calculate the average price for each condition
4) Handle the cases where no vehicle of a certain type or condition were purchased

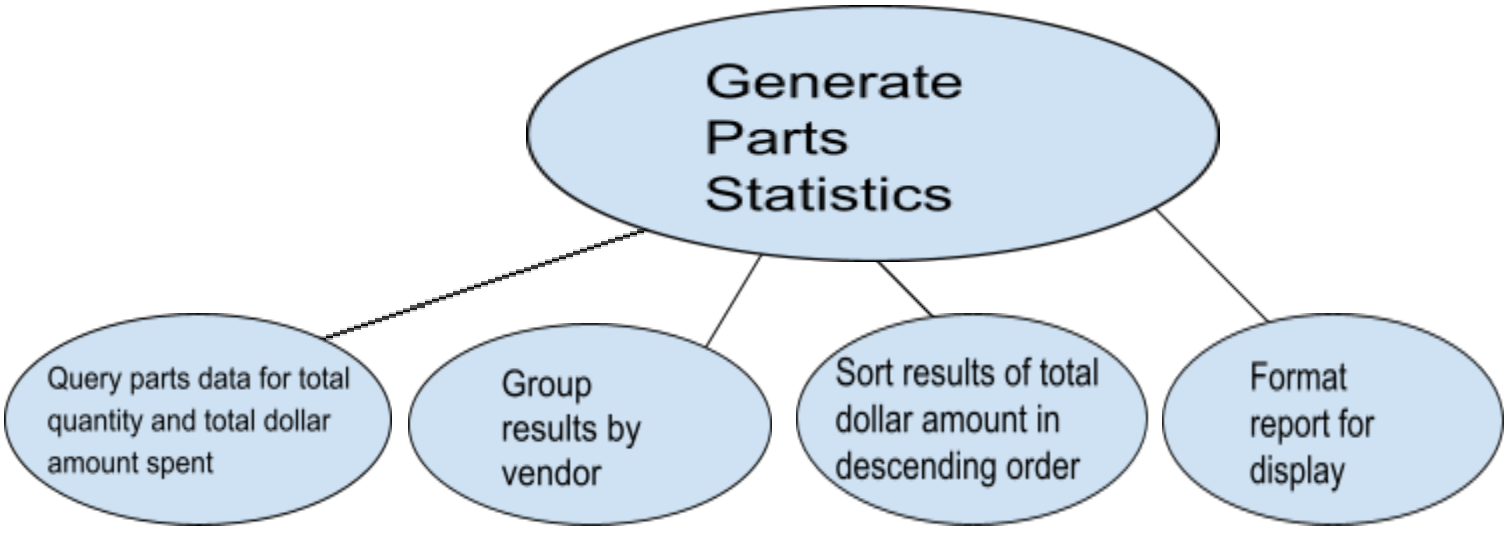
Abstract Code

Calculate Price per Condition

```
For each vehicle type:
        Get list of all vehicles of vehicle type
        For each condition:
                If no vehicles of that type and condition purchased:
                        Average purchase price = 0
                Else:
                        Get average purchase price of vehicles
Display Purchase Prices in pivoted report showing Vehicle type vs. Condition
```

Generate Parts Statistics

Task Decomposition



Lock Types: Read only for the parts table, Read only for the vendors table.
Number of Locks: Two
Enabling Conditions: None
Frequency: 20-30 reports generated per week - estimated
Consistency (ACID): Moderate. The data should show the current state.
Subtasks:

1) Query parts data to get total quantity and total dollar amount spent from each vendor.
2) Group the results by vendor.
3) Sort the results by total dollar amount spent in descending order.
4) Format the report for display.

Abstract Code

Generate Parts Statistics
User selects ***Generate Parts Statistics***

Query the database

- Retrieve all records from the parts table.
- Join with the vendor table to get vendor details.

Group results by vendor:

- Calculate total quantity of all parts supplied by each vendor.
- Calculate total dollar amount spent on parts from each vendor.

Sort results by total dollar amount spent in descending order.

If results are found:

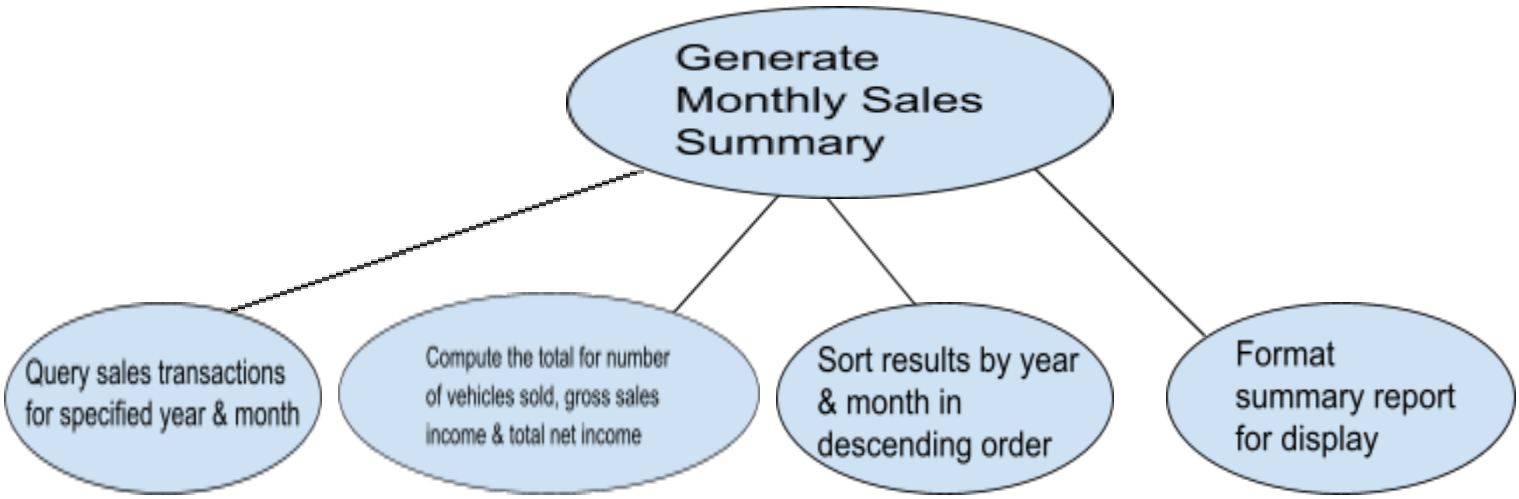
- Format and display the statistics report.

Else:

- Display a message indicating that no parts statistics are available.

Generate Monthly Sales Summary

Task Decomposition



Lock Types: Read only on the sales transaction table.
Number of Locks: Single
Enabling Conditions: None
Frequency: 15-25 reports generated per month - estimated
Consistency (ACID): High. The summary needs to accurately reflect the sales data for the specified month.
Subtasks:

1) Query sales transactions for the specified month and year.
2) Aggregate data to calculate:
    - Total number of vehicles sold.
    - Total gross sales income.
    - Total net income (sales price minus purchase price and total expenses).
3) Sort the results by year and month in descending order.
4) Format the summary report for display.

Abstract Code

Generate Monthly Sales Summary
User selects ***Generate Monthly Sales Summary***

Prompt user for the *year* and *month*

Query the database:
- Retrieve all sales transactions for the specified month and year.

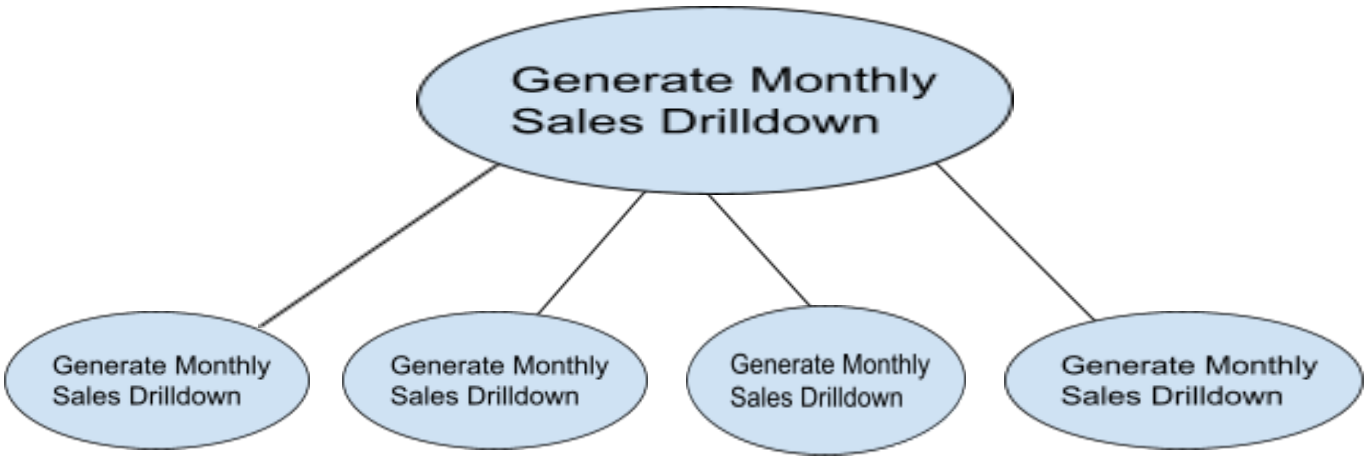
If the results are found:
- Initialize the counters for total vehicle sold, total gross income, and total net income.
- For each sales transaction:
    - Increment the total vehicles sold.
    - Add sale price to total gross sales income.
    - Subtract purchase price and the total expenses from sales price to calculate net income.

Prepare the summary report:
- Display total vehicles sold.
- Display total gross income.
- Display total net income.

If no sales transactions are found:
- Display a message indicating that no sales occurred for the specified month.

Generate Monthly Sales Drilldown

Task Decomposition



Lock Types: Read only on Sales Transaction table, Read only on Salesperson table.

Number of Locks: Two (one for reading sales transactions, one for reading salesperson details).
Enabling Conditions: The specified year and month must have sales transactions recorded.
Frequency: 20-30 drilldowns per month - estimated.
Consistency (ACID): Critical. The operation must have accurate reporting of sales data and also maintain consistency across the transactions.
Subtasks:

1) Get the specified year and month from the user.
2) Query the sales transaction table for sales data matching the specified year and month.
3) Aggregate the number of vehicles sold and total sales amount per salesperson.
4) Sort the results by the number of vehicles sold, then by total sales amount.
5) Display the drilldown report with salesperson name, vehicle counts, and total sales.

Abstract Code

Generate Monthly Sales Drilldown
Salesperson selects ***Generate Monthly Sales Drilldown*** button

Input the *year* and *month*.

Validate input for required fields and correct formats.

If validation is successful:
    Retrieve sales data for the specified *year* and *month*:
        - Query sales transactions table for records with matching *year* and *month*.

Aggregate results:
- For each salesperson:
    - Count the number of vehicles sold.
    - Sum total sales amount.

Sort results by:
- Number of vehicles sold (descending).
- Total sales amount (descending).

Display drilldown report:
- For each salesperson in results:
    - Show first name, last name, number of vehicles sold, and total sales amount.

Else:
    Return to form with error messages.
    Specify which fields are invalid.