Data types

Customer

Attribute	Data type	Nullable
phone_number	String	FALSE
street	String	FALSE
city	String	FALSE
state	String	FALSE
postal_code	String	FALSE
email	String	TRUE

Individual

Attribute	Data type	Nullable
drivers_license	String	FALSE
first_name	String	FALSE
last_name	String	FALSE

Business

Attribute	Data type	Nullable
business_tin	String	FALSE
business_name	String	FALSE
primary_contact	String	FALSE
title	String	FALSE

Users

Attribute	Data type	Nullable
username	String	FALSE
password	String	FALSE
first_name	String	FALSE
last_name	String	FALSE
role	String	FALSE

Managers

View of Users table

Inventory Clerk

View of Users table

Salesperson

View of Users table

Owner

View of Users table

Vehicle

Attribute	Data type	Nullable
vin	String	FALSE
manufacturer	String	FALSE
model	String	FALSE
model_year	Integer	FALSE
vehicle_type	String	FALSE

inventory_entry_date	Date	FALSE
condition	String	FALSE
kbb_price	Float	FALSE
mileage	Float	FALSE
color	List <String>	FALSE
description	String	FALSE
inventory_exit_date	Date	TRUE
sales_price	Float	FALSE

Vendor

Attribute	Data type	Nullable
vendor_name	String	FALSE
street	String	FALSE
city	String	FALSE
state	String	FALSE
postal_code	String	FALSE

Recall

Attribute	Data type	Nullable
nhtsa_number	String	FALSE
manufacturer	String	FALSE
description	String	FALSE

Repair

Attribute	Data type	Nullable
start_date	Date	FALSE
end_date	Date	FALSE
total_cost	Float	FALSE

description	String	FALSE
status	String	FALSE

Manufacturer

Attribute	Data type	Nullable
manufacturer	String	FALSE

VehicleType

Attribute	Data type	Nullable
type_name	String	FALSE

Business constraints

1. Customer
    - State should be exactly two characters long
    - Email should be in a valid email-format
    - Driver's license number must be unique
2. Business
    - Business tax identification number should always be 9 digits long and unique
3. Vehicle
    - VIN should always be 17 alphanumeric characters long and unique
    - The maximum value of modelYear is (current year + 1). For the purpose of this exercise, we are fixing the minimum value of modelYear to 1900
    - Sales price of the vehicle cannot be changed
    - Sales price must be calculated as 125% of the original purchase price combined with 110% of any repair costs associated with the vehicle
    - Inventory Exit Date can only be the current date
    - Manufacturer Name must be unique
    - A newly-added car will show $0 total for repairs because it has no repairs yet
4. Repair
    - Vendor Name, Start Date, and VIN have to be unique for every repair
    - Cars under repair cannot be sold or returned in public search results
    - The repair duration, defined as the period between the start and end dates should not overlap. For example, if a repair ends on 17th Jun, the next repair can start only on or after 18th Jun
    - A newly added repair should have the status of "pending" even if it starts on the day it is entered – the clerk will need to update the repair to "in progress" on the vehicle detail page.
    - Once a repair has been marked as completed, the status cannot be updated
    - Updating the repair status must step through this order: "Pending", "In Progress", then "Complete"
5. Dealership Website
    - A User's username and password each must be greater than 0 characters
    - Login usernames must be unique
    - NHTSA recall campaign number must be unique
    - Every Authenticated User must have a role assigned to them
    - There can only be one Owner role assigned to a User

- The Color dropdown is a predefined list of single colors, but a multi colored car will still populate in search results if the color chosen matched one of the colors in the multi colored car
- If no selections or filters are made on the Search page and the user clicks the enter button by default all vehicles without pending repairs will populate
- When Adding a Vehicle, the “inventory entry date” field will auto populate to the current date with no option to edit
- For reports:
    - If a vehicle type has no unsold units, the report should display “N/A” for that vehicle type.
    - If a vehicle type has no sales history, the report should display “N/A” for that vehicle type.
    - If a vehicle type or condition has never been purchased, the report should display “$0” for that result.

Task decomposition

Login

Task Decomposition of Login

Lock types	Read-only on User table
Number of locks	Single
Enabling conditions	None
Frequency	Around 200 logins per day
Consistency (ACID)	Not critical, order is not critical
Subtasks	Mother task not needed. No decomposition needed

Abstract Code Login:

- User enters *username*, *password* input fields.
- If data validation is successful for both *username* and *password* input fields, then:
    - When Login button is clicked:
        - If User record is found but user.password != '$Password' OR user not found:
            - Go back to Login form, with error message.
        - Else:
            - More options appear on Search Page
                - All authenticated Users see Search by VIN option
                - If User == “Inventory Clerk” OR “Owner”:
                    - Add Vehicle option appears
                    - Add Repair option appears
                - If User == “Manager” OR “Owner”:
                    - Find all repairs where status != “Complete” and display Number of Vehicles on page
                    - ***Reports*** link appears on page

Search

Task Decomposition of Search

Lock types	Two read-only on tables: Vehicle and Repair
Number of locks	Several different schema constructs are needed
Enabling conditions	None
Frequency	Around 1000 searches per day
Consistency (ACID)	Not critical, order is not critical
Subtasks	Mother task not needed. No decomposition needed

Abstract Code of Search:

- Public Facing Search Screen defined as:
    - Show login screen
    - Show total number of cars available for purchase in the system
        - Find all vehicles where sales date is null from Vehicle table
        - Find all vehicles with or without a repair and filter vehicles with status != “Pending” OR “In Progress” from Repair table
        - From vehicles that are not sold or under repair, count unique VIN
    - Prepopulate dropdown lists from database:
        - VehicleType
            - Select all VehicleTypes from Vehicle table
        - Manufacturer
            - Select all Manufacturer from Repair table
    - Static Lists predefined on customer specifications document
        - Pull Colors from list of colors predefined on customer specifications document
        - Pull ModelYear from range 1900 to current year + 1
    - User selects from none, one, or more of the following dropdowns:
        - VehicleType
        - Manufacturer
        - ModelYear
        - Color

    - User additionally can enter keywords in freeform field searching Manufacturer, ModelYear, ModelName, and Description as entire text or substring
    - If user inputs text in field, and selects from dropdown(s) and presses the ***Enter*** button
        - Find all vehicles without pending repairs based on user input and display
        - Selecting a result will load that vehicle's detail page
    - If no cars found, display:
        - "Sorry, looks like we don't have that in stock!"
    - If no criteria selected and User presses the ***Enter*** button
        - Find all vehicles without pending repairs and display
    - Else:
        - For each vehicle return VIN, VehicleType, ModelYear, Manufacturer, Model, Color (single row), Mileagle, and SalesPrice
            - Sort by VIN by default, otherwise sorted by specified field, and display
- If authenticated User:
    - Show Public Facing Search Screen
    - Provide additional freeform text field for searching by VIN
        - User inputs VIN in input field and clicks the ***Enter*** button
            - Find all vehicles where vehicle.VIN == *VIN* and display
    - If authenticated User.Role == "Inventory Clerk":
        - Search results also include vehicles repairs "pending" or "in progress"
        - Show ***Add Vehicle*** button
    - Else if authenticated User.Role == "Salespeople":
        - Search results display same as public facing search
    - Else if authenticated User.Role == "Managers":
        - Show total number of cars with repairs "pending" or "in progress"
        - Provide additional dropdown for filtering search results by sold vehicles, unsold vehicles, or all vehicles
        - Search results also include vehicle repairs "pending" or "in progress"
        - Show ***Reports*** link
    - Else if authenticated User.Role == "Owner":
        - Search results also include vehicle repairs "pending" or "in progress"
        - Show total number of cars with repairs "pending" or "in progress"

        - Provide additional dropdown for filtering search results by sold vehicles, unsold vehicles, or all vehicles
        - Show ***Add Vehicle*** button
        - Show ***Reports*** link
- Else if not authenticated User:
    - Show Public Facing Search Screen

Search/Add Customer

Task decomposition

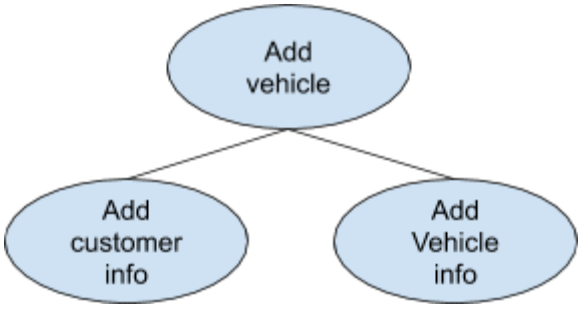
Lock types	Write lock on Customer table
Number of locks	Single
Enabling conditions	Authenticated user must be either an Inventory Clerk or Owner and they should have clicked Add Vehicle link from the search page
Frequency	Less than 20 per day
Consistency (ACID)	Not critical, order is not critical
Subtasks	Mother task not needed. No decomposition needed

Abstract code of Search/Add Customer

- Provide a drop-down to select the type of customer with the following options
    - Individual
    - Business
- If “Individual” is selected
    - Provide a text field to the user to input a *driver’s license number*
    - User inputs a String in the input field and presses the ***Enter*** button
        - Display the customer for which customer.drivers_license == *user_input*. The following fields are displayed: Last name, first name, driver’s license number, phone number, address
- If “Business” is selected
    - Provide a text field to the user to input a *business tax identification number*
    - User inputs a String in the input field and presses the ***Enter*** button

        - Display the customer for which customer.business_tin == *user_input*. The following fields are displayed: Business name, business tax identification number, primary contact, title, phone number and address
- No writes to the Customer table happens if user selects the customer
- Add customer fields for adding a new customer are displayed when the Sales Order form or Purchase Vehicle form are selected
    - Show a drop-down to select the type of customer with the following two options
        - Individual
        - Business
    - If “Individual” is selected from the drop-down, provide the following fields
        - First name
        - Last name
        - Driver’s license
        - Phone number
        - Street
        - City
        - State
        - Postal code
        - Email
    - If “Business” is selected from the drop-down, provide the following fields
        - Business name
        - Business tax identification number
        - Primary contact name
        - Primary contact title
        - Phone number
        - Street
        - City
        - State
        - Postal code

Add Vehicle



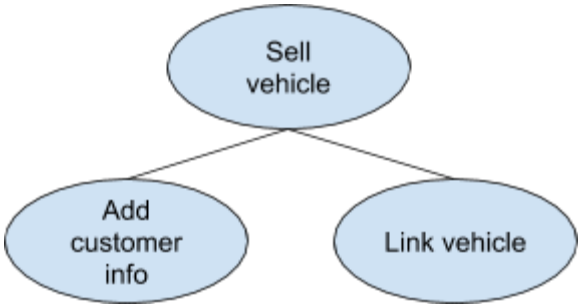
Task decomposition of Add Vehicle

Lock types	Write locks on Customer and Vehicle tables
Number of locks	2 locks
Enabling conditions	Authenticated user must be either an Inventory Clerk or Owner and they should have clicked Add Vehicle link from the search page
Frequency	Less than 20 per day
Consistency (ACID)	Not critical, order is not critical
Subtasks	Mother task not needed. No decomposition needed

Abstract code for Add Vehicle

- First populate the customer details following the actions in Search/add customer task
- To add a vehicle, provide the following fields to the user for input
    - VIN
    - Manufacturer (User selects one value from a dropdown populated using list of manufacturers in database)
    - Vehicle type (User selects one value from a dropdown populated using list of vehicle types in database)
    - Model year (User selects one value from a dropdown ranging from 1900 to (current_year + 1))
    - Color (User selects one or more values from a static dropdown)
    - Model name
    - Vehicle condition (User selects one value from dropdown populated using the following values: Excellent, Very Good, Good, Fair)
    - KBB value
    - Mileage
    - Description
    - Auto-populate current date in the “inventory entry date” field with no option to edit
    - Set sales price = 125% of KBB value
- After user populates all the provided fields and presses the ***Save*** button.
    - Save customer data to the Customer table only if customer is a new customer
    - Save vehicle data to Vehicle table

Sell Vehicle



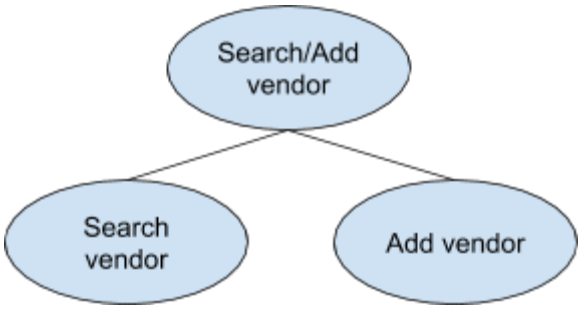
Task decomposition of Sell Vehicle

Lock types	Write locks on Vehicle and Repair tables
Number of locks	2 locks
Enabling conditions	Authenticated user must be either a Salesperson or Owner and they should have clicked Sell Vehicle link from the search page
Frequency	Less than 20 per day
Consistency (ACID)	Not critical, order is not critical
Subtasks	Mother task not needed. No decomposition needed

Abstract code of Sell Vehicle

- First populate the customer details following the actions in Search/add customer task
- On the Sales order form, provide a field to enter the *sales date*
    - User enters the *sales date* and presses the ***Enter*** button
        - Update the vehicle record with the “sales date” in the Vehicle table

Search/Add Vendor



Task decomposition

Lock types	Write locks on Vendor table
Number of locks	Single
Enabling conditions	Authenticated user must be either Inventory Clerk or Owner; and should have clicked “Add vendor”
Frequency	Less than 20 per day
Consistency (ACID)	Not critical, order is not critical
Subtasks	Mother task not needed. No decomposition needed

Abstract code of Search/Add Vendor

- Provide a text field to the user to input a *vendor name*
    - User inputs a String in the input field and presses the ***Enter*** button
        - Display the vendors that match the criteria vendor.vendor_name == *user_input*. The following fields are displayed: Vendor name, phone number, address (street, city, state and postal code)
- If user selects a vendor from the output list, then fields in the parent form (Add Repair form) get populated with the selected vendor
- Also, provide an add vendor fields for adding a new vendor
    - Vendor name
    - Phone number
    - Street
    - City
    - State
    - Postal code
- Update Vendor table with *user input*

Add Repair

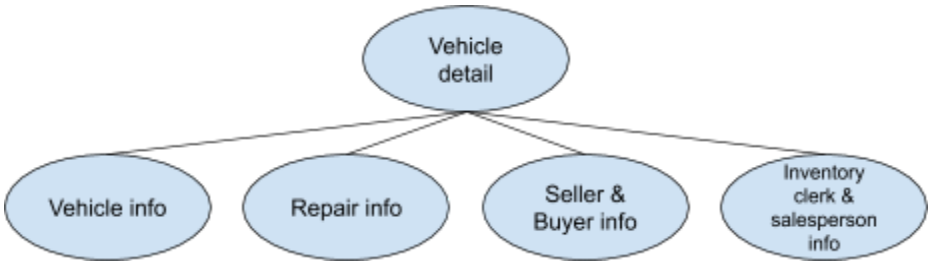
Task decomposition of Add Repair

Lock types	Write locks on Repair table, read lock on Vehicle table
Number of locks	2 locks
Enabling conditions	Authenticated users should be either an Inventory clerk or Owner; and should have clicked the ***Add Repair*** button from the Vehicle Detail page
Frequency	Less than 200 per day
Consistency (ACID)	Not critical, order is not critical
Subtasks	Mother task not needed. No decomposition needed

Abstract code for Add repair

- First populate the vendor details following the actions in Search/add vendor task
- Find all applicable recalls by filtering the Recall table
    - User enters the *sales date* and presses the ***Enter*** button
        - Update the vehicle record with the “sales date” in the Vehicle table

View Vehicle Detail



Task Decomp of Vehicle Detail:

Lock types	Read lock on Vehicle table
Number of locks	Single
Enabling conditions	The vehicle detail page is reached via the Search page
Frequency	Less than 200 per day
Consistency (ACID)	Not critical, order is not critical
Subtasks	Mother task not needed. No decomposition needed

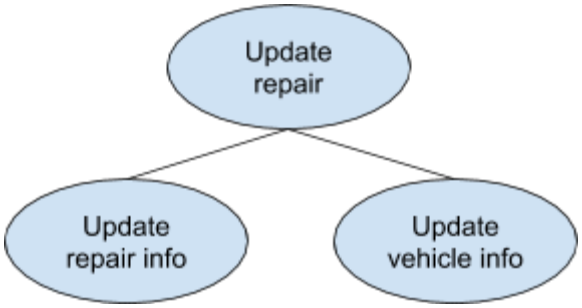
Abstract Code of View Vehicle Detail:

- The vehicle detail page is reached via the Search page
    - Pull information for a vehicle where vehicle.VIN == *VIN* for the selected vehicle
- Define Public Facing View Vehicle as:
    - For the selected vehicle get vehicle type, Model Year, Manufacturer, color(s), mileage, sales price, and the description of the car from the Vehicle table
- If authenticated User == “Manager”:
    - Show Public Facing View Vehicle page
    - Additional fields are populated in the view including:
        - Seller Information
            - For the selected vehicle get the seller and buyer information from the Customer table except their driver’s license or tax ID number
                - Name (first and last) of the inventory clerk that purchased the car

            - Total cost of repairs, and a repairs section listing details for all repairs just like would be shown on an inventory clerk's view.
            - Buyer's contact information (everything except their driver's license or tax ID number)
        - Repairs Information
            - For the selected vehicle get all the repairs information from the Repair table
                - Total for all repair costs
                - Details for all repairs: vendor, start date, end date, status, cost, and the recall number, if applicable
        - Vehicle Information
            - For the selected vehicle get original purchase price and purchase from the Vehicle table
                - Original purchase price
                - Purchase date
                - Sales date
        - User Information
            - For the selected vehicle get the sales person's name from the User table
                - Salesperson's name (first and last)
- If authenticated User == "Salespeople":
    - Show Public Facing View Vehicle page
    - Additional link is displayed to sell the car, ***sell vehicle*** , and link to the Sales Order form
    - If User clicks on ***sell vehicle*** link:
        - Load Sales Order form
- If authenticated User == "Inventory Clerk":
    - Show Public Facing View Vehicle page
    - Additional fields are populated in the view including:
        - Vehicle Information
            - For the selected vehicle get original purchase price and purchase from the Vehicle table
                - Original purchase price
        - Repairs Information
            - Show the same information as a Manager's view for the repair section
    - There should also be a popup to display the repair description when user clicks the ***repair description*** button against each repair record

    - A button for ***update repair*** should also be displayed and when clicked the Update Repair form will display
        - User will invoke the update repair task
    - A link for ***add repair*** should also be displayed and when clicked the Add Repair form will display
- If authenticated User == " Owner ":
    - Show Public Facing View Vehicle page
    - Additional fields that are a union of the view for Managers, Salespeople, and Inventory Clerk will display
    - If vehicle displayed in details page is sold:
        - Disable ***add repair*** link
        - Disable ***update repair*** button
        - Disable ***sell vehicle*** link

Update Repair



Task Decomp of Vehicle Detail:

Lock types	Write locks on Repair and Vehicle table
Number of locks	Single
Enabling conditions	User clicks ***update repair*** link from vehicle detail page and be a "Owner" or "Inventory Clerk"
Frequency	Less than 100 per day
Consistency (ACID)	Not critical, order is not critical
Subtasks	Mother task not needed. No decomposition needed

Abstract Code of Vehicle Detail:

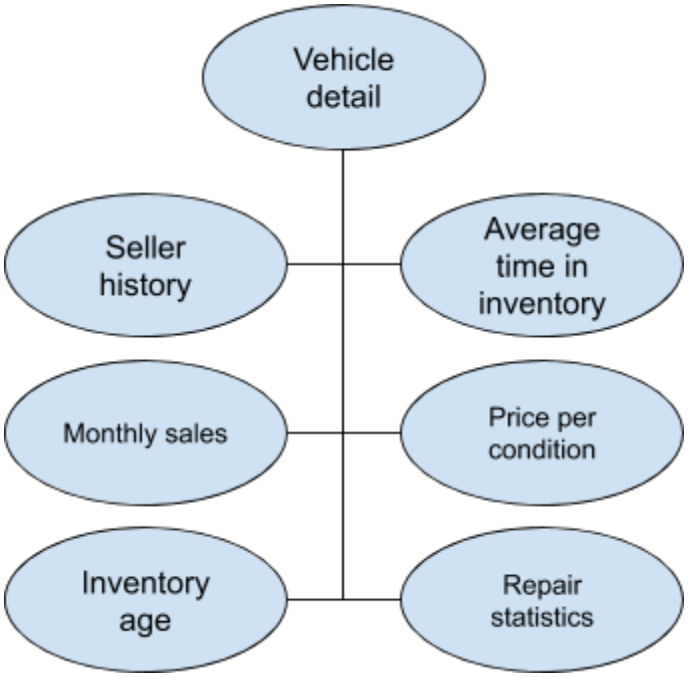
- For the selected vehicle and selected repair:
    - User is presented with a drop down for the repair status:

- Dropdown is populated with 3 options: “Pending”, “In Progress”, and “Complete”
- The default selection on the dropdown will be the current status of the vehicle
- User can update the status to “In Progress” or “Complete”
- User clicks ***save*** button to save the status of the repair and the status is written to the Repair table

○ If the updated status == “Complete”:

- Find the corresponding vehicle, get the current sales price and change it to current sales price + 110% of the total cost of the repair

View Reports



Task Decomp of View Reports:

Lock types	Write locks on Repair and Vehicle table
Number of locks	Single
Enabling conditions	User clicks ***reports*** link from search page and be an “Owner” or “Manager”
Frequency	Less than 100 per day

Consistency (ACID)	Not critical, order is not critical
Subtasks	Mother task not needed. No decomposition needed

Abstract Code of View Reports:

- If User.Role == “Manager” OR “Owner” :
    - User is able to click reports link from the search page and is presented with dropdown of all the report types
    - If a report is selected from the dropdown the selected report will be displayed
    - Seller History Report
        - Find all purchased vehicles from the Vehicle table
        - For each vehicle sold get the name of the seller from the Customer table
            - For each seller find all the vehicles sold:
                - Count vehicles as total number of vehicles
                - Take average purchase price
                - For each vehicle find all the repairs and sum the number of repairs
                - Average number of repairs is calculated as sum of all repairs for a seller / count of vehicles sold by them
        - Display all metrics and sort by total number of vehicles sold in descending order followed by average purchase price ascending
        - If the average number of repairs >= 5 highlight that seller in red
    - Inventory Age Report
        - Find all purchased vehicles not sold from the Vehicle table
        - By vehicle type for every vehicle:
            - Calculate age of vehicle as current date - purchased date in days
            - Get minimum of age
            - Get maximum of age
            - Get average of age
        - Display all metrics for Inventory Age Report
    - Average Time in Inventory Report
        - Find all sold and purchased vehicles from the Vehicle table
        - By vehicle type for every vehicle:
            - Calculate age of vehicle as sales date - purchased date in days as time in inventory
            - Get average of the time in inventory
        - Display all metrics for Average Time in Inventory Report

- Price Per Condition Report
  - Find all vehicles from the Vehicle table
  - By vehicle type and condition (Excellent, Very Good, Good, Fair) for every vehicle:
    - Get average price paid
  - Display all metrics for Price Per Condition Report and pivot on vehicle type displayed as rows and condition as columns
- Repair Statistics Report
  - Find all vendor names from Vendor table
  - Find all repairs from Repair table
  - For each vendor get all repairs:
    - Count the repairs that have a status == “Complete”
    - Sum the repair cost as total dollar amount spent on completed repairs
    - Get average number of repairs per vehicle
    - Calculate age as end date - start date in days as length of time to complete repair
      - Get average length of time to complete repair
  - Display all metrics for Repair Statistics Report
- Monthly Sales Report
  - Find all sales dates, and sales price, purchase price from Vehicle table
  - Find all repair costs from Repair table
  - Find salespeople from Users table
  - From sales, repairs, and salespeople:
    - By year, month, and salesperson:
      - Count all sales dates as total number of vehicles sold
      - Sum sales price as total sales income
      - Sum purchase costs as total purchase cost
      - Sum repair costs as total repair cost
      - Calculate total net income as total sales income - total repair cost - total purchase cost
  - Display all metrics for Monthly Sales Report by year and month aggregation and sort in descending order
  - Drilldown on year and month from results is available as a second display of this report
    - Display all metrics for Monthly Sales Report by year, month, and salesperson ordered by top performing salespeople in descending order (in the event of a tie, the salesperson who

has sold the highest dollar value will be considered the top salesperson)