Team 23 Phase 1 Report

Table of Contents:

Data Types

LoggedInUser

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

Customer

Attribute	Data type	Nullable
CustomerPhone	String	Not Null
CustomerStreet	String	Not Null
CustomerCity	String	Not Null
CustomerState	String	Not Null
CustomerPostalCode	String	Not Null
CustomerEmail	String	Null

Individual (Customer)

Attribute	Data type	Nullable
CustomerFirstName	String	Not Null
CustomerLastName	String	Not Null
CustomerDriverLicense	String	Not Null

Business (Customer)

Attribute	Data type	Nullable
B_Name	String	Not Null
TIN	String	Not Null
C_Name	String	Not Null
Title	String	Not Null

Vehicle

Attribute	Data type	Nullable
VIN	String	Not Null
VehicleType	String	Not Null
ModelName	String	Not Null
ModelYear	Integer	Not Null
ManuName	String	Not Null
Colors	List <String>	Not Null
Mileage	Float	Not Null
OptionalDescription	String	Null

InventoryVehicle

Attribute	Data type	Nullable
SalesPrice	Float	Not Null

Repairs

Attribute	Data type	Nullable
RepairDescription	String	Not Null
RepairStatus	String	Not Null
StartDate	Date	Not Null
EndDate	Date	Not Null
TotalCost	Float	Not Null

Recall

Attribute	Data type	Nullable
NHTSARecallNumber	String	Not Null
RecallDescription	String	Not Null
Manufacturer	String	Not Null

Vendor

Attribute	Data type	Nullable

VendorName	String	Not Null
VendorPhone	String	Not Null
Street	String	Not Null
City	String	Not Null
State	String	Not Null
Postal_Code	String	Not Null

SaleTransaction

Attribute	**Data type**	**Nullable**
SalesDate	Date	Not Null

P_Transaction

Attribute	**Data type**	**Nullable**
VehicleCondition	String	Not Null
PurchaseDate	Date	Not Null
KBB_Price	Float	Not Null

Business Logic Constraints:

User

- A user has to be manually added and set permissions by database administrator.
- A user can only have one of the permission sets.
- Database administrator determines a unique username and password (plaintext) to identify a user.
- Appropriate error messages will be displayed if incorrect credentials are submitted during a login.

Vehicles

- List of vehicle types and manufacturer name should be updated regularly by database administrator.
- Vehicle model years cannot exceed the current year plus one and it must include century digits.
- List of vehicle colors is not changeable.
- Vehicle conditions should be one of Excellent, Very Good, Good, Fair.

Inventory Vehicles

- The sales price of inventory vehicle is calculated as 125% of the original purchase price (the price Burdell's paid to buy the car) combined with 110% of any repair costs also associated with the vehicle.
- The sales price of inventory vehicle cannot be changed.

Repairs

- A newly-added car will show $0 total for repairs.
- Once a repair has been marked as completed, its status can no longer be updated.
- A newly added repair should have the status of "pending" and status can be updated from "pending" to "in progress" to "completed".
- Repair start date should not be after the repair end date.
- If one vehicle has many repairs, repair dates of each repair cannot overlap.
- Repair cannot be added on a car that has been sold.

Search for vehicles

- The search result should meet all the search criteria users selected or inputted.
- The search results should be sorted by VIN. Allowing the user to sort results by other columns is optional.
- Public users and salespeople cannot search vehicles under repair.

Task Decomposition And Abstract Code:

Login

Task Decomposition

Lock Types: Read-only on “User” table
Number of Locks: Single read-only lock
Enabling Conditions: None
Frequency: Around 100 logins per day
Consistency: Since order is not important, consistency is not critical
Subtasks: Since no task decomposition needed, no need of a Mother Task.

Abstract code

- User sees a ***Login*** button on the Public **Search** form and clicks on it.
- A login prompt asking for username(‘$username’) and password(‘$password’) appears.
- User enters username and password in the corresponding input fields.
- Data validation is done on username and password fields.
- If the data validation is successful:
    - When “***Enter***” button is clicked by the user:
        - If ‘$username’ belongs to a user in User table:
            - if user.password!=’$password’:
                - redirect to **Login** form with error message “Incorrect credentials for the user!”
            - if user.password==’$password’:
                - Go to **Search** form for logged-in users
        - else display on **Login** form“User does not exist!”
- Else an error message is displayed on the **Login** form that says “Username and password input fields are invalid!”

Search for Vehicle

Task Decomposition

Lock Types: Read-only lookup of Vehicle, User, Inventory vehicle and Repair.
Number of Locks: 4 read-only locks. (Several different schema constructs are needed.)

Enabling Conditions: None for public search; Enabled by user login for privileged user search.
Frequency: High (around 300 per day).
Consistency (ACID): not critical
Subtasks: Mother task is required to coordinate subtasks. Subtasks are not in parallel since only one type of search performed at a time

Abstract Code:

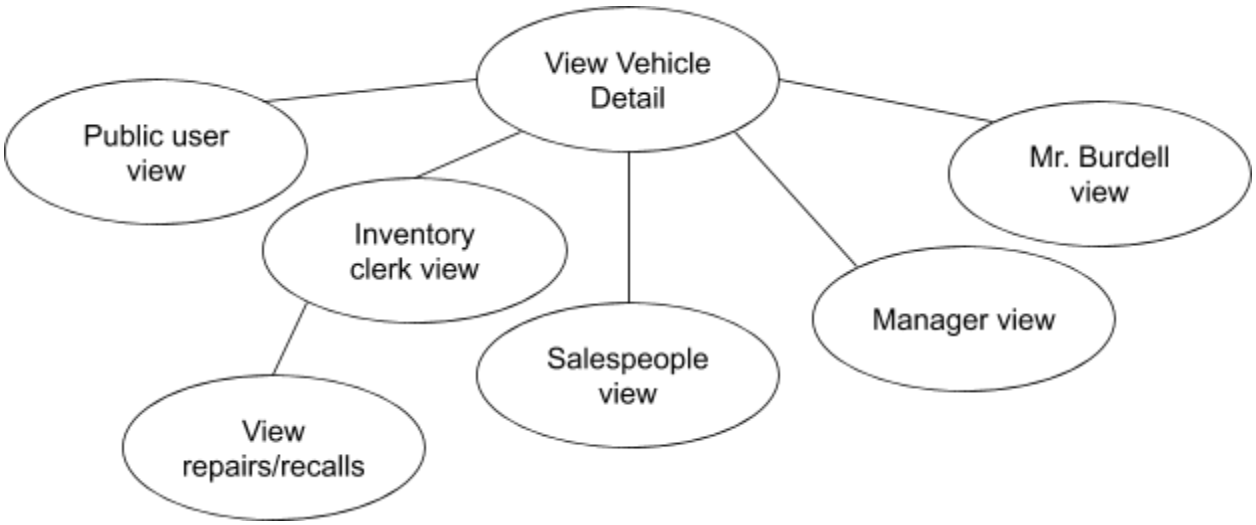
- Display ***Search for vehicle, Login*** buttons in the public-facing search screen.
- Query Inventory vehicle and calculate the total number of vehicles available for purchase.
- Display " There are '$calculation_result' vehicles available for sale" prominently.
- When user clicks ***Login*** button - Run **Log in** subtask.
- Once a user is logged in, instead of the ***Login*** button, ***Logout*** button appears in the search page for that user.
- When logged-in user clicks ***Logout*** button - Invalidate login session and go back to the **Login** form

- When User clicks ***Search for vehicle*** button, query user login status by *Username:*
    - If no login information detected, run **Public search** task:Display **Search for vehicle** form with drop-downs of *Vehicle type, Manufacturer, Model year, Color(s) and a "Search by keyword" text input.*
    - if user is logged-in user:
        - If user is sales people or inventory clerks - Run **Inventory clerks and salespeople search** task
            - Display **Search for vehicle** form with drop-downs of *Vehicle type, Manufacturer, Model year, Color(s),* a "*Search by keyword*" text input and a "*Search by VIN*" text input.
        - If user is Manager or Mr. Burdell - Run **Manager/Mr. Burdell search** task
            - Display **Search for vehicle** form with drop-downs of *Vehicle type, Manufacturer, Model year, Color(s), Sold/Unsold*, a "*Search by VIN*" text input and a "*Search by keyword*" text input.
            - Display ***View Seller History Report, View Inventory Age Report, View Average Time in Inventory Report, View Price Per Condition Report, Repair Statistics Report*** and ***View Monthly Sales Report*** buttons
- Inputted data on user selected search criteria, texted VIN or texted "Keyword" is used to retrieve search results as follows:
    - Lookup "Vehicles" table and retrieve results that match all the criteria and contain the keyword(AND logic for all selected criteria and keyword).
    - Query information about sales transaction and repair for the vehicles on the search result, by looking up on "Sales Transaction" and "Repair" tables.
    - If the user is a public user or sales people, run **Retrieve and display results for salespeople and public** subtask:
        - Delete vehicles having relationship with sales transaction(sold vehicle) or repair(repair vehicle) and
        - Return(Keep) vehicles ready for sale
    - If the user is inventory clerk, run **Retrieve and display results for inventory clerks** subtask:
        - Delete vehicles having relationship with sales transaction(sold vehicle)
        - Return(Keep) unsold vehicle (with/without repair)

    - If the user is Manager or Mr. Burdell, run **Retrieve and display results for Manager/Mr. Burdell** subtask:
        - Return(Keep) all the results retrieved from the lookup “Vehicles” table.
- If there are vehicles matching the search:
    - Sort vehicle by VIN
    - Display VIN, Vehicle type, Model Year, Manufacturer, Model, Color(s), Mileage and Sales Price for vehicles on the lookup result.
- If there is no vehicle matching the search keywords:
    - Display: “Sorry, it looks like we don’t have that in stock!”

View Vehicle Detail

Task Decomp



Lock Types: Read-only lock for Vehicle, Customer, User and Purchase Transaction;
Number of Locks: 4 read-only lock. Several different schema constructs are needed
Enabling Conditions: After user performed **Search for vehicle** task and selected a vehicle from the vehicle search results.
Frequency: Medium (around 50 per day).
Consistency (ACID): order and consistency are not critical
Subtasks: Mother task is required to coordinate subtasks. Subtasks are not in parallel since only one type of view vehicle detail performed at a time

Abstract Code:

- User selected a vehicle from vehicle search results.
- Run **View Vehicle Detail** task:
    - Find the selected vehicle using the VIN.
    - Identify the user by login status and username.
    - If the user is a public user, run **Public user view** subtask:
        - Query Vehicle table for information about the selected vehicle
        - Display the query result with *VIN, vehicle type, Model Year, Manufacturer, color(s), mileage, sales price,* and *the description* of the car.
    - If the user is inventory clerk, run **Inventory Clerk view** subtask:
        - Query Vehicle table for information about the selected vehicle and display *VIN, vehicle type, Model Year, Manufacturer, color(s), mileage, sales price and the description of the car.*
        - Find and display KBB_Price(original purchase price) associated with the vehicle by querying Purchase_transaction table.
        - Query Repair table to find total repair cost for the selected vehicle (calculate and display total of all repair costs).
        - Display each repair associated with the selected vehicle using ***view repairs/recalls*** subtask:
            - Display all repairs with *vendor, start date, end date, status, cost,* and *the recall number* if it is a recall.
            - To edit the repair status or to view description of repair, click on **Edit repairs/recalls status and view descriptions** button and it directs to **Edit repairs/recalls status and view descriptions** task.
        - Display an ***Add repair*** button:
            - If user presses the ***Add repair*** button, it directs to **Add repair/recall** task.
    - If the user is a Salespeople, run **Salespeople view** subtask
        - Query Vehicle table for information about the selected vehicle by VIN and display the query result with VIN, vehicle type, Model Year, Manufacturer, color(s), mileage, sales price, and the description of the car.
        - Display a “Sell” button:
            - If user press the “Sell” button, direct to task **Perform Sales**.
    - If the user is a Manager, run **Manager view** subtask
        - Query for information about the selected vehicle from the Vehicle table by VIN and display *VIN, vehicle type, Model Year,*

Manufacturer, color(s), mileage, sales price, the description of the car

- Query for information of total cost and repairs list from Repair table for the selected vehicle.
- Display each repair associated with the selected vehicle using ***view repairs/recalls*** subtask:
    - Display all repair with *vendor, start date, end date, status, cost,* and *the recall number* if it is a recall
- Calculate and display total of all repair costs.
- Query the Inventory clerk table and Purchase Transaction table to find who purchased the vehicle.
- Find and display the *First Name* and *Last Name* of the inventory clerk who purchased the vehicle.
- Query the Customer table and Purchase transaction table for finding who sold the vehicle to Burdell's.
- Find and display the seller's information:
    - Display *First Name, Last Name, Phone, Address* and *email* (if provided) for individual customer.
    - Display *B_Name, Primary Contact, phone, Address* and *email* (if provided) for business customer.
- Query the Purchase transaction table and display Vehicle condition, KBB_Price and PurchaseDate for the selected vehicle.
- If the selected vehicle has been sold from Burdell's,
    - Query for the information about Sales transaction and the Salespeople to find who sold the vehicle
    - Find and display Sales people's *First Name* and *Last Name from Salespeople table.*
    - Query information about Sales transaction and Customer to find which customer bought the vehicle from Burdell's.
    - Find and display the buyer's information
        - Display *First Name, Last Name, Phone, Address* and *email* (if provided) for individual customer
        - Display *B_Name, Primary Contact, home, Address* and *email* (if provided) for business customer

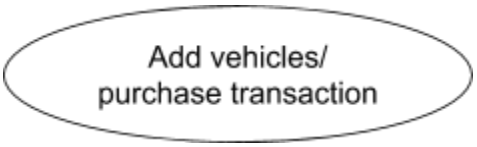
- If the user is Mr. Burdell, run **Mr. Burdell view** subtask:
    - Display "***View as Salespeople***", "***View as Inventory Clerks***" and ""***View as Manager***" buttons
    - If user press ***View as Salespeople*** button, run **Salespeople view** subtask

- If user press ***View as Inventory Clerks*** button, run **Inventory clerk view** subtask
- If user press ***View as Manager*** button, run **Manger view** subtask

---

Add vehicles/purchase transaction

Task decomposition:



Lock Types: Write-only on "Vehicles" table and read-only task on "Customers" table.
Number of Locks: 1 write lock and 1 read lock.
Enabling Conditions: During purchase transactions.
Frequency: Around 50 times per day.
Consistency: Since order is not important, consistency is not critical.
Subtasks: Since no task decomposition needed, no need of a Mother Task.

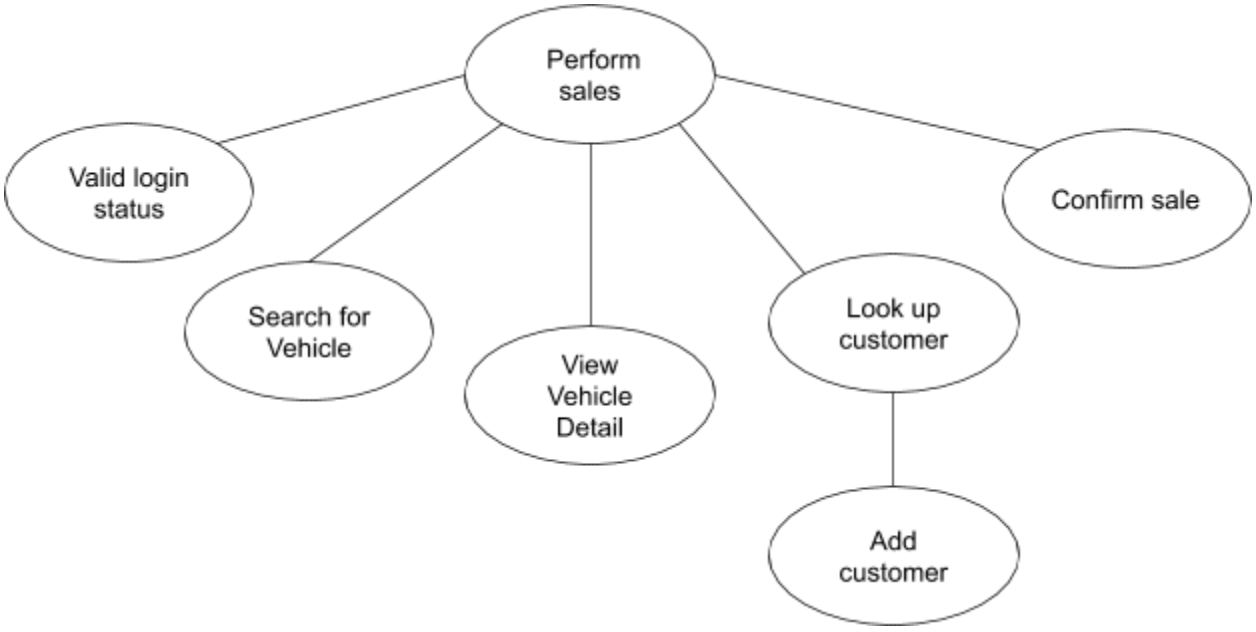
Abstract code:

During purchase transactions:

- click on ***Add vehicle*** button.
- the **Add vehicle** form appears:
  - Enter the VIN
  - Enter the type of vehicle from the list in database
  - Enter the manufacturer name from the list.
  - Enter the model name
  - Enter the model year
    - Check that model year <= current year +1
    - Check that model year includes century digits.
  - Enter the colors from the list of generic color names.
  - Enter the mileage (odometer reading).
  - Enter the description (optional).
- Click on ***Save*** button to save the vehicle data into the database.

Perform sales

Task Decomposition



Lock Types: Read-write lock of Inventory vehicle, write-only lock of Sales transaction
Number of Locks: One read-write lock and one write-only lock. Several different schema constructs are needed
Enabling Conditions: When sales people are logged in.
Frequency: Around 50 times per day
Consistency (ACID): Critical, because a sale transaction causes Inventory vehicle table and Sale transaction table change, and may cause Customer table change by adding new customers. All these changes should happen as a transaction (atomic) and in order.
Subtasks: Mother task is needed. All tasks must be done, but can't be done in parallel.

Abstract Code:

- Sales people logged in and press ***Search for vehicle*** button – Run **Search for vehicle** task
- Sales people select the vehicle for sale – Run **View vehicle detail** task
- Sales people click ***Sell*** button – Load **Sales order** form with a ***Look up Customer*** button and a confirm date text input
- Sales people press ***Look up Customer*** button on **Sales order** form – Run **Look up Customer** task
- If customer doesn't exist, run **Add customer** task
- Sales people selected the customer as a buyer

- Sales people confirm the sale by entering sales date on the **Sales order** form – Update Vehicle table, Inventory vehicle table and Sales transaction table

---

Look up customers

Task Decomposition

Lock Types: Read-only lookup of customer table
Number of Locks: Single
Enabling Conditions: Enabled in a vehicle sales transaction or in a vehicle purchase transaction, when a logged-in user press Look up Customer button.
Frequency: Around 100 times per day
Consistency (ACID): Order is not critical
Subtasks: Mother task is not needed. No decomposition needed

Abstract Code:

- User click ***Look up Customer*** button on **Sales Order** form (for sales people in vehicle sales transaction) or in **Add Vehicle** form (for inventory clerks in vehicle purchase transaction)
- Display **Look up Customer** form with *driver's license number/tax ID* text input and a ***Search*** button
- User input customer *driver's license number* or *tax ID* and press the ***Search*** button
- Query Customer table and lookup the customer whose *driver's license* or *tax ID* matches the text input.
- Display lookup result: customer whose driver's license or tax ID matches the input, if such a customer exists.
- If no result is found, show ***Add new customer*** button and display: "Customer does not exist, please add new customer"
    - If customer pressed ***Add new customer*** button, direct to **Add new customer** task

---

Add new customers

Task Decomposition

Lock Types: Write-only on “Customer” table.
Number of Locks: Single write-only lock.
Enabling Conditions: During sales or purchase transactions, after performing look-up customers.
Frequency: Around 100 times per day
Consistency: Since order is not important, consistency is not critical.
Subtasks: Since no task decomposition needed, no need of a Mother Task.

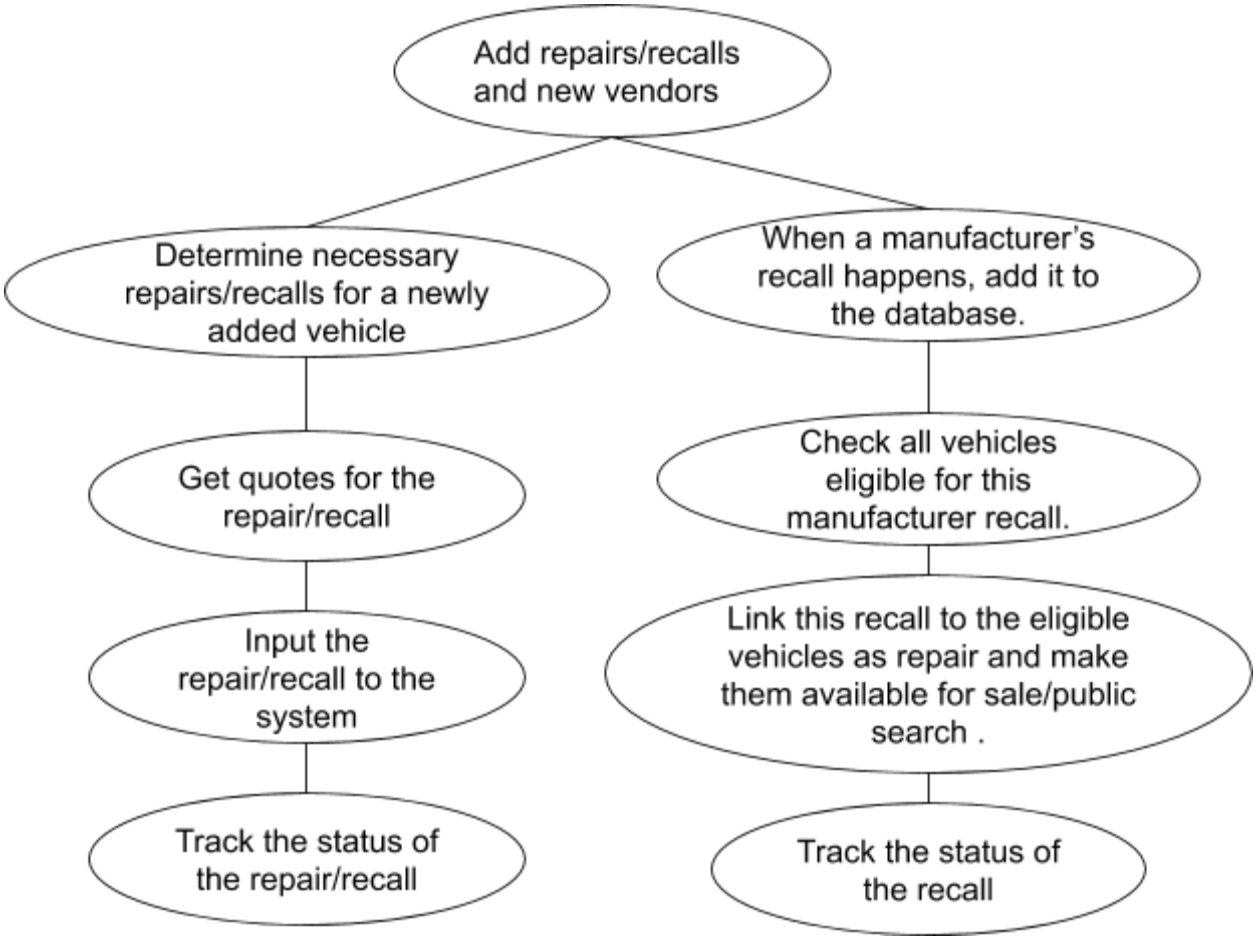
Abstract code

- To add the new customer, enter the following details on **Add new customers** form:
    - Enter address as street/city/state/postal code.
    - Enter phone number.
    - Enter email address (optional)
    - If the customer is an individual:
        - Enter first name.
        - Enter last name.
        - Enter driver’s license number.
    - If the customer is a business:
        - Enter Tax Identification Number.
        - Enter business name.
        - Enter primary contact name.
        - Enter primary contact title.
- Click on the ***Save*** button to save the customer data into the database.

Add repairs/recalls and new vendors

Task decomposition:

Lock Types: Write-only on “Repairs” table, and read and write on “Vehicles” table and “Recalls” table.
Number of Locks: 1 write lock and 2 read-write locks.
Enabling Conditions: When a vehicle is purchased or when manufacturer’s recalls happen.
Frequency: Around 50 times per day.
Consistency: Since order is important as the subtasks need to be performed in a sequential order, consistency is also critical.
Subtasks: Since task decomposition is needed, Mother Task is needed.



Abstract code:

When a vehicle is purchased:

- Determine necessary repairs/recalls for this new vehicle.
- Get quotes for the required repairs/recalls.
- Input the determined repair/recall to the system and link to the vehicle associated.
- If the repair is a recall:
    - Check if the recall is already in the database using look_up_recall(NHTSA recall campaign number).
        - If yes, then link the vehicle to this existing recall.
        - Else, add a new recall to the database using the fields under the add_new_recall heading in the **Add repairs/recalls and new vendors** form :
            - Enter NHTSA recall campaign number
            - Enter recall description.

        - Enter associated manufacturer name.
    - Enter the status of the recall as “pending”.
    - Save the recall by clicking on ***Save*** button.
- Perform Lookup_vendor to see if the selected vendor is already in the database.
- If Lookup_vendor returns false, the vendor is not in the database, then enter the following details in the fields under the add new vendor heading in **Add repairs/recalls and new vendors** form :
    - Enter the name of the vendor performing the repair/recall.
        - Check that the vendor name is unique.
    - Enter the Address of vendor (street/city/state/postal code).
    - Enter the Phone number of vendor.
    - Save the vendor information by clicking on ***Save*** button.
- Enter description of the repair.
- Enter start date and end date of the repair.
    - Check that the start date is not after the end date.
- Enter total cost of the repair.
- Enter the status of the repair.
- Track the status of all repairs/recalls.
- If the status of each repair and recall is ’completed’, add the vehicle to the inventory and make it available for public search and for sale.

When a manufacturer’s recall happens:

- Add the recall to the system using the fields under the add_new_recall heading in the **Add repairs/recalls and new vendors** form:
    - Enter NHTSA recall campaign number
    - Enter recall description.
    - Enter associated manufacturer name.
    - Save the recall by clicking on the ***Save*** button.
- Determine all vehicles eligible for the recall.
- Link this recall to all the eligible vehicles as repair.
- Make the eligible vehicles unavailable for public search or for sale.
- Get quotes from vendors for all eligible vehicles.
- Perform Lookup_vendor to see if the selected vendor is already in the database.
- If Lookup_vendor returns false, the vendor is not in the database, then enter the following details in the fields under the add new vendor heading in **Add repairs/recalls and new vendors** form:
    - Enter the name of the vendor performing the repair/recall.
        - Check that the vendor name is unique.
    - Enter the Address of vendor (street/city/state/postal code)
    - Enter the Phone number of vendor

- Enter description of the recall.
- Enter start date and end date of the recall.
    - Check that the start date is not after the end date.
- Enter total cost of the recall.
- Enter the status of the recall.
- Track the status of the recall for these vehicles.
- If the status of recall is 'completed' for a vehicle and if it has no other repairs/recalls not completed, add the vehicle back to the inventory and make it available for public search and for sale.

Edit repairs/recalls status and View descriptions.

Task decomposition:

Lock Types: Read and write on "Repairs" table, "Recalls" table and "Vehicles" table.
Number of Locks: 3 read-write locks.
Enabling Conditions: Inventory clerks, managers and Mr.Burdell can access "Edit repairs/recalls status" button on vehicle detail page for repairs/recalls that are not yet in "completed" status.
Frequency: Around 30 times per day.
Consistency: Since order is not important, consistency is not critical.
Subtasks: Since task decomposition is not needed, Mother Task is also not needed.

Abstract code:

When an inventory clerk or manager or Mr.Burdell accesses the vehicle detail page and clicks on ***edit repairs/recalls status*** button:

- All repairs/recalls with status as "pending" or "in progress" for that vehicle appears as a list.
- The corresponding description for each repair/recall is also populated in the list.
- The status of the repairs can be changed in this list.
- Once the status of a repair/recall is changed to "completed", that repair/recall is no longer accessible from ***Edit repairs/recalls status*** button.
- Once all repairs/recalls of a vehicle is in the status "completed", the vehicle is made available for public search and for sale.

View Seller History Report

Task Decomposition

Lock Types: Read-only lookup of Customer table, Purchase_transactions,Vehicle table, Repair table.
Number of Locks: 4 read-only locks
Enabling Conditions: Enabled by a user's login as Manager or Mr.Burdell.
Frequency: Low, Around 10 logins per day.
Consistency(ACID): Not critical. Order is not critical.
Subtasks: No Mother task is required. No decomposition is needed.

Abstract code

- Logged-in User Manager or Mr.Burdell click ***View Seller History Report*** button that was displayed on the search page.
- Query for all Seller and the total number of vehicles they have sold to Burdell's from Purchase_transactions table and Customers table, the average price for the vehicles from Purchase_transactions table and the average number of repairs per vehicle from Repair table.
- Open a new window (**View Seller History Report** form) to show the Query result. Each row shows one seller's name, the total number of vehicles this seller have sold to Burdell's, the average purchase price, the average number of repairs per vehicle.
- Sort above query results by the total number of vehicles sold descending and then by average purchase price ascending.
- If the average number of repairs per vehicle >= 5 for a seller, set this seller's row background to red.
- If user clicks the ***Close*** button, close the window.

View Inventory Age Report

Task Decomposition

Lock Types: Read-only lookup of Inventory Vehicle table and Sales Transaction table.
Number of Locks: 2 read-only lock
Enabling Conditions: Enabled by a user's login as Manager or Mr.Burdell.
Frequency: Low, Around 10 logins per day.
Consistency(ACID): Not critical. Order is not critical.
Subtasks: No Mother task is required. No task decomposition is needed.

Abstract code

- Logged in User Manager or Mr.Burdell clicks ***View Inventory Age Report*** button that was displayed on the search page.
- Query for unsold Inventory vehicles from Inventory vehicle table and Sales Transaction table.
- Compute the Inventory age in days for every vehicle by the formula: Inventory age(in days) = current date – Purchase Date
- Find the minimum, average, and maximum age of unsold Inventory vehicles, in days.
- Open a new window to show the Query result. Each row shows one type of vehicle, the minimum, average, and maximum age of unsold vehicles in inventory, in days.
- If one vehicle type has no unsold vehicles, display "N/A" for minimum, average, and maximum age.
- If user clicks the ***Close*** button, close the window.

View Average Time in Inventory Report

Task Decomposition

Lock Types: Read-only lookup of Inventory Vehicle table and Sales Transaction table.
Number of Locks: 2 read-only lock
Enabling Conditions: Enabled by a user's login as Manager or Mr.Burdell.
Frequency: Low, Around 10 logins per day.
Consistency(ACID): Not critical. Order is not critical.
Subtasks: No Mother task is required. No decomposition is needed.

Abstract code

- Logged in User Manager or Mr.Burdell click ***View Average Time in Inventory Report*** button that was displayed on the search page.
- Query for sold vehicles from Inventory vehicles table and Sales Transaction table.
- Compute the Average Time in Inventory in days by the formula: Average Time in Inventory (in days) = total (Sales date – Purchase Date) / total number of sold vehicles.
- Open a new window to show the result. Each row shows one type of vehicle, the Average Time in Inventory for sold vehicles, in days.
- If one vehicle type has no sales history, display "N/A" for Average Time in Inventory.
- If user clicks the ***Close*** button, close the window.

View Price Per Condition Report

Task Decomposition

Lock Types: Read-only lookup of Purchase Transaction table and Vehicle table.
Number of Locks: 2 read-only lock
Enabling Conditions: Enabled by a user's login as Manager or Mr.Burdell.
Frequency: Low, Around 10 logins per day.
Consistency(ACID): Not critical. Order is not critical.
Subtasks: No Mother task is required. No decomposition is needed.

Abstract code

- Logged in User Manager or Mr.Burdell click ***View Price Per Condition Report*** button that was displayed on the search page.
- Query for Purchase Price (Purchase Price = kbbPrice) for each vehicle from Purchase Transaction table and Vehicle table.
- Compute the price per condition by the formula: Price Per Condition = total Purchase Price for vehicles with the Condition / total number of vehicles with the condition.
- Open a new window to show the result. Each row shows one type of vehicle, Price Per Condition for each condition.
- If one vehicle type or/and condition have no purchase history, display "$0" for Price Per Condition.
- If user clicks the ***Close*** button, close the window.

.

View Repair Statistics Report

Task Decomposition

Lock Types: Read-only lookup of RepairVehicle table and Repair table
Number of Locks: 2 read-only locks
Enabling Conditions: Enabled by a user's login as Manager or Mr.Burdell.
Frequency: Low, Around 10 logins per day.
Consistency(ACID): Not critical. Order is not critical.
Subtasks: No Mother task is required. No decomposition is needed.

Abstract code

- Logged in User Manager or Mr.Burdell click ***View Repair Statistics Report*** button that was displayed on the search page.
- Query for Vendor name, repair, total cost, start date, end date for each vendor from Repair Vehicle table and Repair table.
- Compute the average number of repairs per vehicle for a specific vendor by the formula: average number of repairs per vehicle for the vendor = total number of repairs for the vendor / total number of Repair Vehicles for the vendor.
- Compute the total length of time (in days) for a vendor by the formula: length of time (in days) = end date – start date. Sum all repair's length of time (in days) to get total length of time (in days) for the vendor.
- Compute the average length of time (in days) for a specific vendor by the formula: average length of time (in days) for the vendor = total length of time (in days) for the vendor / total number of repairs for the vendor.
- Open a new window to show the result. Each row shows one vendor, the number of repairs completed by the vendor, the total cost spent on completed repairs, the average number of repairs per vehicle completed by that vendor, and the average length of time (in days) to complete repairs by that vendor..
- If user clicks the ***Close*** button, close the window.

View Monthly Sales Report

Task Decomp

Lock Types: Read-only lookup of SalesPeople, Sales transaction, Purchase transaction, Vehicle and Repair,
Number of Locks: 5 read-only locks. Several different schema constructs are needed
Enabling Conditions: Enabled by user logged in as manager or Mr. Burdell
Frequency: Around 50 times per day (the most frequently used report)
Consistency (ACID): is not critical
Subtasks: No Mother task is required. No decomposition is needed.

Abstract Code

- User logged in as a Manager or Mr. Burdell and selected Monthly Sales Report
- Query for information about all vehicle “Sales transaction”
- Sort “Sales transaction” by SalesDate with year and month descending.
- Display a list for all “Sale transaction” by year and month descending.
- Calculate and display the total number of vehicles sold by year and month descending.
- Calculate and display the total sales income by year and month descending.
- Query for information about each “Purchase transaction” and “Repair” corresponding to each “Sales transaction” from Purchase transaction table and Repair table.
- Calculate the total purchase price corresponding to the “Sales transactions” from Purchase transaction table.
- Calculate the total repair cost corresponding to the “Sales transactions” from Repair table.
- Calculate and display the total net income by subtracting total purchase price and repair cost from total sales income.
- Query for information about each “Salespeople” corresponding to each “Sales transaction” from SalesPeople table and Sales transaction table.
- Calculate the number of vehicles sold and the total sales by each “SalesPeople” by year and month.
- Sort the “SalesPeople” based on the total vehicle and then by total sales.

- Display “SalesPeople” first name and last name, the number of vehicles they sold and the total sales for the year and month.
- Determine and display the top sales person by the highest total sales by year and month.
