CS 6400 Database Systems Concepts and Design

Team 22 - Phase 1 Report

Jinjun Liu (jliu788)

Zijian Xie (zxie86)

Hui Xia (hxia40)

Chen Zhang (czhang613)

1. Tables

1.1. User Table

Attribute	Data Type	Nullable
Logged_in_User_Name	String	Not
Password	String	Not
Logged_in_First_Name	String	Not
Logged_in_Last_Name	String	Not
User_Type	String	Not

1.2. Repair Table

Attribute	Data Type	Nullable
VIN	String	Not
Repair_Status	String	Not
Vendor_Name	String	Not
Vendor_Address	String	Not
Vendor_Phone_Number	Integer	Not
Description	String	Not
Start_Date	Timestamp	Not
End_Date	Timestamp	Not
Repair_Cost	Decimal	Not
NHTSA_Recall_Campagin_Number	String	Null

1.3. Recall Table

Attribute	Data Type	Nullable
NHTSA_Recall_Campagin_Number	String	Not
Recall_Description	String	Not
Recall_Manufacture	String	Not

1.4. Vehicle Table

Attribute	Data Type	Nullable
VIN	String	Not
Vehicle_Type	String	Not
Vehicle_Manufacturer	String	Not
Model_Name	String	Not
Model_Year	Integer	Not
Vehicle_Color	String	Not
Mileage	Integer	Not
Sales_Price	Decimal	Not
Vehicle_Description	String	Null

1.5. Sell_Transaction Table

Attribute	Data Type	Nullable
VIN	String	Not
Sale_date	Timestamp	Not
Sale_price	Decimal	Not
Salesperson_username	String	Not

1.6. Buy_Transaction Table

Attribute	Data Type	Nullable
VIN	String	Not
Inventory_clerk_username	String	Not
Customer_tax_identification_number	String	Null, see constraints
Customer_driver_license_number	String	Null, see constraints
Purchase_date	Timestamp	Not
Purchase_price	Decimal	Not
Blue_book_value	Integer	Not
Purchase_condition	String	Not

1.7. Customer Table

Attribute	Data Type	Nullable
Address	String	Not
Phone	String	Not
Email	String	Null
Customer_name	String	Null, see constraints
Driver_license_number	String	Null, see constraints
Tax_identification_number	String	Null, see constraints
Business_name	Integer	Null, see constraints
Primary_contact_name	String	Null, see constraints
Primary_contact_title	String	Null, see constraints

2. Business Logic Constraints

- Only the Owner and Inventory Clerk have the access to read and write the Recall/Repair and Vehicle Tables.
- The *Model_Year* in the Vehicle Table cannot exceed current year plus one and must be 4-digits form.
- Inventory clerk and owner have links to Add Repair and Edit repair button;
- If Repair.Status is 'complete' or Vehicle.*VIN* not in Repair table, no **Add Repair** nor **Edit Repair** is allowed;
- If Vehicle.*VIN* in Sell_transcation table, neither changes in repair nor sell vehicle is allowed;
- If a VIN number is associated with an item in Sell_transcation, no further sale transaction on the item with same *VIN* is allowed.
- Repair.*End_date* >= Repair.*Start_date*;
- Repair.*Repair_cost* >= 0;
- Vehicle.*Vehicle_mileage* >= 0;
- Vehicle.*Model_year* >= 1900;
- Sell_transcation.*Sale_date* >= Buy_transcation.*Purchase_date*;
- In table Buy_transcation, one of the attributes *Customer_driver_license_number* and *Customer_tax_identification_number* will be null, and the other one will be not null.
- If person is selected in add new customer subtask, in table Customer, *Driver_license_number* and *Customer_name* are not null while *Tax_identification_number*, *Business_name*, *Primary_contact_name* and *Primary_contact_title* are forced to be null;
- If business is selected in add new customer subtask, in table Customer, *Driver_license_number* and *Customer_name* are forced to be null while *Tax_identification_number*, *Business_name*, *Primary_contact_name* and *Primary_contact_title* are not null;

3. Assumptions

- The *Username* on User table is unique.
- The *User_type* on the User table could only be one of "*Inventory_clerk*", "*Salespeople*", "*Manager*" and "*Owner*".
- The *NHTSA_Recall_Campagin_Number* in Recall table and Repair table is unique and may link with multiple repairs, and will not be changed.
- The *VIN* in Vehicle table is unique and will not be changed.
- The *Status* in the Repair table could only be one of "*Pending*", "*In Progress*" and "*Complete*".
- The vehicle in the Repair table which is under "*Pending*" or "*In Progress*" could not be presented in the public search results.
- Vehicle.*Sales_price* =Vehicle.*Purchase_price* * 125% + Repair.*Repair_cost* * 110%;

4. Task Decomposition and Abstract Code

Public Search

Task Decomposition:

Lock Types: Read-only on Vehicle, Sell_Transacation and Repair Table.

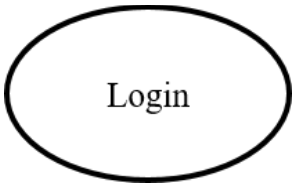
Number of Locks: 3
Enabling Conditions: None
Frequency: About 100 vehicle search requires per day
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show total number of vehicles that Repair.*Status* != "pending" or "In progress" and are not existed on Sell_Transacation table
- Show drop-down menus for vehicle type, manufacturer, model year and color selections
- Show blank field for Keyword input
- Show ***Search*** button and ***Login*** button
- Users can click the title of any given columns to sort on that respective column;
- Upon:
  - ❖ User enters vehicle type, manufacturer, model year, color or keyword, and clicks ***Search*** button:
    - ➢ If all the input fields are empty, show warning message;
    - ➢ If there are vehicles that exit on Vehicle table, then look up Repair table and Sell_Transacation table by VIN, if Repair.*Status* != "pending" or "in progress", or not exist on Repair table, and not exited on Sell_Transacation table, then show results, sorted by VIN;
    - ➢ If no vehicles meet the search criteria on Vehicle table, show error message "Sorry, it looks like we don't have that in stock!"
  - ❖ User clicks an individual result:
    - ➢ Go to **View Vehicle Detail** task;
  - ❖ User clicks ***Login*** button:
    - ➢ Go to **Login** task.

Login

Task Decomposition:



Lock Types: Read-only to User table
Number of Locks: Single
Enabling Conditions: Enabled by clicking on ***Employee Login***
Frequency: About 20 logins per day
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***Login*** and ***Cancel*** button
- Show empty field for *Username* and *Password* input.
- Upon:
  - ❖ User enters *Username*, *Password* and clicks ***Login*** button:
    - ➢ If either inputted *Username* or *Password* is empty, show warning message;

- ➢ If user inputted username is not found on User table, or user inputted username could be found on User table, but the user inputted password does not match the respective User.*Username*, clear the input field, show error message;
- ➢ If both the user inputted username and password match an item in User table, go to **Initial Search** page;

❖ User clicks ***Cancel*** button:

- ➢ ***G***o to **Public Search** page.

Initial Search

Task Decomposition:

Lock Types: Read-only to the Vehicle, Repair, Sell_Transaction Table
Number of Locks: 3
Enabling Conditions: Enabled by successful login.
Frequency: About 100 vehicle search requires per day.
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***Search*** and ***Logout*** button
- Show the same input form as on the **Public Search** page along with a VIN input field
- If User.*User_Type* == "Inventory_clerk":
  - ❖ Upon:
  - ❖ When user input and clicks ***Search*** button:
    - ➢ If all the input fields are empty, show warning message;
    - ➢ If there are vehicles that exit on Vehicle table and are not on Sell_Transaction table, show results, sorted by VIN;
    - ➢ If no vehicles meet the search criteria on Vehicle table, show error message "Sorry, it looks like we don't have that in stock!"
- If User.*User_Type* == "Salesman":
  - ❖ Upon:
  - ❖ When user input and clicks ***Search*** button:
    - ➢ The same will happen as on the **Public Search** page;
- If User.*User_Type* == "Manager":
  - ❖ Show number of items on Vehicle table where Repair.*Status* == "Pending" as "Number of cars repair pending"
  - ❖ Show number of items on Vehicle table where Repair.*Status* == "In progress" as "Number of cars repair in progress"
  - ❖ Show number of items on Vehicle table where Repair.*Status* == "Complete" and not in Sell_Transaction table as "Number of available vehicles for purchase"
  - ❖ Upon:
  - ❖ When user input and clicks ***Search*** button:
    - ➢ If all the input fields are empty, show warning message;

- ➢ If there are vehicles that exit on Vehicle table, show results, sorted by VIN;
- ➢ If no vehicles meet the search criteria on Vehicle table, show error message "Sorry, it looks like we don't have that in stock!"

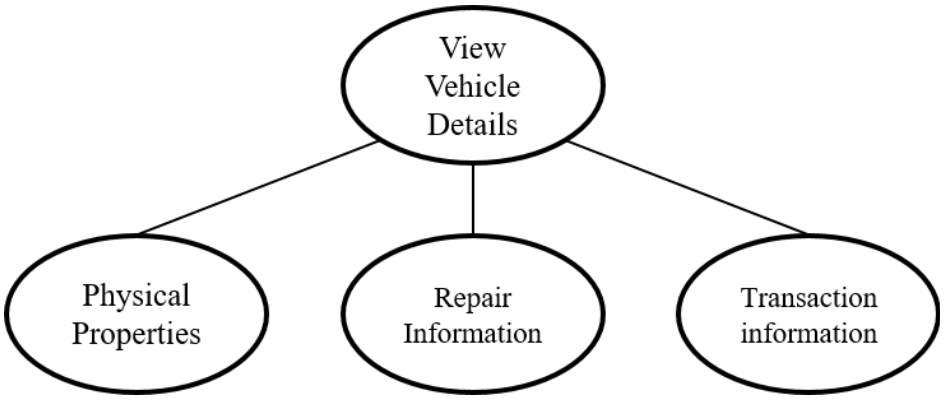
❖ Have additional filter by sold vehicles, unsold vehicles, or all vehicles:

- ➢ If user choose "sold vehicles", filter the results so only show vehicles exist on Sell_Transaction table.
- ➢ If user choose "unsold vehicles", filter the results so not show vehicles exist on Sell_Transaction table.
- ➢ If user choose "all vehicles", do not filter, show all the results.

- if User.*User_Type* == "Owner":
  - ❖ Have all the functionality as other users described above
- When user selects a search result: Go to **View Vehicle Detail** Form
- When ***Logout*** button is clicked, go to **Public Search** page.

View Vehicle Detail Form

Task Decomposition:



Lock Types: Read-only on Vehicle, User, Customer, Repair, Buy_transaction, Sell_transaction and Recall Table.
Number of Locks: 7
Enabling Conditions:
Physical Properties subtask is enabled by clicking ***View Vehicle Detail*** button on selecting a single entry on the public search or employee search result list
Repair information subtask is enabled by Users Type as "Manager" by clicking ***View Vehicle Detail*** button;
Repair information subtask, along with ***Add Repair*** and ***Edit Repair*** button, is enabled by Users Type as "Inventory Clerk" or "Owner" by clicking ***View Vehicle Detail*** button;
Transaction information subtask is enabled by Users Type as "Manager" or "Owner" by clicking ***View Vehicle Detail*** button;
Sell Vehicle button is enabled by Users Type as "Salesperson" or "Owner" by clicking ***View Vehicle Detail*** button;

Frequency: About 500 view vehicle detail per day; subtasks have different frequency;
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is needed; Decomposed into three subtasks.

Abstract Code

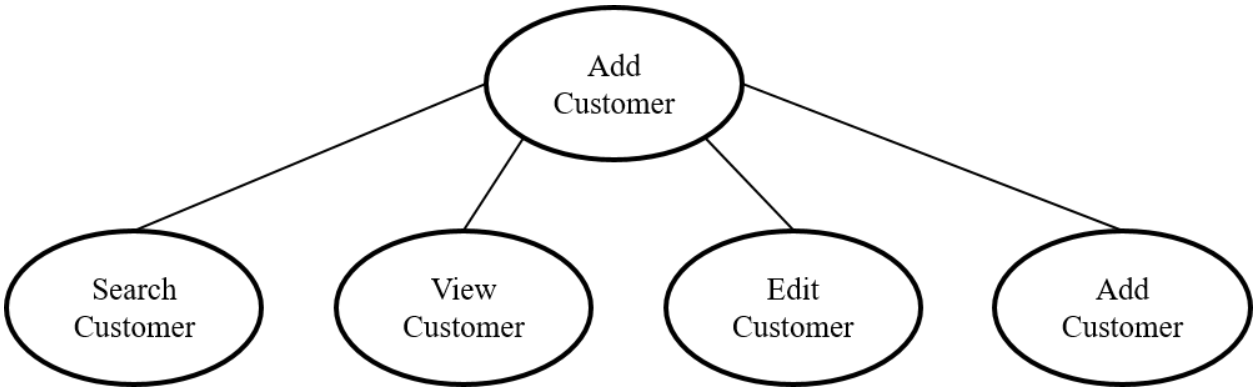
- Show ***View Vehicle Detail*** on public search or employee search result list
- Upon:
  - ❖ Click ***View Vehicle Detail*** button:

- ➢ View physical properties, repair information, and/or transaction information as stated in Enabling Conditions;
- ➢ Show ***Repair Description*** pop-out button;
- ➢ Click ***Repair Description*** pop-out button: View Repair.*description*;

❖ Click ***Done*** button - Go back to search result list.

Add Customer Form

Task Decomposition:



Lock Types: Read and write on Customer Table; Read-only on User Table
Number of Locks: 2
Enabling Conditions: Enabled by clicking ***Add Customer*** button on public search or employee search result list
Frequency: About 20 add customer per day; subtasks have different frequency;
Consistency (ACID): Search runs prior to vier, edit or add customer
Subtasks: Mother Task is needed. Decomposed into four subtasks.

Abstract Code:

- Show ***Add Customer*** button upon clicking ***Sell Vehicle*** button by salesperson or ***Add Vehicle*** button by inventory clerk.
- Upon:
  - ❖ Click ***Add Customer*** button
    - ➢ Run Search Customer subtask, user input customer's driver_lisence_number (as a person) or Tax_identification_number (as a business);
    - ➢ If input in Customer.driver_lisence_number or Customer.Tax_identification_number:
      - ■ View Customer; Show ***Edit Customer*** button;
      - ■ Upon Click ***Edit Customer***:
        - User enters customer information
        - Click ***Save*** button: Update Customer Table; Return to View Customer;
        - Click ***Cancel*** button: Return to View Customer;
    - ➢ If the user's input does not match any items in Customer.driver_lisence_number nor Customer.Tax_identification_number:
      - ■ Show message 'new customer!'
      - ■ Show ***Add New Customer*** button;
      - ■ Upon clicking ***Add New Customer*** button:
        - User select either Person or Business;
        - If Person is selected:

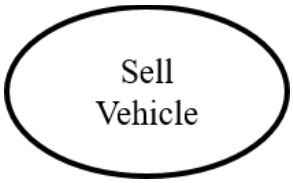
- User enter address, email (optional), phone_number, first_name, last_name;
- Click ***Save*** button: Write to Customer Table; Return to Add Customer; View customer;
- Click ***Cancel*** button: Return to **Add New Customer**;

- If Business is selected:
  - User enter address, email, phone_number, business_name, primary_contact_name, primary_contact_title;
  - Click ***Save*** button: Write to Customer Table; Return to Add Customer; View customer;
  - Click ***Cancel*** button: Return to Add New Customer;

❖ Click ***Done*** button – Go back to **Add New Vehicle** Form (inventory clerk) or **Sell Vehicle** Form (salesperson)

Sale Order Form

Task Decomposition:



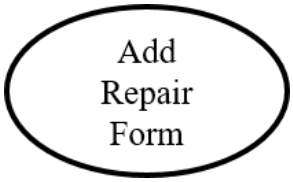
Lock Types: Read and write on Sell_Transaction Table; Read-only on Vehicle, User and Customer Table;
Number of Locks: 4
Enabling Conditions: Enabled by clicking ***Sell Vehicle*** button on vehicle detail form
Frequency: About 10 add sell vehicle per day
Consistency (ACID): Consistency is not critical; order is not critical
Subtasks: No decomposed is needed.

Abstract Code

- Show Sell Vehicle on vehicle detail screen.
- Upon:
  - ❖ User click ***Sell Vehicle*** button;
    - ➢ Show ***Add Customer*** button;
    - ➢ Upon Add Customer finished, show Sale Order Form;
      - ➢ User enters *sale_date*;
      - ➢ Click Done button: Write into Sell_Transaction Table; Display Vehicle detail form;
      - ➢ Click ***Cancel*** Button: Return to Vehicle Detail Form;

Add Repair Form

Task Decomposition:



Lock Types: Read and Write to the Repair Table and Recall Table. Read and Write exclusives locks needed on Repair table, Read-only on Recall Table.
Number of Locks: 3
Enabling Conditions: Enabled by Users Type as "Inventory_clerk" or "Owner" by clicking ***Add Repair Form*** button in **Repair/Recall Repository**
Frequency: About 50 adding repairs/recalls requires in total
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***Add Repair Form***, ***Edit Repair Form, View Repair From, Delete Repair Form*** buttons in **Repair/Recall Repository**
- User clicks ***Add Repair Form*** button and the system displays a new repair form table with ***Submit*** and ***Cancel*** button
- User enters *VIN, Repair_Vendor_Name, Repair_Vendor_Address, Repair_Vendor_Phone_Number, Repair_Description, Repair_Start_Date, Repair_Cost, NHTSA_Recall_Campagin_Number*
- User selects one of "*Pending*", "*In Progress*" and "*Complete*" in *Repair_Status*
- Upon:
  - ❖ Click ***Submit*** button;
    - ➢ If *NHTSA_Recall_Campagin_Number* is not Null and is not in Recall.*NHTSA_Recall_Campagin_Number*:
      - ■ Show Error Message and Return to **Repair/Recall Repository**
    - ➢ Else if all items except *NHTSA_Recall_Campagin_Number* are not Null:
      - ■ Write to Repair Table and go back to **Repair/Recall Repository**
    - ➢ Else:
      - ■ Show Error Message and Return to **Repair/Recall Repository**
  - ❖ Click ***Cancel*** button - Go back to **Repair/Recall Repository**

Edit Repair Form

Task Decomposition:

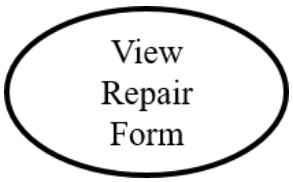
Lock Types: Read and Write to the Repair Table and Recall Table. Read and Write exclusives locks needed on Repair table, Read-only on Recall Table.
Number of Locks: 3
Enabling Conditions: Enabled by Users Type as "Inventory_clerk" or "Owner" by clicking ***Edit Repair Form*** button in **Repair/Recall Repository**
Frequency: About 100 editing requires in total
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***Add Repair Form***, ***Edit Repair Form, View Repair From, Delete Repair Form*** in **Repair/Recall Repository**
- User clicks ***Edit Repair Form*** button and show ***Submit*** and ***Cancel*** button
- User edits Repair Form Items
- User selects one of "*Pending*", "*In Progress*" and "*Complete*" in *Repair_Status*
- Upon:
  - ❖ Click ***Submit*** button;
    - ➢ If *NHTSA_Recall_Campagin_Number* is not Null and is not in Recall.*NHTSA_Recall_Campagin_Number*:
      - ■ Show Error Message and Return to **Repair/Recall Repository**
    - ➢ Else if all items except *NHTSA_Recall_Campagin_Number* are not Null:
      - ■ Write to Recall Table and go back to **Repair/Recall Repository**
    - ➢ Else:
      - ■ Show Error Message and Return to **Repair/Recall Repository**
  - ❖ Click ***Cancel*** button - Go back to **Repair/Recall Repository**

View Repair Form

Task Decomposition:



Lock Types: Read-only to the Repair Table. Read exclusives locks needed on Repair table.
Number of Locks: 1
Enabling Conditions: Enabled by Users Type as "Inventory_clerk" or "Owner" by clicking ***View Repair Form*** button in **Repair/Recall Repository**
Frequency: About 200 viewing requires in total
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***Add Repair Form***, ***Edit Repair Form, View Repair From, Delete Repair Form*** in **Repair/Recall Repository**
- User clicks ***View Repair Form*** button and show ***Search*** and ***Done*** button
- User inputs the *VIN* number in the search bar
- Upon:
  - ❖ Click ***Search*** button:
    - ➢ If *VIN* is Null or is not in Recall.*VIN*:
      - ■ Show Error Message and Return to **Repair/Recall Repository**
    - ➢ Else:
      - ■ Find and display all repair forms for this *VIN* ordered by the descending *Start_Date*
  - ❖ Click ***Done*** button - Go back to **Repair/Recall Repository**

Delete Repair Form

Task Decomposition:

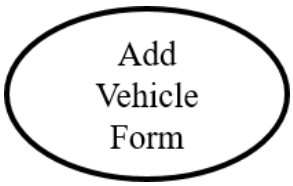
Lock Types: Read and Write to the Repair Table. Read and Write exclusives locks needed on Repair table
Number of Locks: 2
Enabling Conditions: Enabled by Users Type as "Inventory_clerk" or "Owner" by clicking ***Delete Repair Form*** button in **Repair/Recall Repository**
Frequency: About 20 deleting requires in total
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***Add Repair Form***, ***Edit Repair Form, View Repair From, Delete Repair Form*** in **Repair/Recall Repository**
- User clicks ***Delete Repair Form*** button and show search bar with ***Search*** and ***Done*** button
- User inputs the *VIN* number in the search bar
- Upon:
  - ❖ Click ***Search*** button:
    - ➢ If *VIN* is Null or is not in Repair.*VIN*:
      - ■ Show Error Message and Return to **Repair/Recall Repository**
    - ➢ Else:
      - ■ Find and display all repair forms for this *VIN* ordered by the descending *Start_Date*.
      - ■ Select and Delete one or more repair form for this *VIN*
  - ❖ Click ***Done*** button - Go back to **Repair/Recall Repository**

Add Vehicle Form

Task Decomposition:



Lock Types: Read and Write to the Vehicle Table
Number of Locks: 2
Enabling Conditions: Enabled by Users Type as "Inventory_clerk" or "Owner" by clicking ***Add Vehicle Form*** button in **Vehicle Category**
Frequency: About 100 adding requires in total
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

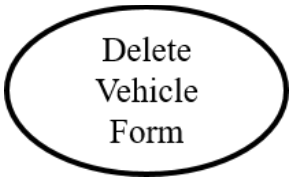
Abstract Code:

- Show ***Add Vehicle Form, Edit Vehicle Form, View Vehicle Form, Delete Vehicle Form*** button in **Vehicle Category**

- User clicks ***Add Vehicle Form*** button and the system displays a new vehicle form table with ***Submit*** and ***Cancel*** button
- User enters *VIN, Vehicle_Type, Vehicle_Manufacturer, Model_Name, Model_Year, Color, Mileage, Vehicle_Description*
- Upon:
    - ❖ Click ***Submit*** button;
        - ➢ If *VIN* is in Vehicle.*VIN*::
            - ■ Show Error Message and Return to **Vehicle Category**
        - ➢ Else if all items except *Vehicle_Description* are not Null:
            - ■ Write to Vehicle Table and go back to **Vehicle Category**
        - ➢ Else:
            - ■ Show Error Message and Return to **Vehicle Category**
    - ❖ Click ***Cancel*** button - Go back to **Vehicle Category**

Delete Vehicle Form

Task Decomposition:



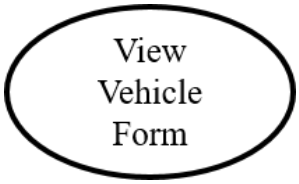
Lock Types: Read and Write to the Vehicle Table, Read-only on Repair Table
Number of Locks: 3
Enabling Conditions: Enabled by Users Type as "Inventory_clerk" or "Owner" by clicking ***Delete Vehicle Form*** button in **Vehicle Category**
Frequency: About 50 deleting requires in total
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***Add Vehicle Form, Edit Vehicle Form, View Vehicle Form, Delete Vehicle Form*** button in **Vehicle Category**
- User clicks ***Delete Vehicle Form*** button and the system displays a search bar with ***Search*** and ***Done*** button
- User enters *VIN* in the search bar
- Upon:
    - ❖ Click ***Submit*** button;
        - ➢ If If *VIN* is Null or is not in Vehicle.*VIN*:
            - ■ Show Error Message and Return to **Vehicle Category**
        - ➢ Else if there is Repair.*VIN* same as *VIN*:
            - ■ Show Error Message and Return to **Vehicle Category**
        - ➢ Else:
            - ■ Display the information of this *VIN* from Vehicle Table
            - ■ Delete the *VIN* information and Click ***Done*** button: Go back to **Vehicle Category**
    - ❖ Click ***Done*** button - Go back to **Vehicle Category**

View Vehicle Form

Task Decomposition:



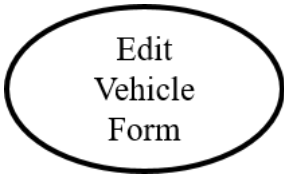
Lock Types: Read and Write to the Vehicle Table
Number of Locks: 2
Enabling Conditions: Enabled by Users Type as "Inventory_clerk" or "Owner" by clicking ***View Vehicle Form*** button in **Vehicle Category**
Frequency: About 100 viewing requires in total
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***Add Vehicle Form, Edit Vehicle Form, View Vehicle Form, Delete Vehicle Form*** button in **Vehicle Category**
- User clicks ***View Vehicle Form*** button and the system displays a new empty vehicle form with ***Search*** and ***Done*** button
- User enters *VIN* in the search bar
- Upon:
    - ❖ Click ***Submit*** button;
        - ➢ If *VIN* is Null or is not in Vehicle.*VIN*:
            - ■ Show Error Message and Return to **Vehicle Category**
        - ➢ Else:
            - ■ Display the information of this *VIN* from Vehicle Table
            - ■ Click ***Done*** button: Go back to **Vehicle Category**
    - ❖ Click ***Done*** button - Go back to **Vehicle Category**

Edit Vehicle Form

Task Decomposition:



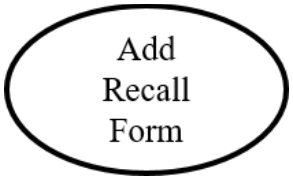
Lock Types: Read and Write to the Vehicle Table
Number of Locks: 2
Enabling Conditions: Enabled by Users Type as "Inventory_clerk" or "Owner" by clicking ***Edit Vehicle Form*** button in **Vehicle Category**
Frequency: About 100 editing requires in total
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***Add Vehicle Form, Edit Vehicle Form, View Vehicle Form, Delete Vehicle Form*** button in **Vehicle Category**
- User clicks ***Edit Vehicle Form*** button and the system displays a search bar with ***Search*** and ***Done*** button
- User enters *VIN* in the search bar
- Upon:
    - ❖ Click ***Submit*** button;
        - ➢ If *VIN* is Null:
            - ■ Show Error Message and Return to **Vehicle Category**
        - ➢ Else:
            - ■ Display the information of this *VIN* from Vehicle Table and User edits the information
            - ■ Click ***Done*** button:
                - If all of the items except *Vehicle_Description* are not null and *Vehicle_Description* can be Null or Not Null:
                    - ◆ Write to the Vehicle table and Go back to **Vehicle Category**
                - Else:
                    - ◆ Show Error Message and Return to **Vehicle Category**
    - ❖ Click ***Done*** button - Go back to **Vehicle Category**

Add Recall Form

Task Decomposition:



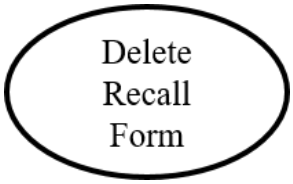
Lock Types: Read and Write to the Recall Table
Number of Locks: 2
Enabling Conditions: Enabled by Users Type as “Inventory_clerk” or “Owner” by clicking ***Add Vehicle Form*** button in **Repair/Recall Repository**
Frequency: About 5 adding requires in total
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***Add Recall Form, Edit Recall Form, View Recall Form, Delete Recall Form*** button in **Repair/Recall Repository**
- User clicks ***Add Recall Form*** button and the system displays a new recall form with ***Submit*** and ***Cancel*** button
- User enters *NHTSA_Recall_Campagin_Number, Recall_Description, Recall_Manufacture* information
- Upon:
    - ❖ Click ***Submit*** button;
        - ➢ If *NHTSA_Recall_Campagin_Number* is in Recall.*NHTSA_Recall_Campagin_Number*:
            - ■ Show Error Message and Return to **Repair/Recall Repository**
        - ➢ Else if all items are not Null:
            - ■ Write to Recall Table and go back to **Repair/Recall Repository**
        - ➢ Else:
            - ■ Show Error Message and Return to **Repair/Recall Repository**
    - ❖ Click ***Cancel*** button - Go back to **Repair/Recall Repository**

Delete Recall Form

Task Decomposition:



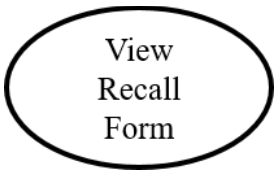
Lock Types: Read and Write to the Recall Table, Read-only on Repair Table
Number of Locks: 3
Enabling Conditions: Enabled by Users Type as "Inventory_clerk" or "Owner" by clicking ***Delete Recall Form*** button in **Repair/Recall Repository**
Frequency: About 10 deleting requires in total
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***Add Recall Form, Edit Recall Form, View Recall Form, Delete Recall Form*** button in **Repair/Recall Repository**
- User clicks ***Delete Vehicle Form*** button and the system displays a search bar with ***Search*** and ***Done*** button
- User enters *NHTSA_Recall_Campagin_Number* in the search bar
- Upon:
    - ❖ Click ***Submit*** button;
        - ➢ If *NHTSA_Recall_Campagin_Number* is Null or is not in Recall.*NHTSA_Recall_Campagin_Number*:
            - ■ Show Error Message and Return to **Repair/Recall Repository**
        - ➢ Else if there is Repair.*NHTSA_Recall_Campagin_Number* same as *NHTSA_Recall_Campagin_Number*:
            - ■ Show Error Message and Return to **Repair/Recall Repository**
        - ➢ Else:
            - ■ Display the information of this *NHTSA_Recall_Campagin_Number* from Recall Table
            - ■ Select the *NHTSA_Recall_Campagin_Number* information and delete. Click ***Done*** button: Go back to **Repair/Recall Repository**
    - ❖ Click ***Done*** button - Go back to **Repair/Recall Repository**

View Recall Form

Task Decomposition:



Lock Types: Read and Write to the Recall Table
Number of Locks: 2

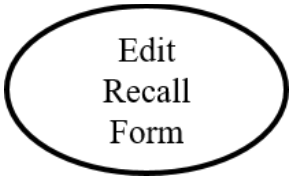
Enabling Conditions: Enabled by Users Type as "Inventory_clerk" or "Owner" by clicking ***View Recall Form*** button in **Vehicle Category**
Frequency: About 50 viewing requires in total
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***Add Recall Form, Edit Recall Form, View Recall Form, Delete Recall Form*** button in **Repair/Recall Repository**
- User clicks ***View Recall Form*** button and the system displays a search bar with ***Search*** and ***Done*** button
- User enters *NHTSA_Recall_Campagin_Number* in the search bar
- Upon:
    - ❖ Click ***Submit*** button;
        - ➢ If If *NHTSA_Recall_Campagin_Number* is Null or is not in Recall.*NHTSA_Recall_Campagin_Number*:
            - ■ Show Error Message and Return to **Repair/Recall Repository**
        - ➢ Else:
            - ■ Display the information of this *NHTSA_Recall_Campagin_Number* from Recall Table
            - ■ Click ***Done*** button: Go back to **Repair/Recall Repository**
    - ❖ Click ***Done*** button - Go back to **Repair/Recall Repository**

Edit Recall Form

Task Decomposition:



Lock Types: Read and Write to the Recall Table
Number of Locks: 2
Enabling Conditions: Enabled by Users Type as "Inventory_clerk" or "Owner" by clicking ***Edit Recall Form*** button in **Vehicle Category**
Frequency: About 100 editing requires in total
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***Add Vehicle Form, Edit Vehicle Form, View Vehicle Form, Delete Vehicle Form*** button in **Repair/Recall Repository**
- User clicks ***Edit Vehicle Form*** button and the system displays a search bar with ***Search*** and ***Done*** button
- User enters *NHTSA_Recall_Campagin_Number* in the search bar
- Upon:
    - ❖ Click ***Submit*** button;
        - ➢ If *NHTSA_Recall_Campagin_Number* is Null or *NHTSA_Recall_Campagin_Number* is not in Recall.*NHTSA_Recall_Campagin_Number*:
            - ■ Show Error Message and Return to **Repair/Recall Repository**
        - ➢ Else:

- Display the information of this *NHTSA_Recall_Campagin_Number* from Recall Table and User edits the information
- Click ***Done*** button:
  - If all of the items are not Null:
    - Write to the Recall table and Go back to **Repair/Recall Repository**
  - Else:
    - Show Error Message and Return to **Repair/Recall Repository**

❖ Click ***Done*** button - Go back to **Repair/Recall Repository**

View Seller History Report

Task Decomposition:

Lock Types: Read-only on the Sell_transcation Table. Read-only on Customer Table. Read-only on Repair Table.
Number of Locks: 3
Enabling Conditions: Enabled by Users Type as "Manager" or "Owner" by clicking ***View Sale History Report*** button in **Initial Search** Page.
Frequency: About 10 views per day.
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***View Sale History Report*** button in **Initial Search** Page.
- Show ***Done*** button.
- For each *Driver_ license_ number* and *Tax_ identification_ number* in Customer table:
  - ❖ Calculate the total number of vehicles sold to Burdell's in the Sell_transcation Table.
  - ❖ Calculate the average price for the vehicles each customer has sold to Burdell's in the Sell_transcation Table.
  - ❖ Calculate the average number of repairs per vehicle in the Repair Table.
- Find and display the name of the customers (either first or last name or the company name) from the Customer table in descending order of the total number of vehicles the respective customer have sold to Burdell's.
- Display the average price for the vehicles and the average number of repairs per vehicle for each customer.
- Highlight the customers that show an average of five or more repairs per vehicle sold to Burdell's with a red background.
- Upon:
  - ❖ Click ***Done*** button - Go back to **Initial Search** Page.

View Inventory Age Report

Task Decomposition:

Lock Types: Read-only on the Sell_transcation Table. Read-only on Vehicle Table.
Number of Locks: 2
Enabling Conditions: Enabled by Users Type as "Manager" or "Owner" by clicking ***View Inventory Age Report*** button in **Initial Search** Page.
Frequency: About 10 views per day.
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***View Inventory Age Report*** button in **Initial Search** Page.
- Show ***Done*** button.
- For each type of vehicle (identified with *Vehicle_type* in Vehicle table):
  - Calculate the minimum, average, and maximum age (decided by *Model_year* in Vehicle table) of unsold vehicles in inventory, in days.
- Find and display *Vehicle_type* in Vehicle table in alphabetical order.
- Display the minimum, average, and maximum age of unsold vehicles in inventory, in days, for each *Vehicle_type*. If a vehicle type has no unsold units, the report should display "N/A" for that *Vehicle_type*.
- Upon:
  - Click ***Done*** button - Go back to **Initial Search** Page.

View Average Time in Inventory Report

Task Decomposition:

Lock Types: Read-only on the Sell_transcation Table. Read-only on Buy_transcation Table. Read-only on Vehicle Table.
Number of Locks: 3
Enabling Conditions: Enabled by Users Type as "Manager" or "Owner" by clicking ***View Average Time in Inventory Report*** button in **Initial Search** Page**.**
Frequency: About 10 views per day.
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***View Inventory Age Report*** button in **Initial Search** Page**.**
- Show ***Done*** button.
- For each type of vehicle (identified with *Vehicle_type* in Vehicle table):

- ❖ Calculate the average time in the inventory of unsold vehicles, in days. Using *Purchase_date* from the Buy_transcation Table and *Sale_date* from the Sell_transcation Table.
- Find and display all *Vehicle_type* in Vehicle table in alphabetical order.
- Display the average time in the inventory of unsold vehicles, in days, for each *Vehicle_type*. If a *Vehicle_type* has no unsold units, the report should display "N/A" for that vehicle type.
- Upon:
  - ❖ Click ***Done*** button - Go back to in **Initial Search** Page**.**

View Price per Condition Report

Task Decomposition:

Lock Types: Read-only on the Buy_transcation Table. Read-only on Vehicle Table.
Number of Locks: 2
Enabling Conditions: Enabled by Users Type as "Manager" or "Owner" by clicking ***View Inventory Age Report*** button in **Initial Search** Page**.**
Frequency: About 10 views per day.
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***View Price per Condition Report*** button in **Initial Search** Page.
- Show ***Done*** button.
- For each *Vehicle_type* in the Vehicle table and for each P*urchase_condition* in the Buy_transcation Table:
  - ❖ Calculate the average price of vehicles purchased. If a vehicle type or condition has never been purchased, the report should display "$0" for that result.
- Display the average price of vehicles per *vehicle_type* and per *purchase_condition* in a pivot table.
- Upon:
  - ❖ Click ***Done*** button - Go back to **Initial Search** Page.

View Repair Statistics Report

Task Decomposition:

Lock Types: Read-only on the Repair Table. Read-only on the Vehicle Table.
Number of Locks: 2
Enabling Conditions: Enabled by Users Type as "Manager" or "Owner" by clicking ***View Inventory Age Report*** button in **Initial Search** Page.

Frequency: About 10 views per day.
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- Show ***View Repair Statistics Report*** button in **Initial Search** Page.
- Show ***Done*** button.
- For each V*endor_name* in Repair table:
  - ❖ Calculate the number of repairs by that vendor.
  - ❖ Calculate the total cost spent on completed repairs from that vendor using *Repair_cost* in the Repair table.
  - ❖ Calculate the average number of repairs per vehicle (identified by *VIN* from the Vehicle Table) from that vendor using *Repair_cost* in the Repair table.
- Calculate the average length of repair time (in days) from that vendor using *Start_date* and *End_date* in the Repair table
- Find and display all V*endor_name* in Repair table in alphabetical order.
- Display the number of repairs, the total cost spent on completed repairs, the average number of repairs per vehicle, and the average repair time (in days) for each vendor.
- Upon:
  - ❖ Click ***Done*** button - Go back to **Initial Search** Page.

View Monthly Sales Report

Task Decomposition:

Lock Types: Read-only on the Sell_transcation Table. Read-only on Buy_transcation Table. Read-only on Repair Table. Read-only on User Table. .
Number of Locks: 4
Enabling Conditions: Enabled by Users Type as "Manager" or "Owner" by clicking ***View Monthly Sales Report*** button in **Initial Search** Page.
Frequency: About 50 views per day.
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is to show a summary page, which lists for all sales transactions, by year and month. Subtask is for each year/month show a drilldown report for that year and month.

Abstract Code:

- Show ***View Monthly Sales Report*** button in **Initial Search** Page.
- Show ***Done*** button in both the mother task and subtask.

- Find and display *Sale_date* in the Sell_transcation Table in descending order.
- For each calendar year and month in *Sale_date*:
    - ❖ Calculate and display the number of vehicles sold (i.e. the number of items) from the Sell_transcation Table. If a year or month have 0 items in Sell_transcation, do not show that year or month.
    - ❖ Calculate and display the total sales income using *Sale_price* from the Sell_transcation Table.
    - ❖ Calculate and display the net sales income using *Sale_price* from the Sell_transcation Table, *Purchase_price* from the Buy_transcation Table, and *Repair_cost* from the Repair Table
- For each year/month result, create a clickable link to a drilldown report, find all items from the Salesperson Table as a subitem of the User Table.
- For each item from the Salesperson Table:
    - ❖ Calculate the number of vehicles sold (i.e. the number of items) from the Sell_transcation Table.
    - ❖ Calculate the total sales using *Sale_price* from the Sell_transcation Table.
- Find and display all items from the Salesperson Table, sorted by total vehicles in descending order, followed by total sales in descending order. Display these items' *Logged_in_first_name* and *Logged_in_last_name* from the User Table.
- Upon:
    - ❖ Click ***Done*** button - Go back to **Initial Search** Page**.**
    - ❖ Click each individual ***year*** *or* ***month*** link - Go to the drill down report for that respective year or month.