

|  |
| --- |
| **CS 6400 Database Systems Concepts and Design**  Team 22 - Phase 2 Report |
| Jinjun Liu (jliu788) |
| Zijian Xie (zxie86) |
| Hui Xia (hxia40) |
| Chen Zhang (czhang613) |

[**Tables**](#_fnwoy0m3pj6c) **3**

[User Table](#_b99602fgp9b4) 3

[Repair Table](#_xw93w0khh7v5) 3

[Recall Table](#_pzfpjbmvhqw7) 3

[Vehicle Table](#_hvu0hwpgyasb) 4

[Sell\_Transaction Table](#_9pcze3uz2a5s) 4

[Buy\_Transaction Table](#_alz42o1qdch9) 5

[Customer Table](#_umfw1okilpws) 5

[**Business Logic Constraints**](#_wl01onp4abac) **5**

[**Assumptions**](#_1c176b8vpqrr) **6**

[**Task Decomposition and Abstract Code**](#_jvpmhpdcumvc) **6**

[Public Search](#_9xo2km4tc4go) 6

[Login](#_i84d6jolmf4g) 7

[Initial Search](#_1ynwyft2gvb9) 8

[View Vehicle Detail Form](#_uvdd7rmv400k) 9

[Add Customer Form](#_e3mkryipokls) 10

[Sale Order Form](#_f3v2ywd8eqqw) 11

[Add Repair Form](#_qt623j6ctsrc) 11

[Edit Repair Form](#_9d0fbrgiqwrv) 12

[View Repair Form](#_82jdo2x54bq) 13

[Delete Repair Form](#_jfk32hv22v22) 13

[Add Vehicle Form](#_169m4wgv18dy) 14

[Delete Vehicle Form](#_764cb0l8papj) 15

[View Vehicle Form](#_4jgovm4ijhnu) 15

[Edit Vehicle Form](#_yvaw918fmzdx) 16

[Add Recall Form](#_axfy1kpxll2q) 17

[Delete Recall Form](#_jwtv6vo646zo) 17

[View Recall Form](#_tdbkhlkh42vy) 18

[Edit Recall Form](#_pwc2z5d4g9mx) 19

[View Seller History Report](#_uddwxh7ivtq5) 20

[View Inventory Age Report](#_1aa60joes35v) 20

[View Average Time in Inventory Report](#_1j8khpdmzlw6) 21

[View Price per Condition Report](#_tsfb1temc84w) 22

[View Repair Statistics Report](#_jjhh69r2m4d0) 22

[View Monthly Sales Report](#_4pn1t7phsb4h) 23

### Tables

#### 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 |

#### 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 |

#### Recall Table

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Nullable** |
| **NHTSA\_Recall\_Campagin\_Number** | String | Not |
| **Recall\_Description** | String | Not |
| **Recall\_Manufacture** | String | Not |

#### 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 |

#### Sell\_Transaction Table

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Nullable** |
| **VIN** | String | Not |
| **Sale\_date** | Timestamp | Not |
| **Sale\_price** | Decimal | Not |
| **Salesperson\_username** | String | Not |

#### 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 |

#### 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 |

### Business Logic Constraints

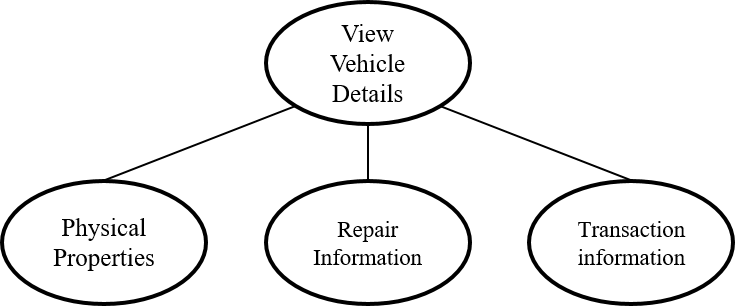
* Only the Owner and Inventory Clerk have the access to read and write the Recall/Repair and Vehicle Tables.
* The *Model\_Year* in the Vehicle Table cannot exceed current year plus one and must be 4-digits form.
* Inventory clerk and owner have links to Add Repair and Edit repair button;
* If Repair.Status 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;

### 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%;

##### 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

// assume $vin of current vehicle is managed by application

// assume $username of current user is managed by application

|  |
| --- |
| SELECT *vin, vehicle\_mileage, vehicle\_description, model\_name, model\_year, manufacturer\_name*  FROM *`*Vehicle*`* WHERE Vehicle.vin = '$vin' |

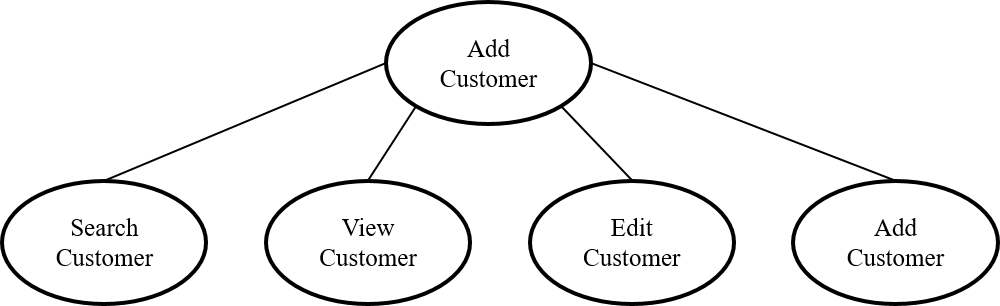
SELECT Repair.vin, vehicle\_mileage, vehicle\_description, model\_name, model\_year, manufacturer\_name, start\_date, end\_date, repair\_status, repair\_description, repair\_cost, inventory\_clerk\_permission

FROM Vehicle JOIN Repair on Vehicle.vin = Repair.vin WHERE Vehicle.vin = '$vin'

* 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;

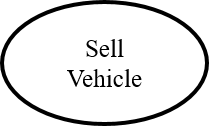
Click ***Save*** button: Write to Customer Table; Return to Add Customer; View customer;

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.

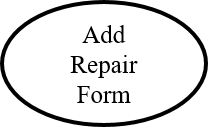
**Abstract Code**

* 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:**

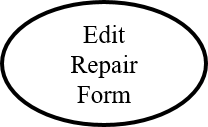
* Show ***Add Repair Form***, ***Edit Repair Form, View Repair From, Delete Repair Form*** buttonsin **Repair/Recall Repository**
* User clicks ***Add Repair Form*** button and the system displays a new repair form table with ***Submit*** and ***Cancel*** button
* User enters *VIN, Repair\_Vendor\_Name, 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**

|  |
| --- |
| //Use Repair;  INSERT INTO *`*Repair*`*  VALUES(  '$enteredVIN', '$enteredRepair\_Vendor\_Name', '$enteredRepair\_Vendor\_Address',  '$enteredRepair\_Vendor\_Phone\_Number', '$enteredRepair\_Description', '$enteredRepair\_Start\_Date',  '$enteredRepair\_Cost', '$enteredNHTSA\_Recall\_Campagin\_Number', '$enteredRepair\_Status'  ); |

* + 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*

|  |
| --- |
| SELECT *VIN, Repair\_Vendor\_Name, Repair\_Vendor\_Address, Repair\_Vendor\_Phone\_Number,*  *Repair\_Description, Repair\_Start\_Date, Repair\_Cost, NHTSA\_Recall\_Campagin\_Number*  FROM *`*Repair*`* WHERE Repair.VIN = '$enteredVIN' |

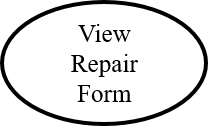
* Upon:
* Click ***Submit*** button;
  + If *NHTSA\_Recall\_Campagin\_Number* is not Null and is not in Recall.*NHTSA\_Recall\_Campagin\_Number*:
    - Show Error Message and Return to **Repair/Recall Repository**
  + Else if all items except *NHTSA\_Recall\_Campagin\_Number* are not Null:
    - Write to Recall Table and go back to **Repair/Recall Repository**

|  |
| --- |
| //Use Repair;  UPDATE *`*Repair*`*  SET VIN = '$enteredVIN',  Repair\_Vendor\_Name = '$enteredRepair\_Vendor\_Name',  Repair\_Vendor\_Address = 'enteredRepair\_Vendor\_Address',  Repair\_Vendor\_Phone\_Number = 'enteredRepair\_Vendor\_Phone\_Number',  Repair\_Description = 'enteredRepair\_Description',  Repair\_Start\_Date = 'enteredRepair\_Start\_Date',  Repair\_Cost = 'enteredRepair\_Cost',  NHTSA\_Recall\_Campagin\_Number = 'enteredNHTSA\_Recall\_Campagin\_Number',  Repair\_Status = 'enteredRepair\_Status'; |

* + 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:**

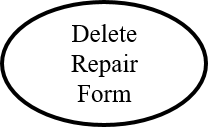
* Show ***Add Repair Form***, ***Edit Repair Form, View Repair From, Delete Repair Form*** in **Repair/Recall Repository**
* User clicks ***View Repair Form*** button and show ***Search*** and ***Done*** button
* User inputs the *VIN* number in the search bar
* Upon:
* Click ***Search*** button:
  + 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*

|  |
| --- |
| SELECT  *VIN, Repair\_Vendor\_Name, Repair\_Vendor\_Address, Repair\_Vendor\_Phone\_Number,*  *Repair\_Description, Repair\_Start\_Date, Repair\_Cost, NHTSA\_Recall\_Campagin\_Number*  FROM *`*Repair*`* WHERE Repair.VIN = '$enteredVIN'  ORDER BY *Repair\_Start\_Date* DESC; |

* 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:**

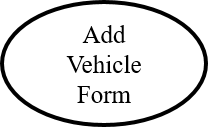
* Show ***Add Repair Form***, ***Edit Repair Form, View Repair From, Delete Repair Form*** in **Repair/Recall Repository**
* User clicks ***Delete Repair Form*** button and show search bar with ***Search*** and ***Done*** button
* User inputs the *VIN* number in the search bar
* Upon:
* Click ***Search*** 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*.

|  |
| --- |
| DELETE FROM *`*Repair*`* WHERE Repair.VIN = '$enteredVIN' AND Repair.Start\_Date = '$enteredStart\_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:**

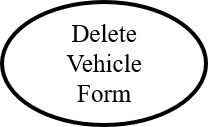
* Show ***Add Vehicle Form, Edit Vehicle Form, View Vehicle Form, Delete Vehicle Form*** button in **Vehicle Category**
* User clicks ***Add Vehicle Form*** button and the system displays a new vehicle form table with ***Submit*** and ***Cancel*** button
* User enters *VIN, Vehicle\_Type, Vehicle\_Manufacturer, 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**

|  |
| --- |
| INSERT INTO *`*Vehicle*`*  VALUES(  '$enteredVIN', '$enteredVehicle\_Type', '$enteredVehicle\_Manufacturer', '$enteredModel\_Name',  '$enteredModel\_Year', '$enteredVehicle\_Color', '$enteredMileage', '$enteredSales\_Price',  '$enteredVehicle\_Description'  ); |

* + 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:**

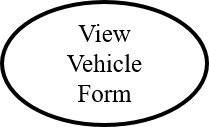
* Show ***Add Vehicle Form, Edit Vehicle Form, View Vehicle Form, Delete Vehicle Form*** button in **Vehicle Category**
* User clicks ***Delete Vehicle Form*** button and the system displays a search bar with ***Search*** and ***Done*** button
* User enters *VIN* in the search bar
* Upon:
* Click ***Submit*** 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**

|  |
| --- |
| DELETE FROM *`*Vehicle*`* WHERE Vehicle.VIN = '$enteredVIN'; |

* 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:**

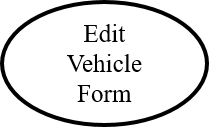
* Show ***Add Vehicle Form, Edit Vehicle Form, View Vehicle Form, Delete Vehicle Form*** button in **Vehicle Category**
* User clicks ***View Vehicle Form*** button and the system displays a new empty vehicle form with ***Search*** and ***Done*** button
* User enters *VIN* in the search bar
* Upon:
* 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

|  |
| --- |
| SELECT  *VIN, Vehicle\_Type, Vehicle\_Manufacturer,*  *Model\_Name, Model\_Year, Color, Mileage, Vehicle\_Description*  FROM *`* Vehicle*`* WHERE Vehicle.VIN = '$enteredVIN'; |

* + - 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

|  |
| --- |
| SELECT  *VIN, Vehicle\_Type, Vehicle\_Manufacturer,*  *Model\_Name, Model\_Year, Color, Mileage, Vehicle\_Description*  FROM *`* Vehicle*`* WHERE Vehicle.VIN = '$enteredVIN'; |

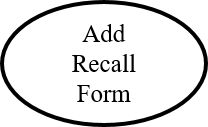
* Upon:
* Click ***Submit*** button;
  + If *VIN* is Null:
    - Show Error Message and Return to **Vehicle Category**
  + Else:
    - Display the information of this *VIN* from Vehicle Table and User edits the information
    - Click ***Done*** button:
      * If all of the items except *Vehicle\_Description* are not null and *Vehicle\_Description* can be Null or Not Null:
        + Write to the Vehicle table and Go back to **Vehicle Category**

|  |
| --- |
| UPDATE *`*Vehicle*`*  SET VIN = '$enteredVIN',  Vehicle\_Type = '$enteredVehicle\_Type ',  Vehicle\_Manufacturer = 'enteredVehicle\_Manufacturer,  Model\_Name = 'enteredModel\_Name',  Model\_Year = 'enteredModel\_Year',  Color = 'enteredColor',  Mileage = 'enteredMileage',  Vehicle\_Description = 'enteredVehicle\_Description'; |

* + - * 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:**

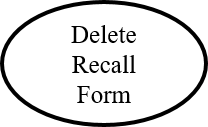
* Show ***Add Recall Form, Edit Recall Form, View Recall Form, Delete Recall Form*** button in **Repair/Recall Repository**
* User clicks ***Add Recall Form*** button and the system displays a new recall form with ***Submit*** and ***Cancel*** button
* User enters *NHTSA\_Recall\_Campagin\_Number, Recall\_Description, 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 itemsare not Null:
    - Write to Recall Table and go back to **Repair/Recall Repository**

|  |
| --- |
| INSERT INTO *`*Recall*`*  VALUES(  '$enteredNHTSA\_Recall\_Campagin\_Number',  '$enteredRecall\_Description', '$enteredRecall\_Manufacture',  ); |

* + 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:**

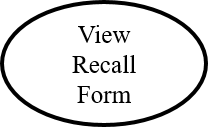
* Show ***Add Recall Form, Edit Recall Form, View Recall Form, Delete Recall Form*** button in **Repair/Recall Repository**
* User clicks ***Delete Vehicle Form*** button and the system displays a search bar with ***Search*** and ***Done*** button
* User enters *NHTSA\_Recall\_Campagin\_Number* in the 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**

|  |
| --- |
| DELETE FROM *`*Recall*`*  WHERE Recall.NHTSA\_Recall\_Campagin\_Number = '$enteredNHTSA\_Recall\_Campagin\_Number'  AND Recall.NHTSA\_Recall\_Campagin\_Number NOT IN (  SELECT DISTINCT(NHTSA\_Recall\_Campagin\_Number)  FROM Repair  WHERE Repair.NHTSA\_Recall\_Campagin\_Number = '$enteredNHTSA\_Recall\_Campagin\_Number'  ); |

* 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:**

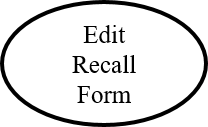
* Show ***Add Recall Form, Edit Recall Form, View Recall Form, Delete Recall Form*** button in **Repair/Recall Repository**
* User clicks ***View Recall Form*** button and the system displays a search bar with ***Search*** and ***Done*** button
* User enters *NHTSA\_Recall\_Campagin\_Number* in the search 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

|  |
| --- |
| SELECT *NHTSA\_Recall\_Campagin\_Number, Recall\_Description, Recall\_Manufacture*  FROM *`*Recall*`*  WHERE Recall.NHTSA\_Recall\_Campagin\_Number = '$enteredNHTSA\_Recall\_Campagin\_Number' |

* + - 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

|  |
| --- |
| SELECT *NHTSA\_Recall\_Campagin\_Number, Recall\_Description, Recall\_Manufacture*  FROM *`*Recall*`*  WHERE Recall.NHTSA\_Recall\_Campagin\_Number = '$enteredNHTSA\_Recall\_Campagin\_Number' |

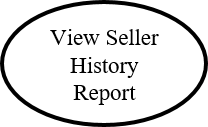
* Upon:
* Click ***Submit*** button;
  + If *NHTSA\_Recall\_Campagin\_Number* is Null or *NHTSA\_Recall\_Campagin\_Number* is not in Recall.*NHTSA\_Recall\_Campagin\_Number*:
    - Show Error Message and Return to **Repair/Recall Repository**
  + Else:
    - Display the information of this *NHTSA\_Recall\_Campagin\_Number* 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**

|  |
| --- |
| //if there is no conflict for the NHTSA\_Recall\_Campagin\_Number.  UPDATE *`*Recall*`*  SET NHTSA\_Recall\_Campagin\_Number = '$enteredNHTSA\_Recall\_Campagin\_Number',  Recall\_Description = '$enteredRecall\_Description ',  Recall\_Manufacture = 'entered Recall\_Manufacture' |

* + - * 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.

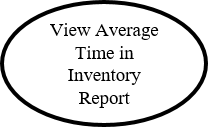
Subtasks: Mother Task is not needed. No decomposition needed.

**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 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**.**

|  |
| --- |
| SELECT Vehicle.Vehicle\_Type AS Vehicle\_Type, AVE(tbl.DateDiff) AS Average\_Time\_In\_Inventory  FROM (  SELECT sell.VIN AS VIN, DATEDIFF(day,buy.Purchase\_date, sell.Sale\_date) AS DateDiff  FROM `Sell\_transaction` AS sell  LEFT JOIN Buy\_transaction AS buy  ON sell.VIN = buy.VIN  WHERE buy.Purchase\_date IS NOT NULL AND sell.Sale\_date IS NOT NULL;  )tbl  LEFT JOIN `Vehicle`  ON Vehicle.VIN = tbl.VIN  GROUP BY Vehicle. Vehicle\_Type  ORDER BY Vehicle. Vehicle\_Type; |