Login

Abstract Code

- User enters *username* ($Username), *password* ($Password) input field.
- If data validation is successful for both fields, then:
    - When ***Enter*** button is clicked:

```
SELECT password FROM User WHERE username = '$Username';
```

        - If User record is found but User.password != '$Password':
            - Go back to Login form, with error message.
        - Else:
            - Store login information as session variable '$UserID'.
            - Go to Main Menu form
- Else username and password input fields are invalid, display **Login** form with error message.

Add Vehicle:

Abstract code:

- If the user has required access, then user can enter the details in the **Add Vehicle** form.
- The user enters the customer details in the seller field then hit search.
- If the customer record is found then seller is linked to the vehicle.
- Else, user is shown no match and prompted to add customer information including the following records:

Street ($Street)

City ($City)

State ($State)

Postal Code ($Zip)

Phone number ($Phone)

Email ($Email)

First name (Individual) ($FirstName)

Last Name (Individual) ($LastName)

Driver License Number (Individual) ($DLNumber)

Tax ID Number (Business) ($TaxID)

Business Name (Business) ($BusinessName)

Primary Contact Name (Business) ($ContactName)

Primary Contact Title (Business) ($ContactTitle)

```
INSERT INTO Customer (email,street,city,state,zip,phone) VALUES
(email=`$Email`,street=`$Street`,city=`$City`,state=`$State`,zip=`$ZipCode`,
phone=`$Phone`);
```

```
INSERT INTO Individual (DLNumber,first_name,last_name) VALUES
(DLNumber=`$DLNumberI`,first_name=`$FirstName`,last_name=`$LastNa
me`);
```

```
INSERT INTO Business
(TaxID,business_name,primary_contact_name,primary_contact_title)
VALUES (TaxID = `$TaxID`, business_name = `$BusinessName`,
primary_contact_name = `$ContactName`, primary_contact_title = `$
ContactTitle`);
```

- Once customer is selected, user fills out Vehicle Information form.
- User enters the *VIN* ($VIN)*, vehicle type* ($Type)*, condition* ($Condition)*, manufacturer* ($Manufacturer)*, model year* ($Year)*, model name* ($Model)*, color(s)* ($Color)*, description* ($Description)*, mileage* ($Mileage)*, Blue Book value* ($PurchasePrice), and *date of purchase* ($PurchaseDate) fields.
- User clicks Submit button.
- If data validation is successful for all required fields:
    - User is brought to the detail page for the vehicle.
- Else:
    - Invalid fields show an error message.

```
INSERT INTO Vehicle (VIN, mileage, model_name, model_year,
vehicle_condition, description, type, manufacturer) VALUES (VIN='$VIN',
mileage='$Mileage', model_name='$Model', model_year='$Year',
vehicle_condition='$Condition', description='$Description', type='$Type',
manufacturer='$Manufacturer');
```

- For each vehicle color:

```
INSERT INTO VehicleColor (VIN, color) VALUES (VIN='$VIN', color='$Color);
```

Record Sales Transaction:

Abstract Code

User clicks on sell vehicle button and brought to the sales order form

If the correct customer record is entered then buyer details is linked to the sale.

Else

The relevant error message is shown and the user is redirected to the customer information page

Once customer is selected and linked to the sale then the user fills in the required details including:

Seller

Sales Price

Sales Date

Buyer

Salesperson

```
INSERT INTO `SalesTransaction` (Seller, SalesPrice, SalesDate, Buyer,
Salesperson) VALUES (`Salesrecords`.Seller=`$Seller Name`,
`Salesrecords`.SalesPrice=`$Sales Price`, `Salesrecords`.SalesDate==`$Sales
Date`, `Salesrecords`.Buyer==`$Buyer Name`,
`Salesrecords`.Salesperson==`$Sales Person`);
```

Registered User Search:

Abstract code:

User need to fill the login details. Then fill the search criteria similar to the public search with addition of the VIN.

If the user access match the inventory clerk, then the result will show unsold vehicles including those under repair.

If the user is a salesperson the result will show unsold vehicles which are not under repair

If the user is a manager or owner the results will show all vehicles without any limitations.

CASE
WHEN `User`.Category = “Inventory Clerk” AND ‘Repairs’.Status=”Not Completed” **OR** ‘Repairs’.Status = ”Completed” AND ‘Vehicle`,Seller =NULL THEN

SELECT VIN, Vehicle type, Model, Manufacturer, Color, Mileage, Sales Price **FROM** `Inventory` **WHERE** Vehicle.VIN=`$VIN`, Vehicle. Vehicle type =`$ Vehicle type ` , Vehicle. Model Year =`$ Model Year `, Vehicle. Manufacturer =`$ Manufacturer `, Vehicle. Model =`$ Model ` , Vehicle. Color =`$ Color ` , Vehicle. Mileage =`$ Mileage ` , Vehicle. Sales Price =`$ Sales Price` limit 50

WHEN `User`.Category = “Sales person” AND ‘Repairs’.Status=”Completed” AND ‘Vehicle`,Seller =NULL THEN

SELECT VIN, Vehicle type, Model, Manufacturer, Color, Mileage, Sales Price FROM `Inventory` WHERE Vehicle.VIN=`$VIN`, Vehicle. Vehicle type =`$ Vehicle type ` , Vehicle. Model Year =`$ Model Year `, Vehicle. Manufacturer =`$ Manufacturer `, Vehicle. Model =`$ Model ` , Vehicle. Color =`$ Color ` , Vehicle. Mileage =`$ Mileage ` , Vehicle. Sales Price =`$ Sales Price` limit 50

WHEN `User`.Category = “Manager” AND ‘Repairs’.Status=”Not Completed” OR ‘Repairs’.Status = ”Completed” AND ‘Vehicle`.Seller =NULL OR ‘Vehicle`.Seller =NOT NULL

SELECT VIN, Vehicle type, Model, Manufacturer, Color, Mileage, Sales Price **FROM** `Inventory` **WHERE** ‘Repairs’.Status=”Not Completed” **OR** WHERE ‘Repairs’.Status = ”Completed”

, Vehicle.VIN=`$VIN`, Vehicle. Vehicle type =`$ Vehicle type ` , Vehicle. Model Year =`$ Model Year `, Vehicle. Manufacturer =`$ Manufacturer `, Vehicle. Model =`$ Model ` , Vehicle. Color =`$ Color ` , Vehicle. Mileage =`$ Mileage ` , Vehicle. Sales Price =`$ Sales Price` limit 50

END;

Public Search

Show **“Vehicle Type”**,**”Manufacturer”**, **“Model year”**, **“Color”** and **“Keyword”** tabs in the navigation bar.

Upon clicking on search:

If any available vehicle matches the above fields and the vehicle is not under repair return the list with 50 records limit and the attributes including: **“VIN”**, **“Vehicle type”**,**” Year”**, **“Manufacturer”** , **“Model”**, **“Color”**,**”Mileage”** and **“Sales Price”**.

Else :

Return the message **“Sorry, it looks like we do not have that in stock!”**

```
SELECT VIN, Vehicle type, Model, Manufacturer, Color, Mileage, Sales Price from `Inventory` WHERE Vehicle.VIN=`$VIN`, Vehicle. Vehicle type =`$ Vehicle type ` , Vehicle. Model Year =`$ Model Year `, Vehicle. Manufacturer =`$ Manufacturer `, Vehicle. Model =`$ Model ` , Vehicle. Color =`$ Color ` , Vehicle. Mileage =`$ Mileage ` , Vehicle. Sales Price =`$ Sales Price` limit 50
```

Generate Seller History Report

```
SELECT I.Lastname + ', ' + I.First name AS seller name, COUNT(V.VIN) AS
number of vehicles, AVG(P.Purchase Price) AS average price, AVG(R.VIN)
AS average repairs
FROM Individual AS I INNER JOIN Purchase Transaction AS P ON I.DL
Number = P.DL Number/Tax ID
LEFT JOIN Vehicle AS V ON P.VIN = V.VIN
LEFT JOIN Repair AS R ON V.VIN = R.VIN
GROUP BY seller name
ORDER BY number of vehicles DESC, average price ASC

SELECT B.Business name AS seller name, COUNT(V.VIN) AS number of
vehicles, AVG(P.Purchase Price) AS average price, AVG(R.VIN) AS average
repairs
FROM Business AS B INNER JOIN Purchase Transaction AS P ON B.Tax ID =
P.DL Number/Tax ID
LEFT JOIN Vehicle AS V ON P.VIN = V.VIN
LEFT JOIN Repair AS R ON V.VIN = R.VIN
GROUP BY seller name
ORDER BY number of vehicles DESC, average price ASC
```

Generate Price Per Condition

```
SELECT VT.Type AS vehicle type, V.Condition AS vehicle condition,
AVG(P.Purchase Price) AS average price
FROM Vehicle Type AS VT LEFT JOIN Vehicle AS V ON VT.Type = V.Type
INNER JOIN Purchase Transaction AS P ON V.VIN = P.VIN
GROUP BY vehicle type, vehicle condition
```

Generate Repair Statistics Report

```
SELECT VE.Vendor name AS vendor name, COUNT(R.Vendor name) AS
number of repairs, SUM(R.Total cost) AS total dollar amount,
AVG(COUNT(R.VIN)) AS average number of repairs per vehicle,
AVG(DATEDIFF(day, R.Start Date, R.End Date)) as average length of
time(days)
FROM Vendor AS VE INNER JOIN Repair AS R ON VE.Vendor name =
R.Vendor name
```

Generate Monthly Sales Report

```
SELECT YEAR(S.Sales Date) AS year, MONTH(S.Sales Date) AS month,
COUNT(S.VIN) AS number of vehicles sold, SUM(S.Sales Price) AS total sales
income, (SUM(S.Sales Price)- (R.Total Cost)) AS total net income
FROM Sales Transaction AS S INNER JOIN Vehicle AS V ON S.VIN = V.VIN
LEFT JOIN Repair AS R ON V.VIN = R.VIN
GROUP BY year, month
ORDER BY year DESC, month DESC

SELECT YEAR(S.Sales Date) AS year, MONTH(S.Sales Date) AS month, U.First
Name + ‘ ’ + U.Last Name AS salespeople, COUNT(S.VIN) AS number of
vehicles sold, SUM(S.Sales Price) AS total sales income
FROM Sales Transaction AS S INNER JOIN Vehicle AS V ON S.VIN = V.VIN
LEFT JOIN User AS U ON S.username = U.username
GROUP BY year, month, salespeople
ORDER BY year DESC, month DESC, number of vehicles sold DESC, total
sales income DESC
```

Generate Average Time in Inventory Report

```
SELECT VT.Type as vehicle type, AVG(DATEDIFF(day, P.Purchase Date,
S.Sales Date)) as average  time in inventory(days)

FROM Vehicle Type AS VT LEFT JOIN Vehicle AS V ON VT.Type = V.Type

INNER JOIN Purchase Transaction AS P ON V.VIN = P.VIN

INNER JOIN Sales Transaction AS S ON V.VIN = S.VIN

GROUP BY vehicle type
```

```
CASE

WHEN `User`.Category = “Manager” THEN

SELECT VIN, Seller, Sales Price,Sales Date, Buyer, Salesperson FROM `Sales`
WHERE `Sales`.Seller = NOT NULL,

SELECT Condition, Sales Price FROM `Sales` WHERE `Sales`.Seller = NOT NULL
ORDER BY ASC,

SELECT (SELECT Sales Date FROM Sales)-(SELECT Purchase FROM Vehicle)) AS
Inventory Age WHERE ‘Sales’.Seller = NOT NULL ORDER BY Inventory Age ASC,

SELECT VIN, Status, Vendor name, Description, Start Date, End Date, NHTSA Recall
Number, Total Cost FROM Repairs WHEN `Repairs’.Status = NOT NULL,

SELECT Count(*) FROM Seller WHERE YEAR (Sales Date) = ‘2019’
GROUP BY MONTH (Sales Date)

WHEN `User`.Category = “Sales person” THEN

SELECT VIN, Seller, Sales Price,Sales Date, Buyer, Salesperson FROM `Sales`
WHERE `Sales`.Seller = NOT NULL,

SELECT Condition, Sales Price FROM `Sales` WHERE `Sales`.Seller = NOT NULL
ORDER BY ASC,

SELECT (SELECT Sales Date FROM Sales)-(SELECT Purchase FROM Vehicle)) AS
Inventory Age WHERE ‘Sales’.Seller = NOT NULL ORDER BY Inventory Age ASC,

SELECT Count(*) FROM Seller WHERE YEAR (Sales Date) = ‘2019’
GROUP BY MONTH (Sales Date)

WHEN `User`.Category = “Inventory Clerk” THEN

SELECT (SELECT Sales Date FROM Sales)-(SELECT Purchase FROM Vehicle)) AS
Inventory Age WHERE ‘Sales’.Seller = NOT NULL ORDER BY Inventory Age ASC,

SELECT VIN, Status, Vendor name, Description, Start Date, End Date, NHTSA Recall
Number, Total Cost FROM Repairs WHEN `Repairs’.Status = NOT NULL,

END;
```