BuzzCars

Table of Contents

Data Types

User

Attribute	Data types	Nullable
username	String	Not Null
password	String	Not Null
firstName	String	Not Null
lastName	String	Not Null
role	String	Not Null

Vehicle

Attribute	Data types	Nullable
vin	String	Not Null
vehicleType	String	Not Null
modelYear	int	Not Null
manufacturer	String	Not Null
modelName	String	Not Null
fuelType	String	Not Null
color	List<String>	Not Null
mileage	int	Not Null
description	String	Null

Customer

Attribute	Data types	Nullable
email	String	Not Null
phoneNumber	String	Not Null
street	String	Not Null
city	String	Not Null
state	String	Not Null

postalCode	String	Not Null

Individual

Attribute	Data types	Nullable
driverLicense	String	Not Null
firstName	String	Not Null
lastName	String	Not Null

Business

Attribute	Data types	Nullable
taxID	String	Not Null
businessName	String	Not Null
name	String	Not Null
title	String	Not Null

PartOrder

Attribute	Data types	Nullable
orderNumber	String	Not Null
orderDate	Date	Not Null
totalCost	float	Not Null
quantity	int	Not Null

Vendor

Attribute	Data types	Nullable
name	String	Not Null
phoneNumber	String	Not Null
street	String	Not Null
city	String	Not Null

state	String	Not Null
postalCode	String	Not Null

Part

Attribute	Data types	Nullable
partNumber	String	Not Null
status	String	Not Null
description	String	Not Null
cost	float	Not Null

Sells to

Attribute	Data types	Nullable
purchaseDate	Date	Not Null
purchasePrice	float	Not Null
condition	String	Not Null

Buys from

Attribute	Data types	Nullable
date	Date	Not Null

Business Logic Constraints

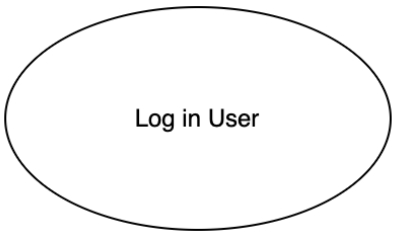
- Only employees of BuzzCars and the owner can log in the system. General public cannot.
- Users (the owner, managers, inventory clerks and sales people) can look up and view customers.
- The owner, managers, and inventory clerks can view all vehicles while public and sales people can view vehicles that are ready for sale
- Users (the owner, managers, inventory clerks and sales people) can search vehicles by VIN while the public cannot.
- Only the owner can update the list of vehicle types and manufacturers.
- A car can only be sold to BuzzCars once and also sold by BuzzCars once.
- Only one vehicle appears in one transaction (sale/purchase).
- Only the owner and managers can see reports.
- Vehicle conditions (Excellent, Very Good, Good, Fair) are determined and entered by inventory clerks.
- Parts status (ordered/received/installed) can be updated by inventory clerks. They cannot revert a part to a previous status (such as installed to ordered).
- The owner has complete functionality of the system.
- If a vehicle has any parts pending or not installed, it cannot be returned for any public search results or be sold.
- The sales price is calculated as 125% of the original purchase price (the price BuzzCars paid to buy the car) combined with 110% of any parts costs also associated with the vehicle. The sales price is not negotiable.
- A newly added car will show $0 total for parts.
- A newly added car should have a sale price of 125% times of the purchase price.

Task Decomposition with Abstract Code

Login

Task Decomp

Lock Types: Read-only on User table
Number of Locks: Single
Enabling Conditions: None
Frequency: Around 200 logins per day
Consistency (ACID): not critical, order is not critical
Subtasks: Mother Task is not needed. No decomposition is needed.



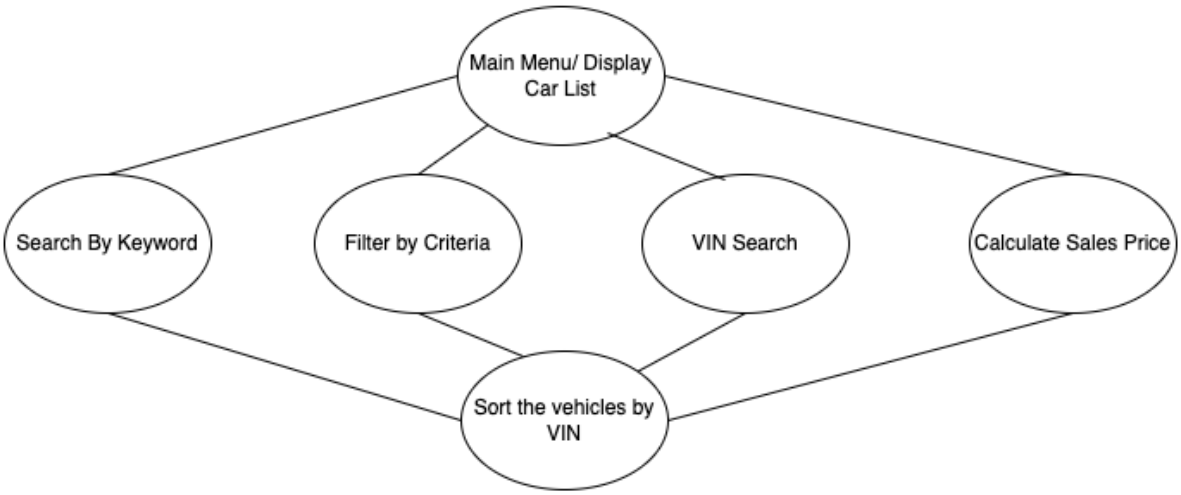
Abstract Code

- The User enters the username and password in the input form
- When the user clicks the ***Login*** button:
    - If both the username and the password fields are not empty:
        - If username is not found from the database or if the user's password does not match:
            - Display an error message ("The combination of your username and passwords do not match")and clear the field
        - Else:
            - Navigate to **Display Cars List (Main Page)** with the Search Bar with a logged in session
    - Else:
        - Display a message saying that both username and password are required
- Click the ***back arrow*** button to navigate to the main page

Display Cars List (Main Page)

Task Decomp

Lock Types: Read-only look up on Vehicle, PartOrders, Parts table
Number of Locks: 3
Enabling Conditions: The search by VIN is only enabled upon logged in as privileged users
Frequency: Depending on the traffic, around 1000 searches per hour
Consistency (ACID): order is important as it should be sorted by VIN in ascending order.
Subtasks: Mother Task is not needed. Search by VIN is enabled for privileged users. A dropdown menu is needed to display different options on the criteria: vehicle type, manufacturer, model year, fuel type, color. Calculation of the sale price is needed as a sub task. The vehicles are ordered in an ascending manner by VIN.



Abstract Code

- The default view is a navigation bar, and a search bar(for keywords) with a drop down button available next to the search bar
- If logged in:
    - Call **SearchbyVIN** function (defined below)
    - For the *navigation bars*
        - If the logged in user is an owner or a manager
            - Show "***Seller History***", "***Average Time***", "***Price Per Condition Report***", "***Parts Statistics Report***", "***Monthly Sales Report***" tabs
            - Click the ***Seller History*** button to navigate to the **View Seller History Report** task
            - Click the ***Average Time*** in Inventory Report button to navigate to the **View Average Time in Inventory Report** task
            - Click the ***Price Per Condition Report*** button to navigate to the **View Price Per Condition Report** task

- Click the ***Parts Statistics Report*** button to navigate to the **View Parts Statistics Report** task
- Click the ***Monthly Sales Report*** button to navigate to the **View Monthly Sales Report** task

- If the logged in user is an Inventory Clerk or owner:
  - Show "***Add Vehicle***" tabs
    - Click the ***Add Vehicle*** button to navigate to the **Add/Edit Vehicle Info** task

- If the ***dropdown*** button is clicked:
  - The menu expands and the different filtering criterias are shown with possible options showing up as buttons
- When the ***Search*** button is clicked:
  - If none of the criterias are chosen and the input field is empty:
    - Display a message "Please enter some keywords or choose at least one filtering criteria."
  - Else:
    - cars_result=SELECT * FROM Vehicle v WHERE criterias=chosen criterias
    - If no cars have the matching criterias:
      - Display a message: "Sorry, it looks like we don't have that in stock!"
    - Else:
      - Order the cars from with ascending VIN
      - For all the returned cars:
  - Display the VIN, vehicle type, model year, manufacturer, model, fuel type, colors, mileage, and sale price(calculated using **CalculateSalesPrice**()) in a ***Card*** component
  - If a ***Card*** component of a car is clicked, navigate to the detailed page for the car

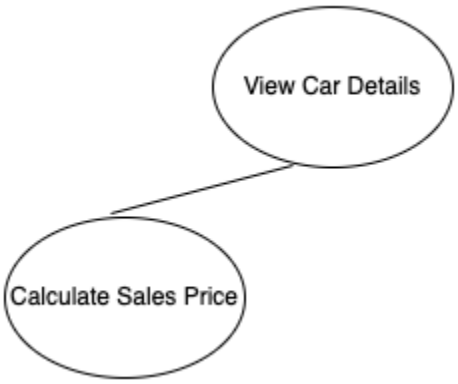
- function **SearchbyVIN**():
  - User enters a full VIN ('$vin') and clicks ***Search by VIN*** button
  - Run the **Search by VIN** task: query for vehicle information where $vin is the vehicle identification number entered by the user
    - if *User*.role is InventoryClerks, Managers, or Owner:
      - find the vehicle(s) using Vehicle.vin that matches $vin
        - if the vehicle information is found:
          - Display Vehicle vin, vehicleType, modelYear, manufacturer, modelName, fuelType, color, mileage, salesPrice (**CalculateSalesPrice()**)

        - else:
            - Display message “Sorry, it looks like we don’t have that in stock!”
    - else if *User*.role is SalesPeople:
        - find the vehicle(s) using Vehicle.vin that includes $vin and none of the associated part status is other than installed
            - if the vehicle information is found:
                - Display Vehicle vin, vehicleType, modelYear, manufacturer, modelName, fuelType, color, mileage, salesPrice(calculated using **CalculateSalesPrice**())
                - Display link to **Sales Order Form** next to each vehicle
            - else
                - Display message “Sorry, it looks like we don’t have that in stock!”
- function **CalculateSalesPrice**():
    - SELECT SUM(Parts cost) *1.1 +1.25* purchase price FROM Vehicle JOIN PartOrder JOIN Part JOIN Buys

View Car Details

Task Decomp

Lock Types: Read-only look up on Vehicle, PartOrders, Parts table
Number of Locks: 3
Enabling Conditions: After the users choose and select the specific car.
Frequency: Depending on the traffic, around 1000 queries per hour
Consistency (ACID): not critical. Order is not important.
Subtasks: Calculate the sale price, no other sub tasks needed.



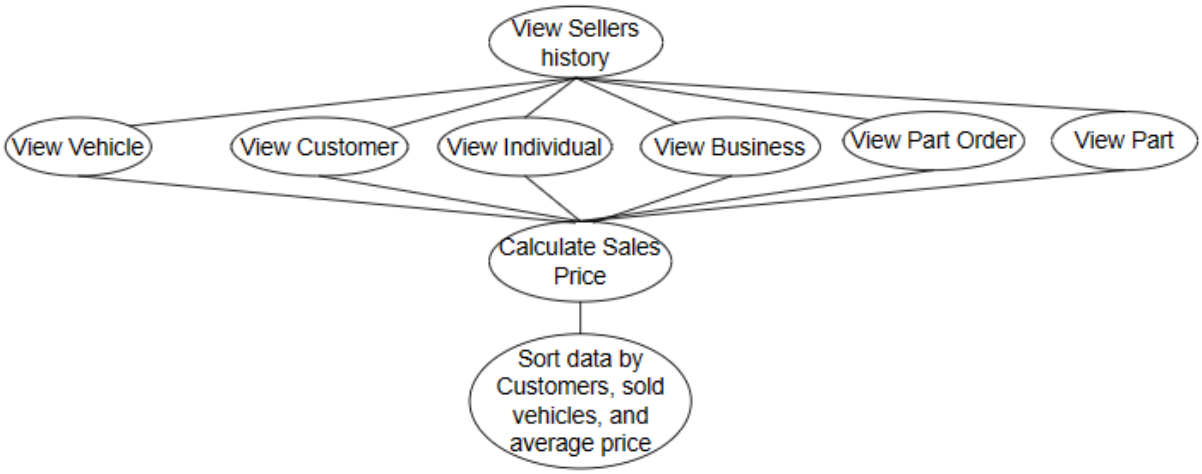
Abstract Code

- ***selected_car***= "SELECT * FROM Vehicle v WHERE v.VIN = @ VIN"
- Show all the attributes of the ***selected_car*** in an unordered list
- Click the back button on the top left corner to go back to the list of cars from the searching result in **Display Car List/Main Menu**
- If the logged in user is a Manager or an Owner:
    - Display ***"View Purchase History"*** button
    - Click the ***View Purchase History*** button to go to **View Purchase History** task
- If the logged in user is an Owner or Inventory Clerk
    - Display "***Update Part Order Status***", "***Add Part Order***" buttons
    - Click ***Update Part Order Status*** to navigate to **Part Order Status Form**
    - Click ***Add Part Order*** to navigate to **Add Parts Order Form**
- If the logged in user is an owner or Salespeople:
    - Display "***Sale Order***" buttons
    - Click the ***Sales Order*** button to navigate to the **Sale Order** form
- function **CalculateSalesPrice**():
    - SELECT SUM(Parts cost) *1.1 +1.25* purchase price FROM Vehicle JOIN PartOrder JOIN Part JOIN Buys

View Seller History (Report Page)

Task Decomp

Lock Types: Read-only on Vehicle, Customer, Individual, Business, PartOrder, and Part Tables
Number of Locks: Six
Enabling Conditions: User role attribute must be Manager or Owner
Frequency: Around 20 reports per day
Consistency (ACID): Critical
Subtasks: Mother Task is needed.



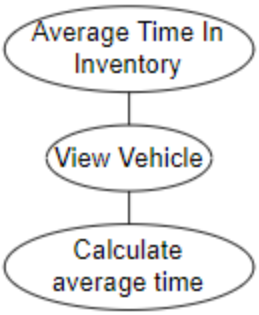
Abstract Code

- User clicks the ***View Seller History*** Button.
- If role validation is successful for manager or owner, then:
    - Initialize report data by creating an empty list to store data for the report.
    - For each ***vehicle*** in the ***Vehicle*** database:
        - Retrieve seller information (***CustomerID***, ***Address***, ***TaxID*** or ***driverLicense***).
        - Create variables to store calculations of the number of parts ordered and the total parts cost for the vehicle.
        - For each seller, calculate:
            - ***totalNumberOfVehiclesSold***.
            - ***averageSoldPrice*** for vehicles sold. Use the function **CalculateSalesPrice**() to calculate each individual vehicle's price first.
            - ***averageNumberOfPartsOrderedPerVehicle***.
            - ***averageCostOfPartsPerVehicle***.
            - For sellers meeting the criteria (average part cost >= $500 or average parts ordered >= 5):
            - Mark the seller's entry for highlighting (e.g., with a red background).
    - Sort by ***totalNumberOfVehiclesSold*** (descending) and Sort by ***averageSoldPrice*** (ascending) and group sellers together.
    - Present the report data, including seller names, metrics, and highlights.
    - The name of the seller (either first name and last name or company name, which should be displayed as a single column, not two different columns for each seller type)
- Else role validation fails then display no user rights message.

View Average Time in Inventory Report (Report Page)

Task Decomp

Lock Types: Read-only for Vehicle table
Number of Locks: One
Enabling Conditions: User role attribute must be Manager or Owner
Frequency: 20 reports per day
Consistency (ACID): Not critical
Subtasks: Mother Task is needed.



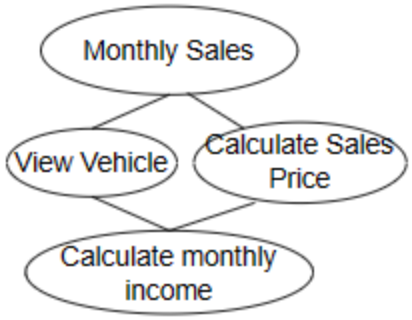
Abstract Code

- User clicks ***Average Time in Inventory*** button.
- If role validation is successful for manager or owner, then:
- Create ***averageTimeList*** variable.
- Create ***totalNumberOfVehicles variable***.
- For each ***vehicle*** in the dataset:
    - Calculate the time spent in inventory by subtracting the ***purchaseDate*** from the ***saleDate***.
    - Accumulate the calculated times for all vehicles.
    - Store each individual vehicle's average time in ***averageTimeList***
- Calculate the overall average time spent in inventory by dividing the accumulated time by the ***totalNumberOfVehicles*** .
- Display the **Average Time in Inventory** report to the user, including the calculated average time.
- Else role validation fails then the display message user doesn't have the necessary rights.

View Monthly/Yearly Summary Report (Report Page)

Task Decomp

Lock Types: Read-only for Vehicle table
Number of Locks: One
Enabling Conditions: User role attribute must be Manager or Owner
Frequency: Around 20 reports per day
Consistency (ACID): Not critical
Subtasks: Mother Task is needed.



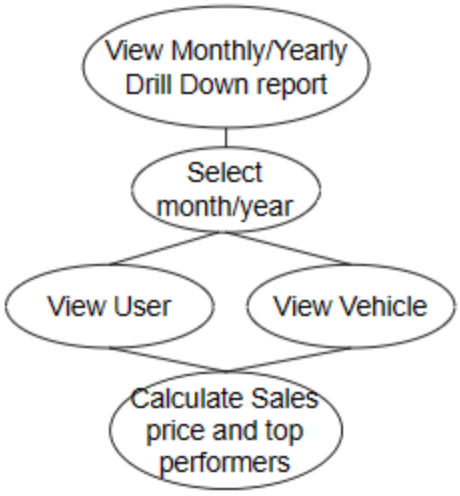
Abstract Code

- User clicks the '***View Monthly/Yearly summary report***' button.
- If role validation is successful for manager or owner, then:
    - Create variables ***SalesSummary*** list to accumulate sales data by year and month, ***totalNumberOfVehiclesSold***, ***totalSalesIncome***, and ***totalNetIncome***. Use the function **CalculateSalesPrice**() to calculate sales income.
    - For each vehicle in the dataset:
        - Extract the ***saleDate***, ***salesPrice***, ***purchasePrice***, and ***partsCosts***.
        - If the selected year or month has no sales data
            - Exclude the selected date from the report.
        - Determine the year and month of the sale date.
        - Update the relevant variables to ***SalesSummary***.
    - Sort the ***SalesSummary*** by year and month in descending order, with the most recent year and month as the first result.
    - Loop through the sorted ***SalesSummary*** and display the summary report to the user, listing for each year and month:
        - ***Total Number of Vehicles sold***.
        - ***Total Sales Income***.
        - ***Total Net Income***.

View Monthly/Yearly Drilldown Report (Report Page)

Task Decomp

Lock Types: Read-only for Vehicle, User tables
Number of Locks: Two
Enabling Conditions: User role attribute must be Manager or Owner
Frequency: On-demand by the user when drilling down from the summary report
Consistency (ACID): Not critical
Subtasks: Mother Task is needed.



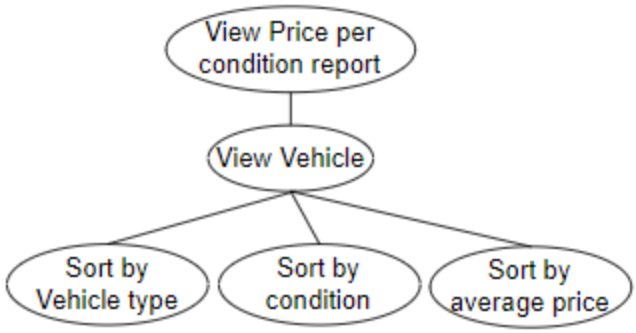
Abstract Code

- User selects the month and year for the drill down report.
- Create variables ***salespeople***, ***numberOfVehicles***, ***totalVehiclesSold***, and ***totalSales***.
- User clicks the '***View Monthly/Yearly Drilldown report***' button.
- If role validation is successful for manager or owner, then:
    - Extract the selected year and month from the user's request.
    - For each vehicle in the selected year and month:
        - Use the function **CalculateSalesPrice**() to calculate the price of vehicles and sort them in descending order for the ***salespeople.***
        - Identify the top-performing ***salespeople*** for the selected year and month.
        - Calculate the ***numberOfVehicles*** they sold and their ***totalSales***.
    - Sort the salespeople data by ***totalVehiclesSold*** (descending) and ***totalSales*** (descending) using a sorting algorithm
    - In the event of a tie in total ***vehiclesSold***, prioritize the salesperson element with the highest total sales.

View Price Per Condition Report (Report Page)

Task Decomp

Lock Types: Read-only for Vehicle table
Number of Locks: One
Enabling Conditions: User role attribute must be Manager or Owner
Frequency: 20 reports per day
Consistency (ACID): Not critical
Subtasks: Mother Task is needed.



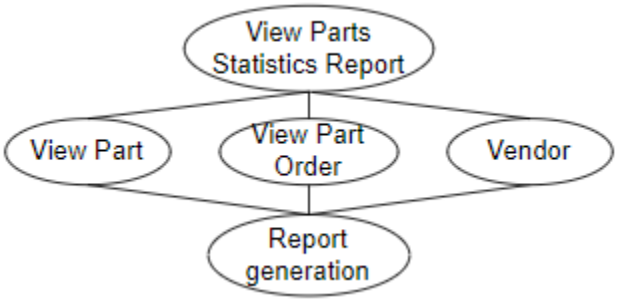
Abstract Code

- User clicks the **View Price Per Condition Report** button.
- Use the function **CalculateSalesPrice**() to calculate the price of vehicles
- If role validation is successful for manager or owner, then:
    - Sort data by ***vehicleType***
    - Sort data by ***condition***
    - Sort data by ***averagePrice***
- For each unique combination of ***vehicleType*** and ***condition***:
    - Filter cars that match the current combination of ***vehicleType*** and ***condition***.
    - If there are matching cars:
        - Calculate the ***averagePrice*** for these cars.
        - Display the ***averagePrice*** in the report.
    - Else display "$0" for this combination in the report.

View Parts Statistics Report (Report Page)

Task Decomp

Lock Types: Read only for Vendor, PartOrder, Part tables
Number of Locks: Three
Enabling Conditions: User role attribute must be Manager or Owner
Frequency: 20 reports per day
Consistency (ACID): Not critical
Subtasks: Mother Task is needed.



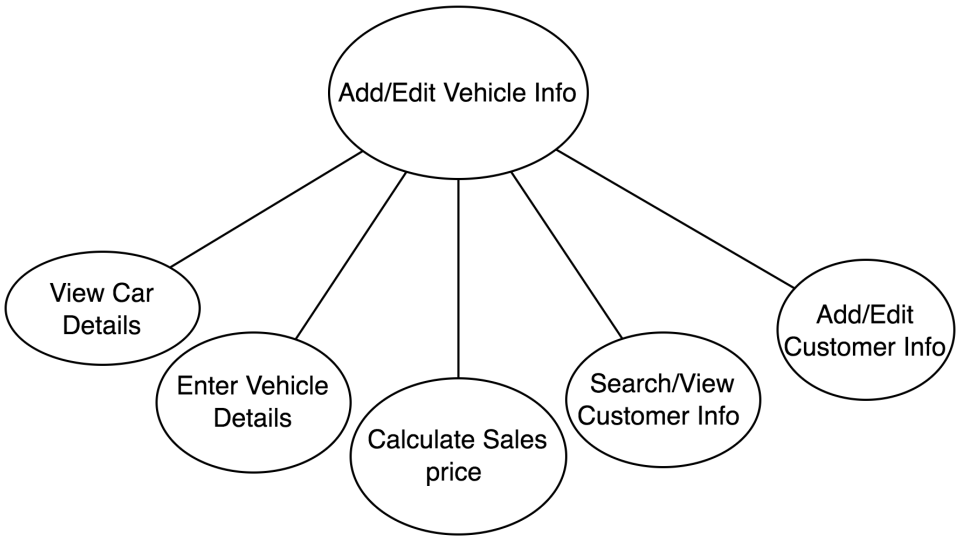
Abstract Code

- User clicks the ***View Parts Statistics Report*** button.
- Initialize lists/arrays for ***part***, ***part order***, ***vendor*** data, and ***report statistics***.
- For each part in the **Part**, **Vendor**, and **Part Order** table:
    - Store ***part***, ***partOrder***, ***vendor*** data in report statistics
- Sort report statistics by ***partNumber*** and ***orderNumber.***

Add/Edit Vehicle Info

Task Decomp

Lock Types: lookups and edit of vehicle and customer, business and individual
Number of Locks: 4
Enabling Conditions: enabled by user login and triggered by separate edit request when there is purchase transaction made by BuzzCars, user role attribute must be Inventory Clerk or Owner
Frequency: Different frequencies, mileage and description may have higher frequency
Consistency (ACID): not critical, order is not critical
Subtasks: Mother Task needed.



\

Abstract Code

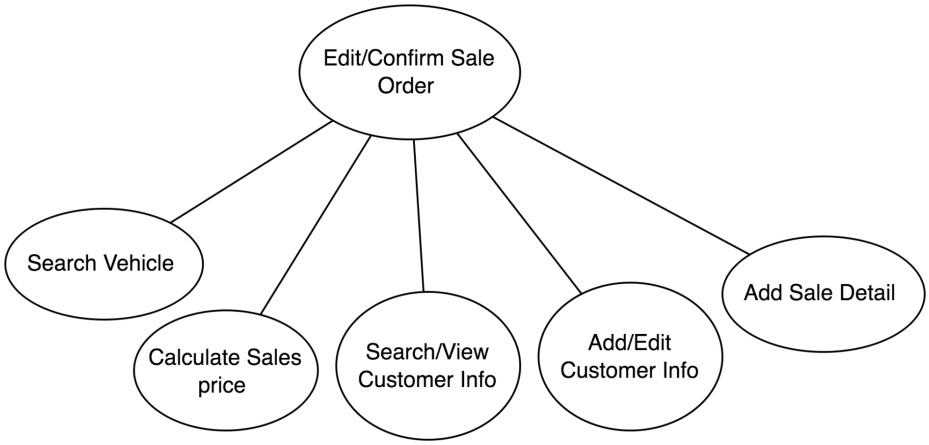
- **View car details**. Populate drop down list for ***vehicleType***, manufacturer, ***fuelType*** and ***color***. More details of this subtask are listed in its relevant section.
- If no View vehicle info available for the VIN entered, present the user with the button of ***Add Vehicle***.
- When no button is pushed, do nothing; when ***Add Vehicle*** is pushed then load the **Add Vehicle Form**., do the following

- **Calculate sale price** based on ***purchasePrice*** and ***partsPrice*** if there are any (i.e part prices most likely would not apply when a vehicle is just purchased)
    - function *CalculateSalesPrice*():
        - SELECT SUM(Parts cost) *1.1 +1.25* purchase price FROM Vehicle JOIN PartOrder JOIN Part JOIN Buys
- **Search/view customer** for existing customer; if not an existing customer, **add customer info**
- Click on ***Add a customer*** button to link the customer from previous step
    - If ***Save***: Update and store Vehicle info. View Vehicle Info.
    - If a wrong data type is entered, display an 'Error: wrong data type, please recheck your entry' message when the user clicks ***Save***.
    - If ***Cancel***: Go to View Vehicle info for existing VIN.
    - If ***Add another color***: Display the drop down list of colors.

Edit/Confirm Sale Order

Task Decomp

Lock Types: Lock Types: Lookup for vehicle, customers, individual and business; edit for sales(sales date and salesperson).
Number of Locks: 5
Enabling Conditions: enabled by user login and triggered by separate edit request when there is sale transaction by BuzzCars, user role attribute must be Salesperson or Owner
Frequency: low frequency
Consistency (ACID): Critical, sale price needs to be accurate and up to date when vehicle is sold
Subtasks: Mother Task needed.



Abstract Code

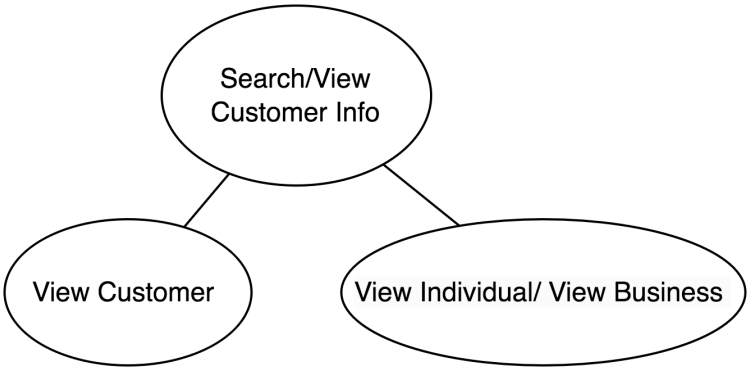
- **Search Vehicle** using VIN. View Vehicle.

- Click on the ***Sale the Car*** button then load the **Sale Order Form**. Do nothing if no button is clicked.
- **Calculate sale price** based on purchase price and parts price if there are any
    - function *CalculateSalesPrice*():
        - SELECT SUM(Parts cost) *1.1 +1.25* purchase price FROM Vehicle JOIN PartOrder JOIN Part JOIN Buys
- **Search for customer,** if not existing customer, **add customer info**
- Click on ***Add a customer*** to link the customer as seller from previous step
    - If ***Save***: Update and store Sale Order info. View Vehicle Info.
    - If a wrong data type is entered or required fields are missing, display an 'Error: please recheck your entry' message when the user clicks ***Save***.
    - If ***Cancel***: Go to View Vehicle info for existing VIN.

Search/View Customer Info

Task Decomp

Lock Types: Lookup for customer basic, individual or business
Number of Locks: 3
Enabling Conditions: enabled by user login and triggered by a separate purchase or sale transaction, user role attribute must be Inventory Clerk, salesperson or Owner
Frequency: low frequency, only happens when sale/purchase transaction happens
Consistency (ACID): not critical, order is not critical
Subtasks: Mother Task needed.



Abstract Code

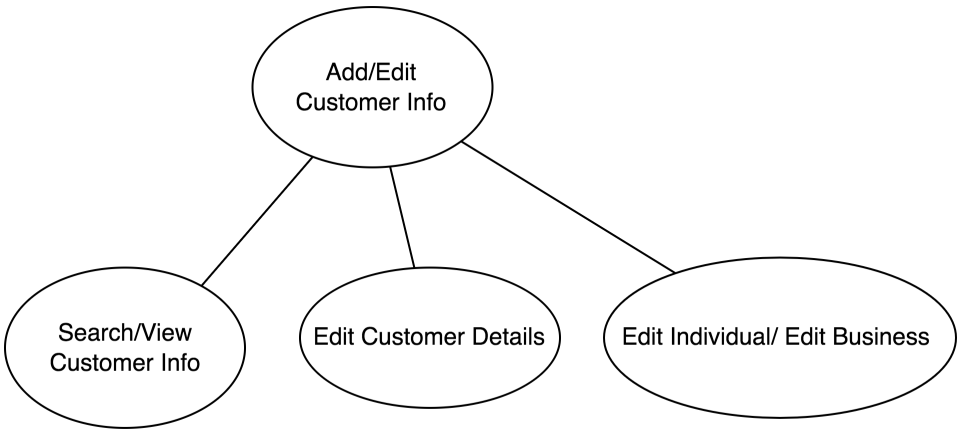
- Search by either driver's license or tax id. Find the current customer by driver's license if customer is individual, find customer using tax id if customer is business.
- Display email, phone number, customerID and address(state/city/street/postal code)
- Find Individual if searched by driver's license
    - Display driver's license, name(firstName/lastName)
- Find Business if searched by tax id

- Display taxid, business name, primary contact(name/title)

Add/Edit Customer Info

Task Decomp

Lock Types: Lock Types: Lookup and edit for customer basic, individual or business
Number of Locks: 3
Enabling Conditions: enabled by user login and triggered by a separate purchase or sale transaction, user role attribute must be Inventory Clerk, salesperson or Owner
Frequency: low frequency, only happens when sale/purchase transaction happens
Consistency (ACID): not critical, order is not critical
Subtasks: Mother Task needed.



Abstract Code

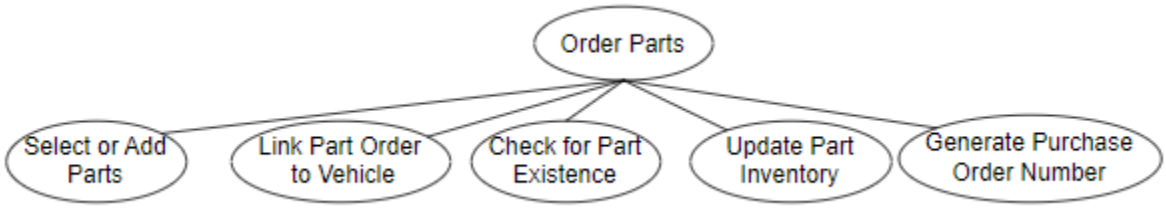
- **Search Customer Info** by either driver's license or tax id.
- If a customer profile is found, populate with the existing profile.
  - If ***Save***, update and store either Individual or Business, update and store Customer, view customer info.
  - if ***Cancel***, back to search customer info page
- If no existing customer profile found, display page with empty address, email and phoneNumber
- Present ***Individual*** and ***Business*** buttons in page bottom and do the following:
  - If click on ***Individual*** , display empty field for driver's license and name
  - If click on ***Business*** , display empty field for taxID, businessName and PrimaryContact
  - If ***Save***, update and store either Individual or Business, update and store Customer, view customer info.
  - If a wrong data type is entered or required fields are missing, display an 'Error: please recheck your entry' message when the user clicks ***Save***.

- if ***Cancel***, back to search customer info page

Order Parts (Add Parts Order Form)

Task Decomp

Lock Types: Read only Vehicle table, Update on Part, PartOrder, Vendor Tables
Number of Locks: Four
Enabling Conditions: User role attribute must be Inventory Clerk and screen must be on **Add Parts Order Form**
Frequency: Varies based on inventory needs
Consistency (ACID): Not critical
Subtasks: Mother Task is needed.



Abstract Code

- Select or Add Parts:
  Inventory clerk is presented with options to either select existing parts or add new parts to the part order.
- Check for Part Existence:
  Before adding a new part to the Part table, the system checks if a part with the same part number already exists.
  If a matching part is found, a message is displayed indicating that the part already exists and cannot be added again.
- If selecting existing parts:
  Inventory clerk chooses parts from a list of available parts. This is populated in a drop down menu via loop from each item in the part table.
  Specifies the quantity and clicks the ***Order*** button.
  The status of selected parts is automatically set to 'ordered.'
- Else adding a new part:
  Users will populate fields for Part table fields: ***partNumber***, ***description***, ***cost*** and Vendor table fields ***name***, ***phoneNumber***, ***street***, ***city***, ***state***, and ***postal code***, PartOrder Table field quantity for new parts.
  Validate user input data:
  - Ensure partNumber is unique.

- Validate cost as a numeric value.
- Validate phone number format.

If data validation fails, display an error message to the user indicating the validation issue.

If the user needs to add a vendor that is already existing:

Use the **Search Part Vendor** to populate the vendor fields.

Use the **Add Part Vendor** to add to the Vendor table.

Else vendor is not existing:

Add the Vendor through the **Add Part Vendor** function.

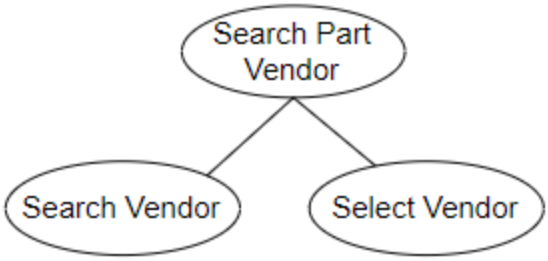
User clicks the ***Add Part*** button to send the part to the order.

- Link PartOrder to Vehicle:
  When the user confirms the part order, a new record is created in the PartOrder table. The PartOrder table includes a reference (e.g., foreign key) to the selected vehicle. Ensure that the part status is set to "ordered" automatically.
- Update Part Inventory:
  This action is performed automatically when a new part is added to the Part table. If a new part is added to the Part table, the system updates the inventory to reflect the addition of the new part.
- Generate Purchase Order Number:
  The system generates an automatic unique purchase order number for the part order based on the ***VIN*** of the associated vehicle and an ordinal number for the order (e.g., 123-01 for the first order for vehicle VIN 123).

Search Part Vendor (Add Parts Order Form)

Task Decomp

Lock Types: Read-only Vendor table
Number of Locks: One
Enabling Conditions: User role attribute must be Inventory Clerk and screen must be on **Add Parts Order Form**
Frequency: Varies based on inventory needs
Consistency (ACID): Not critical
Subtasks: Mother task is needed



Abstract Code

- The inventory clerk initiates the **Search Part Vendor** function while ordering parts within the **Add Parts Order Form**.
- The system provides a text-based search field that allows the clerk to enter keywords or phrases related to vendors.
- As the clerk types in the search field, the system dynamically updates the list of vendors displayed on the screen to match the search criteria.
- The list of vendors should include all information (***vendorID***, ***name***, ***address***, ***phoneNumber***) to help the clerk identify the correct vendor.
- The clerk can click on a vendor from the updated list to select it.
- When the clerk selects a vendor from the search results, the system associates this vendor with the current parts order.

Add Part Vendor (Add Parts Order Form)

Task Decomp

Lock Types: Update on the Vendor Table
Number of Locks: One
Enabling Conditions: User role attribute must be Inventory Clerk and screen must be on the **Add Parts Order Form**
Frequency: Occasional, as needed when adding a new part vendor.
Consistency (ACID): Not critical
Subtasks: Mother Task is not needed.

Abstract Code

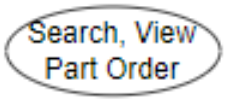
- Inventory clerk enters vendor information into the form fields.
- Click ***Add*** Button: The clerk initiates the process by clicking the ***Add*** button.

- The system verifies whether a vendor with the same name or identifier already exists in the Vendor Table.
- If Vendor Exists:
    - If a matching vendor is found, the system displays an error message indicating that the vendor already exists and cannot be added.
- Else Vendor Doesn't Exist:
    - The system creates a new vendor record in the Vendor Table using the entered information.

Search, View Part Order (Part Order Status form)

Task Decomp

Lock Types: Read-only on the PartOrder Table and Part Table
Number of Locks: Two
Enabling Conditions: User role attribute must be Inventory Clerk, and the screen must be on the **Part Order Status Form**
Frequency: Varies based on inventory needs
Consistency (ACID): Not critical
Subtasks: Mother Task is not needed.



Abstract Code

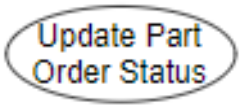
- The clerk can enter search criteria: ***orderNumber***, ***orderDate***, ***vendorID***, ***totalCost*** or ***status***, into the search fields provided on the screen.
- The system retrieves a list of part orders that match the search criteria from the database.
- The system displays the search results on the screen, including relevant information about each part order: ***orderNumber***, ***orderDate***, ***vendorID***, ***totalCost*** or ***status***.
- The clerk can click on a specific part order from the search results to view more details.
    - Click the **Update Part Order Status** button to move to the **Update Part Order Status** task
- The system displays the detailed information about the selected part order, ***orderNumber***, ***orderDate***, ***vendorID***, ***totalCost***, ***description***, and ***status***.

Update Part Order Status (Part Order Status form)

Task Decomp

Lock Types: Update lock on the PartOrder Table

Number of Locks: One
Enabling Conditions: User role attribute must be Inventory Clerk, and the screen must be on the **Part Order Status Form**.
Frequency: Occasional, when an inventory clerk needs to update the status of a part order.
Consistency (ACID): Critical, as it involves modifying data in the database.
Subtasks: Mother Task is not needed.



Abstract Code

- Choose a new status: The clerk chooses the new status for the selected part order. This status can typically be "ordered," "received," or "installed."
- If the ***status*** is ordered:
    - It can be changed to received or installed.
- If the ***status*** is received:
    - It can be changed to installed.
- If the ***status*** is installed:
    - ***Status*** cannot be changed.
- The clerk submits the updated ***status***.
- Update the Part Order: The system updates the ***status*** of the selected part order in the PartOrder Table with the new ***status*** chosen by the clerk.
- The system provides a confirmation message indicating that the part order ***status*** has been successfully updated.