Table of Contents:

Datatypes

Vehicle

Attribute	**Data Type**	**Nullable**
ManufacturerName	String	Not Null
MName	String	Not Null
Model	String	Not Null
Year	String	Not Null
InvoicePrice	Float(money)	Not Null
Color	List<string>	Not Null
VDescription	String	Nullable
VIN	String	Not Null

Customer

Attribute	**Data Type**	**Nullable**
Address	String	Not Null
PhoneNumber	String	Not Null
EmailAddress	String	Nullable

Repair

Attribute	**Data Type**	**Nullable**
RDescription	String	Not Null
CompletionDate	Date	Not Null
OdometerReading	Float	Not Null
LaborCharges	Float	Not Null
VehicleCustomerStartDate	Date	Not Null

Part

Attribute	Data Type	Nullable
PartNumber	String	Not Null
VendorName	String	Not Null
Quantity	Integer	Not Null
Price	Float (money)	Not Null

Sale

Attribute	Data Types	Nullable
PurchaseDate	Date	Not Nullabe
SoldPrice	Float (money)	Not Nullabe
SalesPerson	String	Not Nullabe
ListPrice	Float (money)	Not Nullabe

Individual

Attribute	Data Types	Nullable
Firstname	String	Not Nullable
LastName	String	Not Nullable
DriverLicenseNumber	String	Not Nullable

Business

Attribute	Data Type	Nullable
TIN	String	Not Nullable
BName	String	Not Nullable
PrimaryContact	String	Nullable

PCName	String	Nullable
Title	String	Nullable

Car

Attribute	Data Type	Nullable
NumberofDoors	Integer	Not Nullable

Convertible

Attribute	Data Type	Nullable
RoofType	String	Not Nullable
BackSeatCount	Integer	Not Nullable

Van

Attribute	Data Type	Nullable
VHasDriverSideBackDoor	Boolean	Not Nullable

Minivan

Attribute	Data Type	Nullable
MHasDriverSideBackDoor	Boolean	Not Nullable

Truck

Attribute	Data Type	Nullable
CargoCapacity	Integer	Not Nullable
CargoCoverType	Integer	Not Nullable

NumberOfRearAxles	Integer	Not Nullable

SUV

Attribute	**Data Type**	**Nullable**
DriveTrainType	String	Not Nullable
NumberOfCupholders	Integer	Not Nullable

Business Logic Constraints

Functionality and Users

1. Distinct categories of users include: inventory clerks, who add vehicles to inventory; salesperson, who will only have access to searching available inventory and entering sales transactions; service writers who enter repair details; managers who can view inventory, sales transactions, repair history, and reports; owner - Roland - who has access to everything and can perform any activity in the system.

Vehicle

1. Vehicles are tracked based on characteristics
2. Manufacturer name stored and list is not static. List should be updated from within database.
3. Model, name, model year free-form entered by user with restriction that model years cannot exceed the current year plus one.
4. Year entered must include century digits.
5. Invoice price includes dollars and cents.
6. Vehicle may have multiple colors.
7. List of colors is limited to: Aluminum, Beige, Black, Blue, Brown, Bronze, Claret, Copper, Cream, Gold, Gray, Green, Maroon, Metallic, Navy, Orange, Pink, Purple, Red, Rose, Rust, Silver, Tan, Turquoise, White, Yellow.
8. Optional description that contains accessories or equipment the vehicle has or any other information.

Customer

1. Customers can be either individual person or business.
2. Email address is optionally provided.
3. Business tax identification number is similar to Social Security number and assumed unique.

Sales

1. List price is 125% of invoice price which can be negotiated by customers to sold price. Could sell at higher price based on market conditions.
2. Buyer (customer) can purchase several vehicles, and if purchased at same time, should be handled as separate sales transactions.

Repair

1. Each repair must be associated with a vehicle.
2. Cannot be assumed that a repair for a vehicle is for the customer that originally bought it.
3. Date repair was completely could be same date as the start date or later, but not before.
4. Vehicle will never have more than one repair starting on the same date, and a repair must be completed before a new one can be started.
5. Total cost of repair is sum of cost of all parts used for repair and labor charges.

Anonymous Access

1. Allow for searching vehicles without login.
2. Initial state of application should be to open the search page with an option to login and display total number of vehicles for purchase.
3. Searchable based on vehicle type, manufacture, model year, color, list price (greater and/or less than an entered value), keyword (which searches manufacturer, model year, model name, and description and returns matches).
4. Use dropdown menus for fields other than keyword or list price - just allow for selecting single value and only return unsold vehicles.
5. If no vehicle meet search criteria display message stating "Sorry, it looks like we don't have that in stock!".
6. Return following attributes for each vehicle in search results: VIN, Vehicle Type, Model Year, Manufacturer, Model, Colors (if multiple colors return a single row with all colors listed), List Price, If entered keyword matched the description via checkbox or the like.
7. Results sorted by VIN in ascending order.
8. Optionally allow for user to sort or filter results by other columns.
9. Users can select individual results which will open a detail page that includes the above attributes, but not invoice price, and a description for selected result.

Privileged Access

1. No interface for creating or registering users and granting privileges as this will be done manually.
2. Login using username and password. After logging in will update to include access to appropriate functionality.

3. Common functionality - look up and add customers, but only available when performing various transactions and is not independently accessible.
4. Lookup by driver's license or tax ID.
5. If no result found, option to add customer and based on customer type the appropriate filed should be put into the system.

Inventory Clerks

1. After inventory clerk logins in, given access to "Add Vehicle" button or link, that will allow them to add new vehicles.
2. New vehicle form will gather relevant details along with the date it was added to inventory.
3. After submitting the data and successfully adding the vehicle to the database, clerk will be taken to the detail page for vehicle.
4. View similar to detail page shown to public users and should show same information but include fields for the invoice price.
5. After vehicle added, it is immediately available for sale and can be found from searching.

Salespeople

1. After logging in, on search page, with same layout as a public search, start with added option to search by VIN. Upon loading detail page, salesperson see same details as customers with an added button or link to sell vehicle that will load the sales order form.
2. Sales order form allows salespeople to look up a customer (or add them) and confirm the sale by entering sold price and sold date.
3. If sold price is less than or equal to 95% of invoice price, the system will eject the same.

Service Writer

1. Service writers enter details of repairs into the system. After login, have link or button to open repair form where enter VIN. If vehicle has not been sold or VIN does not match, error message is displayed, other wise the rest of repair form will be displayed.
2. Repair form should show same details as in above search result details.
3. If no repairs are open, repair form will allow user to start a new repair.
4. Odometer reading should be entered, and current date will be stored as the start date.
5. Service writer will be prompted to search for the customer associated with the repair. This may not be customer who originally purchased vehicle, may be necessary for the service writer to add a new customer to the system for the repair.
6. After creating the repair, form should allow service writer to enter labor charges and to add parts. When adding parts, relevant details entered by service writer.
7. Should perform basic input validation.
8. If vehicle has unfinished repair, repair form will only allow for updating labor charges, adding parts, or completing repair. Updates to labor charges cannot be less than previous value. Upon choosing complete repair, current date is stored on the repair as the completion date.

Managers

1. Have view-only access to all information along with reports.
2. After logging in, start on search screen similar to inventory clerk and salespeople.
3. Additionally have option to filter by sold vehicles, unsold vehicles, or all vehicles.
4. See all information on vehicle detail page including inventory clerk that added vehicle, invoice price, and date it was added.
5. If sold, has access to buyer's contact information (except driver's license/taxID number), list price, sold price, sales date, and salesperson's first and last name.
6. Repairs, if any, displayed under Repairs section and list of customer name (first, last for individual, or company name for companies), service writer who entered repair, repair's start date, end date, labor charges, parts cost, and total cost.

Roland Around

1. Complete access and able to vies all information and reports and do any activity.
2. Sales order form allows Roland to enter sold prices that are less than or equal to 95% of the invoice price, and repair form will allow him to update the labor charges on a repair to a value less than their previous value.

Reports

1. Access via link, button, dropdown menu on initial search page.

Sales by Color

1. Data reviewed once/month.
2. Report will list number of vehicles for each color sold in previous 30 days, previous year, over ll time - starting from the last available sale date.
3. More colors than these three categories, colors should be the rows for the table fo the report.
4. If vehicle has multiple colors, should be classified as multiple and not included in the count of vehicles that have only one color.
5. If color has no sales, should be "0".

Sales by Type

1. List vehicles sold by type in previous 30 days, previous year, and over all time, starting from last available sale date.
2. If type has no sales, should be "0".

Gross Customer Income

1. Report will list for top 15 customers the gross, not net, income received from them via vehicle sales and/or repairs, including any repairs in progress.

2. Report has two parts - first part is listing top 15 customers and should provide name (first/lat for individuals or business name for business), date of the first sale or repair start date, date of their most recent sale or repairs tart date, there number of sales, number of repairs, and gross income (sales and total repair costs combined). List of customers will be by gross income descending and by last/sale date descending.
3. Second part is drill-down for selected customer - made accessible by clicking on appropriate link.
4. Drill down will have section for vehicle sales and section for repairs.
5. Vehicle sales section should list the following details from each sale: sale date, sold price, VIN, year, manufacturer, model, and salesperson name.
6. List should be sorted by sale date descending and VIN ascending.
7. Repair section should list the following details for each repair: start date, end date (if repair not finished, this should not display any value), the vIN of the repaired vehicle, odometer reading, parts cost, labor cost, total cost, and service writer who opened the repair.
8. List should be sorted by start date descending, end date descending, and VIN ascending - however any incomplete repairs should be listed before competed ones with the same sorting criteria.

Repairs by Manufacturer/Type/Model

1. Purpose is to identify repair trends by manufacturer, type, and model, and will have two parts.
2. First part will list for each manufacturer: condo of repairs, sum of all parts costs, sum of all labor costs, sum of total repair costs, including any repairs in progress.
3. Manufacturers who's vehicles do not have any repairs should be listed on this report, and the report should be sorted by manufacturer name ascending.
4. From the first part, a drill-down for manufacturers that have repairs will be accessible via link or button.
5. Drill-down report will list a total for each vehicle type with repair count, parts costs, labor costs, and total costs.
6. Will also list further details for the vehicle type, with each model's repair counts, parts costs, labor costs, and total costs.
7. Totals by vehicle type should be displayed before details of its model.
8. Models and/or vehicle types which do not have repairs should be excluded from this report.
9. Drill-down should be sorted by repair count descending (by vehicle type sorted first, and then detail rows sorted).

Below Cost Sales

1. Report used to track all sales of vehicles that were below the invoice price of the vehicle.
2. For each vehicle, should list date, invoice price of vehicle, sold price, sold price/invoice price ration as a percentage, customer's name (as described above), and name of salesperson for the sale.
3. For a sale whose ration is less than or equal to 95%, background of that row should be highlighted red.
4. Sales should be listed by sales date descending and ratio descending.

Average Time in Inventory

1. This report, based on the difference between vehicle sales dates and when the vehicle was added to inventory, will display, by vehicle type, the average amount of time a vehicle remains in inventory, in days.
2. If no sales history, report should display N/A for that vehicle type.

Parts Statistics

1. List: vendor's name, number of parts supplied by that vendor, and total dollar amount spent on parts.

Monthly Sales

1. Most frequently used report and has two parts.
2. First, a summary page, which lists for all sales transactions, by year and month, the total number of vehicles sold, the total sales income, the total net income (sold price less invoice price), and the sold price/invoice ration as a percentage.
3. If a year or month does not have sales data, can be excluded.
4. When ratio is greater than or equal to 125%, its row should be highlighted green.
5. If ratio less than or equal to 110%, highlight yellow.
6. Results ordered by year and month descending with the most recent year and month as the first result.
7. From each year/month result, a drill down report must be accessible. Based on the sales data for that year and month, drill down will display the top performing salesperson, by showing salesperson's first and last name, number of vehicles they sold in that year and month and their total sales for that year and month.
8. Top sales person determined by sorting total vehicles descending followed by total sales descending.

Task
Screen/Page
entity
input field
Button
'$session variable'
Relationship

Task Decomposition with Abstract Code:

Searching for vehicles/Display Main Menu

Abstract Code

- (Initial state of the application and upon ***Return to Main Menu*** button clicks.)
- Clear all input fields
- Disable additional features unless the user is already login('$UserType' is not null or anonymous).
- Show ***search*** and ***login*** button on the **Main Menu screen**
- Query for information of counts of **Vehicle** using Vehicle.VIN, show this information as the total number of vehicles available for purchase in the system in the **Main Menu screen**
- Find available choices of searching input fields (Vehicle type, using **Vehicle**.type, Manufacturer, using **Vehicle**.Manufacturer, Model year, using **Vehicle**.Model and **Vehicle**.Year, Color, using **Vehicle**.Color) and fill available choices in the dropdowns for input fields in the **Main Menu screen**
- All users are able to fill in **List price** and **Keyword** input fields. Users also are able to select a **vehicle type**/**Manufacturer**/**Model**/**Year**/**Color/VIN(if enabled)** from the dropdown list in each input field in the **Main Menu screen**
- Upon:
    - Click the ***search*** button:
        - Read **Type**, **manufacturer**, **model**, **year**, **color, list price**, **keyword, VIN(if enabled)** input fields from the **Main Menu screen**
        - if data is valid and if there are **Vehicle** that matches the search criteria:
            - Display the matching **Vehicle** as a list sorted by **Vehicle**.VIN in ascending order on **Main Menu screen**
            - Session variable '$vehiclevin' = Vehicle.VIN
            - Users are allowed to select an individual result from the list. If the user selects a **Vehicle**:
                - Jump to the **View Vehicle Details** task.
        - Otherwise, display the message "Sorry, it looks like we don't have that in stock!" on the **Main Menu screen**
    - Click the ***login*** button: Jump to the **Login** task:
        - If '$UserType' returned:
            - '$UserType'=='Inventory clerks', enable ***Add Vehicle*** button and **new vehicle form** on **Main Menu screen**

- '$UserType'=='Salespeople', enable input field **VIN** in searching criteria on **Main Menu screen**
- '$UserType'=='Service Writer', enable ***repair form*** button on **Main Menu screen**
- '$UserType'=='Managers', enable ***report*** button, input field **VIN** in searching criteria, option to filter by soldvehicles, unsold vehicles, or all vehicles, Dropdown on **Main Menu screen**
- '$UserType'=='Roland Around', enable all the features on **Main Menu screen**

- Click the ***Add Vehicle*** button:
  - User will fill the **VIN**, **vehicle type**, **invoice price**, etc., along with the **date** it was added to inventory in **new vehicle form** on **Main Menu screen**.
  - Read those values from input fields and if Data is valid and **VIN** does not already exist as a Vehicle.VIN:
    - Inert new **Vehicle** instance with those values, then clear any success/error message, display a success message, '$vehiclevin' = **VIN** and call the **View Vehicle Details** task.
- Click the ***repair form*** button: Jump to the **Repair** task
- Click the ***report*** button: Read the choice from the report dropdown menu on **Main Menu screen** then call the corresponding report task

View Vehicle Details

Abstract Code

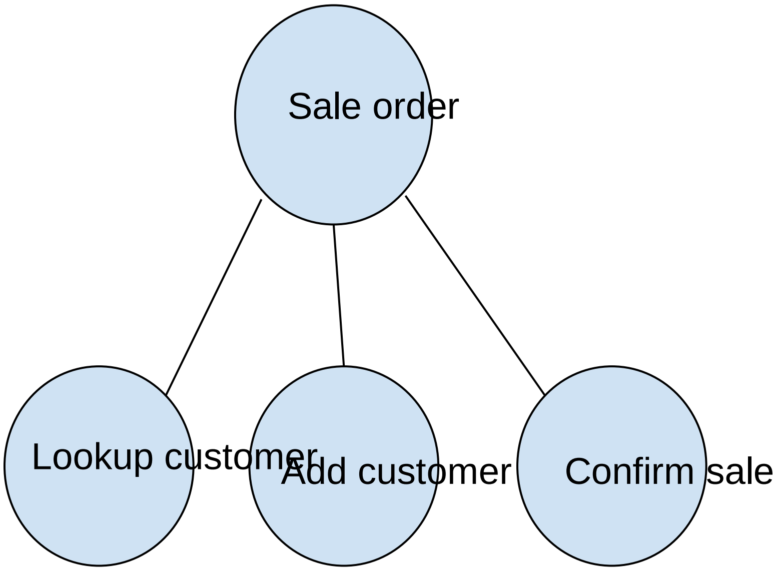
- User selected on Vehicle from the list on the **Main Menu screen**
- Display **Detail Page** Screen
- Enable link to ***sell the vehicle*** on **Detail Page** Screen if '$UserType'=='Salespoeple'
- Query for information about the **Vehicle** and it's details using '$vehiclevin' from the HTTP Session/Cookie.:
  - Display '$vehiclevin' on the **Detail Page**
  - Find and display the **Vehicle**.Type on the **Detail Page**
  - Find and display the attributes of the **Vehicle**.Type on the **Detail Page**
  - Find and display the **Vehicle**.Model and Vehicle.Year on the **Detail Page**
  - Find and display the **Vehicle**.Manufacturer on the **Detail Page**
  - Find and display the **Vehicle**.Color on the **Detail Page**
  - Find **Vehicle**.InvoicePrice and display invoice price times 125% as list price on the **Detail Page**
  - Find and display the **Vehicle**.VDescription on the **Detail Page**
  - If '$UserType'=='Inventory clerks':
    - Find and display the **Vehicle**.invoiceprice on the **Detail Page**
- Upon:
  - Click ***sell the vehicle*** link: Call the **sales order** task.
  - Click ***Return to Main Menu*** button: Call the **Display Main Menu** task.

Login

Abstract Code

- User click ***login*** button on the **Main Menu screen**
- Display **Login form** Screen
- User enters username, password input fields.
- If data validation is successful for both username and password, then:
    - When ***Enter*** button is clicked:
        - If User record is found but password not match **User**.password:
            - Go back to **Login form**, with error message.
        - else:
            - Find the User using combination of User.username and User.password
            - Find the User.Type of the User and Store login information as session variable '$UserType'
            - Call the Display Main Menu task with '$UserType'.
        - Else username and password input fields are invalid, display Login form, with error message.

Sales order



Abstract Code:

- user click ***sell the vehicle*** on **Detail Page**
- **Sales order Form** is Displayed.
- user will fill the customer profile inputs field showed on **Sales order Form**:
    - if customer is an individual: fill their **first** and **last names**, along with their **driver's license number**
    - if customer is a business: fill the business' **tax identification number** and **business name**, along with the **name of a primary contact** and their **title**
- user will fill the transaction detail field including: **Vehicle's VIN**, **sold price**, **sold date**
- upon:
    - click the ***lookup*** button on **Sales order Form**: run the **lookup customer** task by query customer with **driver's license number or tax identification number**
        - if a customer is not found:
            - Read customer profile input fields, and call **Add customer** task.
    - click the confirm sale button on **Sales order Form**: run the **confirm sale** task by query vehicle's invoice price by reading and using **Vehicle's VIN**
        - if **sold price** is less than or equal to 95% of **Vehicle**.invoiceprice:
            - Display error massage of rejecting sale.
        - Otherwise, insert new **sale** instance with those values including **sold price**, **sold date**, customer **driver's license number** or **tax identification number**, **Vehicle's VIN**, **SalesPerson's Name.** Display a success message.
    - Click ***Return to Main Menu*** button: Call the **Display Main Menu** task.

Repair

Abstract Code

- User click ***repair form*** button on the **Main Menu screen**
- Partial **repair form** is displayed.
- User will fill the **VIN** input field.
- If data validation is successful for **VIN**, then:
    - When ***Enter*** button is clicked: run **search vin** task.
        - If the **VIN** does not match a **Vehicle**.VIN in the database:
            - Display an error message
        - Otherwise, the rest of the **repair form** will be displayed:
            - Run **View Vehicle Details** task, the results will be displayed on **repair form screen**
            - Check if the **Vehicle** is associated with a **repair** order.
                - If no repairs are open for the **Vehicle**:
                    - display **add repair button**
                    - run **add repair** task after user click ***add repair*** button
                        - Insert new **repair** order instance with **odometer reading** input field filled by user
                        - click the ***lookup*** button on **Sales order Form**: run the **lookup customer** task by query customer with **driver's license number or tax identification number**
                            - if a customer is not found:

- Read customer profile input fields, and call **Add customer** task. add the customer to the system for repair order
  - otherwise, add the customer to the system for repair order
- User are allowed to fill the inputs field of **labor charge** and **parts**:
  - input field **quantity**, **vendor**, **part number**, **price** can be filled by user.
  - If Data is valid and user click ***add parts*** button, insert new parts instance with those values with the service writer's name, display a success message, Otherwise, display an appropriate error message. run **add part task to** add the part to current **repair** order
  - if ***add labor charge*** button is clicked, add the labor charge to current **repair** order
- Otherwise, upon:
  - click ***updating labor charges*** button: read input field labor charges and run **update repair** task
  - click ***adding parts*** button: read input field for parts and run **add part** task
  - click ***completing*** button: add **repair**.completiondate to current **repair** as current date
- Click ***Return to Main Menu*** button: Call the **Display Main Menu** task.

Sales by Color

Abstract Code

- Called from the **Main Menu screen**
- Query for information about each **Sale**, find the **Sale**.PurchaseDate and corrsponding sold **Vehicle**.color using **Vehicle**.VIN. **Sale** and **Vehicle** are related by **transfers ownership of** relationship. Count the number of **Sale** based on the different periods (**Sale**.PurchaseDate) and group the count by different colors. Put count and color data in a table:
  - Each color is one row.
  - Columns are the count of sales.
  - Columns including sales in the previous 30 days, sales in the previous year, sales overall time.
  - If a color does not have any sales, it is shown with a value of "0".
- Display the table in the **Sales by Color report**
- Click ***Return to Main Menu*** button: Call the **Display Main Menu** task.

Sales by Type

Abstract Code

- Called from the **Main Menu screen**
- Query for information about each **Sale**, find the **Sale**.PurchaseDate and corrsponding sold **Vehicle**.type using **Vehicle**.VIN. **Sale** and **Vehicle** are related by **transfers ownership of** relationship. Count the number of **Sale** based on the different periods (**Sale**.PurchaseDate) and group the count by different types. Put count and type data in a table:
    - Each type is one row.
    - Columns are the count of sales.
    - Columns including sales in the previous 30 days, sales in the previous year, sales overall time.
    - If a type does not have any sales, it is shown with a value of "0".
- Display the table in the **Sales by Type report**
- Click ***Return to Main Menu*** button: Call the **Display Main Menu** task.

Sales by Manufacturer

Abstract Code

- Called from the **Main Menu screen**
- Query for information about each **Sale**, find the **Sale**.PurchaseDate and corrsponding sold **Vehicle**.Manufacturer using **Vehicle**.VIN. **Sale** and **Vehicle** are related by **transfers ownership of** relationship. Count the number of **Sale** based on the different periods (**Sale**.PurchaseDate) and group the count by different Manufacturers. Put count and Manufacturer data in a table:
    - Each Manufacturer is one row.
    - Columns are the count of sales.
    - Columns including sales in the previous 30 days, sales in the previous year, sales overall time.
    - If a type does not have any sales, it will not be put on the table.
- Display the table in the **Sales by Type report**
- Click ***Return to Main Menu*** button: Call the **Display Main Menu** task.

Gross Customer Income

Abstract Code

- Called from the **Main Menu screen**
- Query for information about all **Sale** and **Repair**, group **Sale** and **Repair** the by **Customer**'s driver's license number or tax identification number. And sum **Sale**.soldprice and **Repair** total cost as gross income for each **Customer** ID. Both **Sale** and **Repair** has relationship with **Customer**. Sort Customer ID by gross income and keep the largest 15 one. Find and Place all following data for each one of 15 Customer in a list:
        - **Customer**'s name
        - The date of the first sale or repair start date
        - the date of the most recent sale or repair start date
        - The number of sales
        - The number of repairs
        - the Gross income
    - The list of **customer**s will be by gross income descending and by last sale/repair start date descending.
    - Display the list in the **Gross Customer Income report**
    - User are able to select one ***customer's name*** in the list. If User click one of the ***customer's name***: Jump to **View Drill-Down** task with selected **Customer.** and their **sale** and **repair** which can be get from the table that group **sale** and **repair** by customer.
- Click ***Return to Main Menu*** button: Call the **Display Main Menu** task.

View Drill-Down Customers

Abstract Code

- User selected a customers from the list on the **Gross Customer Income report**
- Retreve the Customer ID, Sales, repairs from the **Gross Customer Income** task.
- Find and place all following data in a list, each row is for one **Sale**:
    - **Sale**.SoldDate
    - **Sale**.SoldPrice
    - Find the **Vehicle** associated with **Sale**, and get the **Vehicle**.VIN
    - **Vehicle**.manufactuer
    - **Vehicle**.model
    - **Sale**.salespersonname
- The listing should be sorted by sale date descending and VIN ascending
- Display the list in the **section for vehicle sales** on the **Drill-Down Screen**
- Find and place all following data in a list, each row is for one **Repair**:
    - **repair**.startdate
    - **repair**.completedate if available
    - Find the **Vehicle** associated with **Repair**, and get the **Vehicle**.VIN
    - **repair.**odometer
    - **repair**.labor cost
    - parts cost
    - total cost
    - the service writer who opened the repair
- This listing should be sorted by start date descending, end date descending, and VIN ascending; however, any incomplete repairs should be listed before completed ones with the same sorting criteria.
- Display the list in the **section for repairs** on the **Drill-Down Screen**
- Click ***Return to Main Menu*** button: Call the **Display Main Menu** task.

Repairs by Manufacturer/Type/Model

Abstract Code

- Called from the **Main Menu screen**
- Query for information about each **Repair**, Count the number of **Repair**, the sum of all parts cost, the sum of all labor cost, and the sum of total repair costs, including any repairs in progress for each **Vehicle**.Manufacturer. The **Vehicle**.Manufacturer is found by using **Vehicle**.VIN associate with each **repair**.
- Populate these data in a list where each row is for one **Vehicle**.Manufacturer, Manufacturers whose vehicles do not have any repairs should be listed on this list, and the list should be sorted by manufacturer name ascending.
- Display the list on **Repairs by Manufacturer/Type/Model screen**
- Users are able to select one manufacturer's name from the list
    - Run **Drill-down** task with the manufacturer's name

View Drill-Down Manufacturer

Abstract Code

- User select a manufacturer from the list on the **Repairs by Manufacturer/Type/Model screen**
- Retrieve the manufacturer, **Vehicle**, **repairs** from the **Repairs by Manufacturer/Type/Model** task.
- Find and place all following data in a list, each row is for one **Vehicle**.Type:
    - repair count
    - parts costs
    - labor costs
    - total costs
- Find and place all following data in a list, each row is for one **Vehicle**.model:
    - repair count
    - parts costs
    - labor costs
    - total costs
- Lists are sorted by repair count descending(by vehicle type sorted first, and then detail rows sorted).
- Display the list on the **Drill-Down Screen**
- Click ***Return to Main Menu*** button: Call the **Display Main Menu** task.

Below Cost Sales

Abstract Code

- Called from the **Main Menu screen**
- Query for information about each **Sale**, find the **Sale** that **Sale**.invoicepprice > **Sale**.soldprice. Find and place all following data in a list, each row is for one **Sale**:
    - **Sale**.completedate
    - **Sale**.invoice price
    - **Sale**.sold price
    - sold price/invoice price ratio as a percentage
    - **customer**.name. Retrieve customer's name by **customer** ID associate with **Sale**.
    - **Sale**.salesperson's name
- For a sale whose ratio is less than or equal to 95%, the background of that row should be highlighted red. Sales should be listed by sales date descending and ratio descending.
- Display the table in the **Below Cost Sales report**
- Click ***Return to Main Menu*** button: Call the **Display Main Menu** task.

Average Time in Inventory

Abstract Code

- Called from the **Main Menu screen**
- Query for information about each **Sale**, find the **Sale**.complete date. Find the Vehicle.Date using VehicleVIN associated with Sale. Calculate the difference between **Sale**.compolete date and **Vehicle**.Date as the amount of time a vehicle remains in inventory group by Vehicle.Type. Calculate and Put the average amount of time a vehicle remains in inventory in a list, each row is for one Vehicle.Type
- If a **vehicle**.type has no sales history, the report should display "N/A" for that **vehicle**.type.
- Display the table in the **Average Time in Inventory report**
- Click ***Return to Main Menu*** button: Call the **Display Main Menu** task.

Parts Statistics

Abstract Code

- Called from the **Main Menu screen**
- Query for information about each **Part**, find the **Part**.price and **Part**.quantity group by **Part**.vendorname. Calculate the total cost and total quantity of part for each **Part**.vendorname. Then put the vendor's name, the number of parts supplied by that vendor, and the total dollar amount spent on parts in a list.
- Display the list in the **Parts Statistics report**
- Click ***Return to Main Menu*** button: Call the **Display Main Menu** task.

Monthly Sales

Abstract Code

- Called from the **Main Menu screen**
- Query for information about **Sale**.
    - Group the Sale by year and month based on **Sale**.Date then calculate the count and sum for each group.
    - create a list which has:
        - the total number of vehicles sold, the total sales income, the total net income (calculate by using soldprice - invoice price), and the sold price/invoice price ratio as a percentage (such as 125%) for each year and month based on **Sale.**Date.
    - If a year or month does not have sales data, it can be excluded from this report.
    - When the ratio for a month is greater than or equal to 125%, its row should be highlighted with a green background. If the ratio is less than or equal to 110%, it should be highlighted with a yellow background.
    - The results will be ordered by year and month descending, with the most recent year and month as the first result.
    - Display the list on **Monthly Sales screen**
- Users are able to select one manufacturer's name from the list:
    - Run **Drill-down** task with the manufacturer's name

Monthly Sales Drill-Down

Abstract Code

- User select a month/year from the list on the **Monthly Sales screen**
- Retrieve the **sale**s in select month/year group from the **Monthly Sales** task.
- Group the **sale** by **sale**.salesperson then calculates the total vehicles and total sales for each **sale**.salesperson. sort the Sale.salesperson by total vehicles descending followed by total sales descending. The first **sale**.salesperson is the top salesperson
- Display the top salesperson on the **Monthly Sales Drill-Down Screen**
- Click ***Return to Main Menu*** button: Call the **Display Main Menu** task.