



Jaunty Jalopies Database Design

Table of Content

Table of Content	1
Data Types	2
Assumptions	4
Business Logic Constraints	4
Task Decomposition with Abstract Code	6
Considerations	6
Access	6
Login	8
View Sale Report	10
View Repair Report.....	13
View Vehicle Report	15
View Customer Report	16
Add Vehicle	18
Search Vehicle.....	19
View Vehicle Detail.....	21
Add/Update Repair	23
Add Vehicle Sale	26
Add/Update Manufacturer	28

Data Types

Logged-in User

Attribute	Data Type	Nullable
Username	STRING	NOT NULL
Password	STRING	NOT NULL
First Name	STRING	NOT NULL
Last Name	STRING	NOT NULL

Vehicle

Attribute	Data Type	Nullable
VIN	STRING	NOT NULL
Vehicle Type	STRING	NOT NULL
Model Name	STRING	NOT NULL
Model Year	YEAR	NOT NULL
Invoice Price	DECIMAL (19,2)	NOT NULL
Added Date	DATETIME	NOT NULL
Is Sold	BOOLEAN	NOT NULL
Description	STRING	NULL
Color	LIST<STRING>	NOT NULL
List Price	DECIMAL (19,2)	NOT NULL
Manufacturer Name	STRING	NOT NULL

Repair

Attribute	Data Type	Nullable
Total Cost	DECIMAL (19,2)	NOT NULL
Labor Charge	DECIMAL (19,2)	NOT NULL
Completion Date	DATETIME	NULL
Start Date	DATETIME	NOT NULL
Odometer Readout	INT	NOT NULL
Description	STRING	Null

Part

Attribute	Data Type	Nullable
Part Number	STRING	NULL
Price	DECIMAL (19,2)	NULL
Vendor Name	STRING	NULL

Customer

Attribute	Data Type	Nullable
State	STRING	NOT NULL
Post Code	STRING	NOT NULL
Street Address	STRING	NOT NULL
City	STRING	NOT NULL
Email	STRING	NULL

Phone Number	STRING	NOT NULL
--------------	--------	----------

Individual-Specific

Attribute	Data Type	Nullable
First Name	STRING	NOT NULL
Last Name	STRING	NOT NULL
Driver License Number	STRING	NOT NULL

Business-Specific

Attribute	Data Type	Nullable
Primary Contact Name	STRING	NOT NULL
Primary Contact Title	STRING	NOT NULL
Business Name	STRING	NOT NULL
Tax Identification Number	STRING	NOT NULL

Sale Transaction

Attribute	Data Type	Nullable
Sold Price	DECIMAL (19,2)	NOT NULL
Purchase Date	DATETIME	NOT NULL

Car-Specific

Attribute	Data Type	Nullable
Number of Doors	INT	NOT NULL

Convertible-Specific

Attribute	Data Type	Nullable
Roof Type	STRING	NOT NULL
Back Seat Count	INT	NOT NULL

Truck-Specific

Attribute	Data Type	Nullable
Cargo Capacity	INT	NOT NULL
Cargo Cover Type	STRING	NULL
Number of Rear Axis	INT	NOT NULL

Van/minivan-Specific

Attribute	Data Type	Nullable
Has Driver Side Backdoor	BOOLEAN	NOT NULL

SUV-Specific

Attribute	Data Type	Nullable
Drivetrain Type	STRING	NOT NULL
Number of Cupholders	INT	NOT NULL

Assumptions

Only one Logged-in User at any point of time.

A mother task is needed if the sub-tasks need to be sequenced, all executed together as an atomic unit.

One year has 365 days.

One Vehicle is bought by one Customer.

Within each Repair, the Part Number is unique.

VIN refers to Vehicle Identification Number.

Driver License Number is unique.

Tax Identification number is unique.

Driver License Number and Tax Identification number have different formats.

At the initial phase of using this database, there will be minimal number of existing customers.

Business Logic Constraints

Log in

The system should verify that all fields are filled in. More specifically, the Username and the Password should match with the system record for a successful login.

Search Vehicle

The results of Vehicle are sorted by VIN are in ascending order.

By filtering the Vehicle, Manufacturer Name is a single selection.

System should indicate whether the keyword matches the Description in the search results.

Customer

Email address is optional when adding a new Customer.

Add Vehicle

Model Year must be smaller or equal to the Current Year +1.

Model Year must be at least 4 digits, to include century digits.

By default, the Is Sold attribute is False.

The invoice price must include dollars and cents.

Description is optional.

Cargo Cover type is optional.

Sales

The List Price is equal to 125% of the Invoice Price.

Look up and add Customer are available only when performing Sales Transaction and Repair.

Adding Customer is visible when there is no Customer record in the system.

The Sold Price should be more than or equal to 95% of the Invoice Price. If Username is Roland Around/Owner, the Sold Price can be less than or equal to 95% of the Invoice Price.

The Purchase Date should not be earlier than the date a Vehicle enters the inventory.

If the number of Color is over 1, the number of vehicles that have only one Color should exclude vehicles with multiple Colors.

If a Manufacturer does not have any Sales Transaction, it should be excluded from the report.

Is Sold attribute of Vehicle should be true if this Vehicle has been sold/ a Sale Transaction involves this Vehicle.

Repair

On the same date, the number of Repairs for the same vehicle is 0 or 1.

The Repair Customer may or may not be the Customer who purchased the vehicle.

The Start Date should not be bigger (later) than Completion Date for the same vehicle.

A repair must be completed before a new one can be started.

If the Vehicle has an unfinished Repair, then the Repair for only allow for updating Labor Charges, adding Parts, or completing the Repair.

The Completion Date is null until completing the Repair is done on the Repair form.

The Labor Charge added should be more than or equal to 0, including cents. If Username is Roland Around/Owner, the Labor Charge added can be lower than the previous value.

Entering Labor Charges and adding Parts are visible when there is Repair.

Task Decomposition with Abstract Code

Notations:

Bold Under Line: Task Definition

Bold: Task Call

Bold Italics: Button names

Italics: Form Input Fields

Considerations

View Repair Form is not independently accessible, it is part of Vehicle Detail Page.

Search Customer/ Add Customer is not independently accessible, it is part of Add Sale/ Add Repair form.

Access

Anonymous User:

- View Vehicle Detail Page
- Search Vehicle

Logged-in Users (have privilege access):

- Login
- Search Vehicle
- View Vehicle Detail Page
- View Sale Report
- View Repair Report
- View Vehicle Report
- View Customer Report
- Add Vehicle
- Add/Update Repairs
- Add Vehicle Sale
- Add/Update Manufacturer

The details of Privilege Access Users are as below:

Inventory Clerk:

- Login
- Search Vehicle
- View Vehicle Detail
- Add/Update Manufacturer
- Add Vehicle

Salesperson:

- Login
- Search Vehicle
- View Vehicle Detail
- Add Vehicle Sales

Service Writer:

- Login
- Search Vehicle
- View Vehicle Detail
- Add/Update Repairs

Manager:

- Login
- Search Vehicle
- View Vehicle Detail
- View Report

Roland Around/Owner:

- Login
- Search Vehicle
- View Vehicle Detail
- Add/Update Manufacturer
- Add Vehicle
- Add Sale
- Add/Update Repairs
- View Report

[Login](#)

Task Decomposition

Lock Types:

Logged-in User table, one Read only lock

Number of Locks:

Single

Enabling Conditions:

None

Frequency:

Around 100 logins per day

Consistency (ACID):

Not critical, order is not critical.

Subtasks:

Mother Task is not needed.

No decomposition needed.

A single atomic action.



Abstract Code

- Logged-in User enters *username* ('\$Username'), *password* ('\$Password') input fields in **Login** form.
- If data validation is successful for both *username* and *password* input fields, then:
 - When **Login** button is clicked:
 - If user record exists, but user Password != '\$Password':
 - Go back to **Login** form, throw error message "Wrong password!".
 - If Logged-in User is Inventory Clerk
 - Show, **Search Vehicle by VIN**, **Add/Update Manufacturer**, **Add Vehicle** and **Log out** tabs.
 - Upon:
 - Click **Search Vehicle by VIN** button – Jump to the **Search Vehicle** task.
 - Click **Add Vehicle** button – Jump to the **Add Vehicle** task.
 - Click **Add/Update Manufacturer** button – Jump to the **Add/Update Manufacturer** task.
 - Click **Log out** button – Invalidate login session and go back to the **Login** form.
 - If Logged-in User is Saleperson
 - Show **Search Vehicle by VIN**, **Add/Update Sale** and **Log out** tabs.

- Upon:
 - Click **Search Vehicle by VIN** button – Jump to the **Search Vehicle** task.
 - Click **Add/Update Sale** button – Jump to the **Add/Update Sale** task.
 - Click **Log out** button – Invalidate login session and go back to the **Login** form.
- If Logged-in User is Service Writer
 - Show, **Search Vehicle by VIN**, **Add/Update Repair Form** and **Log out** tabs.
 - Upon:
 - Click **Search Vehicle by VIN** button – Jump to the **Search Vehicle** task.
 - Click **View Repair Form** button – Jump to the **Add/Update Repair Form** task.
 - Click **Log out** button – Invalidate login session and go back to the **Login** form.
- If Logged-in User is Manager
 - Show **Search Vehicle by VIN**, **View Report** and **Log out** tabs.
 - Upon:
 - Click **Search Vehicle by VIN** button – Jump to the **Search Vehicle** task.
 - Click **View Report** button – Jump to the **View Report** task.
 - Click **Log out** button – Invalidate login session and go back to the **Login** form.
- If Logged-in User is Roland Around/Owner
 - Show, **Search Vehicle by VIN**, **View Report**, **Add Vehicle**, **Add/Update Sale** and **Log out** tabs.
 - Upon:
 - Click **Search Vehicle by VIN** button – Jump to the **Search Vehicle** task.
 - Click **Add Vehicle** button – Jump to the **Add Vehicle** task.
 - Click **Add/Update Sale** button – Jump to the **Add/Update Sale** task.
 - Click **View Report** button – Jump to the **View Report** task.
 - Click **Log out** button – Invalidate login session and go back to the **Login** form.
- Else *username* and *password* input fields are both invalid, display **Login** form, with error message “no such user”.

View Sale Report

Task Decomposition

Lock Types:

Lookup Read only Vehicle

Lookup Read only Sale Transaction

Lookup Read only Manufacturer

Lookup Customer, Read only lock

Lookup Logged-in User, Read only lock

Number of Locks:

5

Sale Transaction is a weak entity, Vehicle entity must be accessed together with Sale Transaction.

Enabling Conditions:

Trigger by Logged-in Users - **view report**

Frequency:

All have same frequency, around 20 times per day.

Consistency (ACID):

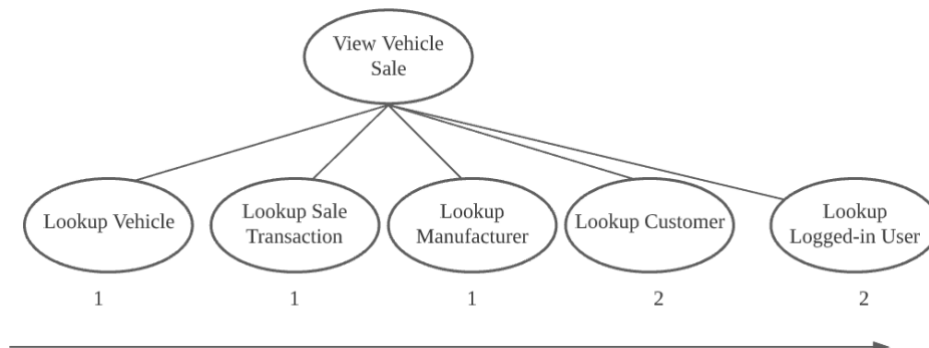
Not critical, order is not critical.

Subtasks:

All tasks must be done but can be done in parallel.

Mother Task is needed.

Decomposition is needed.



Abstract Code

- The Managers and Roland Around/Owner click **View Report** button.
 - If access to **Sales by Color Report**:
 - Get Distinct Colors in **Vehicle** table where Color is a single value
 - Sales date ≤ 30 days
 - Inner join **Vehicle** table with **Sale Transactions** table with condition that:
 - sales date in **Sale Transactions** ≤ 30 days before today and
 - VIN in **Sale Transactions** Table = VIN in **Vehicle** and
 - Color in **Vehicle** is single value

- Count number of vehicles grouping by Color. Display 0 if the count is 0, for that color.
- Sale Transaction Purchase Date < 1 year
 - Inner join Vehicle table with Sale Transactions table with condition that:
 - Purchase Date in Sale Transaction Purchase Date <= 1 year before today and
 - VIN in Sales = VIN in Vehicle and
 - Color in Vehicle is a single value.
 - Count number of vehicles grouping by Color. Display 0 if the count is 0, for that color.
- Sales of overall
 - Inner join Vehicle table with Sale Transactions table with condition that:
 - VIN in Sale Transaction = VIN in Vehicle and
 - Color in Vehicle is a single value.
 - Count number of vehicles grouping by Color. . Display 0 if the count is 0, for that color.
 -
- **Display the count for each color, in 3 different time ranges.**
- If access to ***Sales by Type Report:***
 - Get Distinct type in Vehicle table
 - Sales date <= 30 days
 - Inner join Vehicle table with Sale Transactions table with condition that:
 - sales date in Sales Form <= 30 days before today and
 - VIN in Sale Transactions = VIN in Vehicle and
 - Count number of vehicles grouping by type. Display 0 if the count is 0.
 - Sales date <= 365 days
 - Inner join Vehicle table with Sale Transactions table with condition that:
 - sales date in Sale Transactions <= 365 days before today and
 - VIN in Sale Transactions = VIN in Vehicle and
 - Count number of vehicles grouping by type Display 0 if the count is 0.
 - Sales over all
 - Inner join Vehicle table with Sale Transactions table with condition that:
 - sales date in Sales Form <= 365 days before today and
 - VIN in Sale Transactions = VIN in Vehicle and
 - type in Vehicle is not “multiple”
 - Count number of vehicles grouping by type Display 0 if the count is 0.

- **Sort by Sale Transaction Purchase Date.**
 - **Display the count for each type, in 3 different time ranges.**
- If access to ***Sales by Manufacturer Report***:
- Get Distinct manufacturer in Vehicle table
 - Sales date ≤ 30 days
 - Inner join Vehicle table with Sale Transactions table with condition that:
 - sales date in Sale Transactions ≤ 30 days before today and
 - VIN in Sale Transactions = VIN in Vehicle and
 - Count number of vehicles grouping by manufacturer. Exclude that manufacturer if the count is 0.
 - Sales date ≤ 365 days
 - Inner join Vehicle table with Sale Transactions table with condition that:
 - sales date in Sale Transactions ≤ 365 days before today and
 - VIN in Sale Transactions = VIN in Vehicle and
 - Count number of vehicles grouping by manufacturer. Exclude that manufacturer if the count is 0.
 - Sales overall
 - Inner join Vehicle table with Sale Transactions table with condition that:
 - VIN in Sale Transactions = VIN in Vehicle and
 - Count number of vehicles grouping by manufacturer. Exclude that manufacturer if the count is 0.
 - **Display the count for each manufacturer, in 3 different time ranges.**
- If access to ***Monthly Sales Report***:
- **Load Summary Page**
 - Fetch all data from Sale Transactions Table.
 - Count the number of items, save as a variable totalNumberOfVehiclesSold. Do this for each month and for each year.
 - Sum up the Sold Price for Sale Transactions, save as a variable TotalSaleIncome. Do this for each month and for each year.
 - Find the Vehicle using VIN (from Sale Transaction), and get Invoice Price. Sum up Invoice price for each vehicle sold in the time period, save as TotalInvoicePrice. Do this for each month and for each year.
 - Arrange the above data in each year and month. For each time period, calculate NetIncome by using TotalSaleIncome – TotalInvoicePrice. Also calculate TotalSaleIncome / TotalInvoicePrice for each month and year, as SIPercentage.

- Arrange and order by year and month descending and Display the above data by year and month.
- If the SIPercentage is $\geq 125\%$, highlight with green background.
- If the SIPercentage is $\geq 110\%$, highlight with yellow background.
- Show **Top Sales Report** Button for each year and month
- **When the Top Sales Report is pushed for that year/Month, Load the sales data for selected year and month. For each time period, do the following:**
 - Fetch all the Sale Transactions.
 - Groupby SalesPerson username.
 - Sum up the number of items and Sold Price under each Group (by SalePerson)
 - Sort the group first by number of items, and then by total SoldPrice.
 - Display the SalePerson's First Name and Last Name, alongside with the calculated totalNumberOfItem and TotalSoldPrice for that time period.

View Repair Report

Task Decomposition

Lock Types:

Lookup read-only Repair

Lookup read-only Vehicle

Lookup read-only Manufacturer

Lookup read-only Part

Number of Locks: 4

Enabling Conditions:

Trigger by Logged-in Users- **view report**

Frequency:

All have same frequency, around 20 times per day.

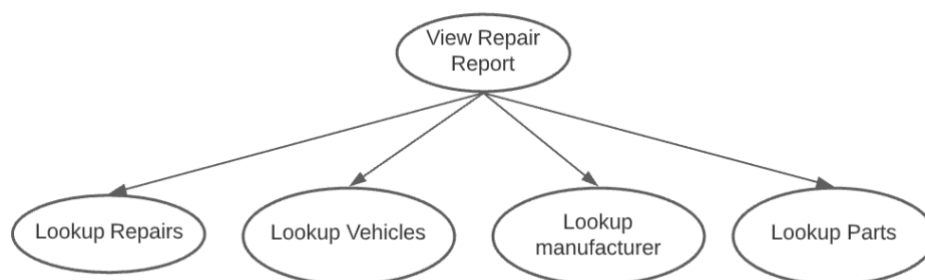
Consistency (ACID):

Not critical, order is not critical.

Subtasks:

All tasks must be done and can be done in parallel.

Mother Task is needed. Decomposition is needed.



Abstract Code

- The Managers and Roland Around/Owner click **View Report** button.
 - If they want to see **Repairs by Manufacturer/Type/Model Reports**:
 - Fetch all Repairs.
 - Each Repair is tied with a Vehicle, group the Repairs by Vehicle Manufacturer Name, regardless of whether Repair Completion Date is null or not.
 - For each Manufacturer, do the following:
 - Count the number of Repairs;
 - Adding up all the Parts Cost for all of the Repairs for this Manufacturer.
 - Adding up all the Labor Cost for all of the Repairs for this Manufacturer.
 - Adding up the above two results to get the Total Repair Costs for this Manufacturer.
 - Load all the Manufacturer Names. For Manufacturer Name not present in the result, add them to the result with Number of Repair 0 and Cost 0.
 - Display the calculated information above for each manufacturer in manufacturer name ascending.
 - For every Manufacturer that have Repairs: User can click on **More Detail** button to **load drill-down report** (sorted by repair count descending by Vehicle Type first then models)
 - Search for Repairs filtering only for this Manufacturer, Group by Vehicle Type.
 - For each Vehicle Type that have Repairs:
 - Count the number of Repairs and Display it.
 - Summing up all the total Parts Cost and display It.
 - Summing up all the Labor Cost and display it.
 - Adding up the above two results, save as total Total Cost and display it.
 - Sorted by Repair Count descending.
 - **Group the Repairs under this Vehicle Type by Model.**
 - For each model that have repairs in that Vehicle Type:
 - Summing up all the total Parts Cost and display It.
 - Summing up all the Labor Cost and display it.
 - Adding up the above two results, save as total Total Cost and display it.
 - Sorted by Repair Count descending.
 -
 - If access to **Parts Statistics Report**:

- Fetch all the Parts by using VIN, Start Date and Part Number.
- Group the Parts by Vendor Name.
- For each group, do the following:
 - Summing up the Quantity for all parts under this Vendor.
 - Calculate Quantity x Price for each part, sum up for each part.
 - Display the above information, and Vendor Name.

View Vehicle Report

Task Decomposition

Lock Types:

Lookup read-only Vehicle

Lookup read-only Sale Transaction

Lookup read-only Logged-in User

Number of Locks:

2 for Average Time in Inventory Report

3 for Below Cost Sales Report

Enabling Conditions:

Trigger by Logged-in Users- **view report**

Frequency:

All have same frequency, around 20 times per day.

Consistency (ACID):

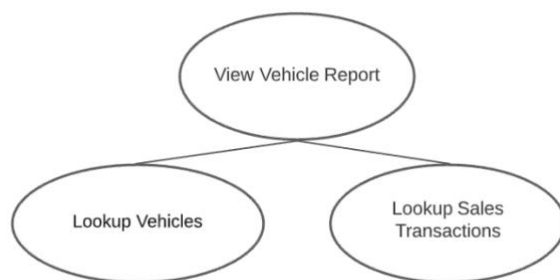
Not critical, order is not critical.

Subtasks:

All tasks must be done and can be done in parallel.

Mother Task is needed.

Decomposition is needed.



Abstract Code

- The Managers and Roland Around/Owner click **View Report** button.
 - If access to **Average Time in Inventory Reports**:
 - Fetch all the Vehicles from Vehicle table.
 - Group by Vehicle Type.
 - For each group (items under each Vehicle Type), do the following:

- Find the Sale Transaction for that Vehicle using VIN. Calculate TimeInInventory as the difference between Vehicle Purchase Date and Added Date)
- Take the average for all the calculated TimeInInventory for this group.
- **Display Average Time in Inventory information for all groups.**
- If access to ***Below Cost Sales Report***:
 - Fetch all the Sale Transactions, which has a VIN and tied with a Vehicle. Lookup **each Vehicle sold using VIN**, Calculate Sold Price/Invoice Price as a percentage and display percentage
 - **If the ratio ≥ 1.0 , skip this and do not display anything.**
 - **Else:**
 - Display the purchase date, the Invoice Price, the Sold Price, and the calculated ratio.
 - **Lookup the customer**, and display the customer's name (first/last for individuals or business name for businesses)
 - **Lookup the salesperson**, display name of the Salesperson for the Sale Transaction
 - If ratio $\leq 95\%$, the background of that row should be highlighted red
 - Sort the list by sales date descending and ratio descending

[View Customer Report](#)

Task Decomposition

Lock Types:

Look up read-only Customer

Lookup read-only Repair

Lookup read-only Sale Transaction

Lookup read-only Vehicle

Lookup read-only Manufacturer

Lookup read-only Logged-in User

Number of Locks:

6

Enabling Conditions:

Trigger by Logged-in Users- ***view report***

Frequency:

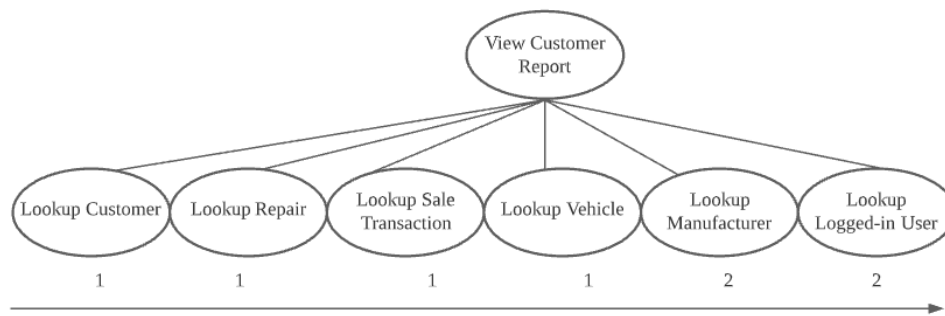
All have same frequency, around 20 times per day.

Consistency (ACID):

Not critical, order is not critical.

Subtasks:

All tasks must be done and can be done in parallel. Mother Task is needed. Decomposition is needed.



Abstract Code

- The Managers and Roland Around/Owner click **View Report** button.
 - Display Gross Customer Income report:
 - **Load the top 15 Customers**, the gross income received from them via Vehicle sales and/or Repairs (including any Repairs in progress)
 - Fetch all Sale Transactions
 - Group the Sale Transactions by Customer's identifier (either Diver License Number or Tax Identification Number)
 - Sum up the total Sold Price for each customer. Save this view as GrossSale.
 - Fetch all Repairs
 - Group the Repairs by Customer's identifier
 - Sum up of the total cost (by adding up labor charge and parts costs for each repair). Save this view as GrossRepair.
 - Both views share the same common Customer identifier.
 - Combine GrossSale and GrossRepair, summing up the GrossSale value and GrossRepair value for each customer, as GrossCustomerIncome for each customer. Save this view including all customers as GrossCustomer. Sort the GrossCustomer by the GrossCustomerIncome descending, and by last Sale Transaction Purchase Date or Repair Start Date.
 - Display the top 15 items from GrossCustomer for the following attributes, for each one of them:
 - **Find** and Display the Customer's Name (first/last for individuals or business name for business)
 - **Find the** the date of the first Sale Transaction Purchase Date and Repair Start date. Display the earlier one.
 - **Find** the most recent Sale Transaction Date and Repair Start Date. Display the most recent one.
 - Count the number of sales, the number of repairs, and the GrossCustomerIncome
 - Show "More Detail" Button near each of the customer Name
 - When *More Detail* Button is pushed:
 - **Load Vehicle sales section:**

- Lookup Vehicle and Sale Transaction using VIN and Customer Identifier.
- For each Sale Transaction:
Display Sale Date, Sold Price, VIN, year, Lookup Manufacturer and display Manufacturer Name, Model and Lookup Logged-in User and display Salesperson Name.
- Sort Vehicle sale Transactions by sale date descending and VIN ascending.
- **Load repair section:**
 - Search for Repairs using VIN and Customer Identifier.
 - For each Repair, display Start date, End date (if the Repair is not finished, this should not display any value). Lookup Vehicle using VIN Display the VIN of the repaired Vehicle, the Odometer Reading, Parts Cost, Labor Cost, Total Cost, and Lookup the Logged-in User (the Service Writer) who opened the Repair and display First Name and Last Name
 - Sort Repairs by Start Date descending, End Date descending

Add Vehicle

Task Decomposition

Lock Types:

Vehicle

Read-only Manufacturer

Number of Locks:

2

Enabling Conditions:

Trigger by login as Inventory Clerk or Roland Around/Owner – **Add Vehicle**

Frequency:

All have the same frequency, around 10 times per day, sometimes might have a spike.

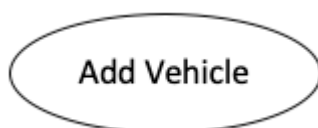
Consistency (ACID):

Consistency is critical. The newly added vehicle is made available immediately for sale.

Subtasks:

Mother Task is not needed to coordinate subtasks.

Decomposition is not needed.



Abstract Code

- Users who login as Inventory Clerk and Roland Around/Owner can click on “*Add Vehicle*” button to **add new Vehicles**:
 - **Load new vehicle form**, the form will wait for user input for all relevant fields such as *VIN, Vehicle Type, Model Name, Model Year, Invoice Price, Description (optional), List Price*, and type-specific fields such as *Number of doors* for Car, *Roof Type* and *Back Seat Count* for Convertible, *Cargo Capacity, Cargo Cover Type (optional)* and *Number of Rear Axis* for Truck, *Driver Side Backdoor* for van&minivan, *Drivertrain Type* and *Number of Cupholders* for SUV.
 - When user click **type, Show** a dropdown of list of Vehicle Types.
 - When user click **Manufacturer Name, Show** a dropdown of list of manufacturer names.
 - When user click **Color, Show** a dropdown of list of Color to add to the Color section.
 - If the user clicks **submit button** to **add new Vehicle**:
 - Validate all fields are not null, except for description field and cargo cover type.
 - Validate *\$Model Year* must be \leq current year + 1, and it must be a 4-digit integer, otherwise throw error message.
 - Validate *Invoice Price* must be a Float.
 - Capture today’s date as AddedDate.
 - Calculate *List Price* to be 125% of *Invoice Price*.
 - **Save data to the database**
 - **Redirect to View Vehicle detail page** for the added vehicle (Comment: This is a separate task, refer to **View Vehicle Detail Task**.)
 - If the user clicks **Cancel**:
 - The new vehicle form will be closed.

Search Vehicle

Task Decomposition

Lock Types:

Lookup read-only vehicle

Lookup read-only manufacturer

Number of Locks:

2

Enabling Conditions:

Triggered by Anonymous Users or privileged user by visiting the home page.

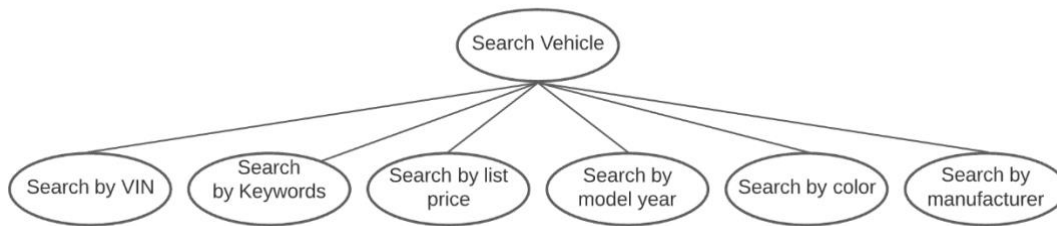
Frequency:

All have the same frequency, 1000 times a day.

Consistency (ACID):

Consistency is critical for all task

Subtasks: Mother Task is needed to coordinate subtasks. Decomposition is needed.



Abstract Code

- Anonymous User and privileged user land on the **Search Page**. **Load** the total number of vehicles available for purchase in the system (is Sold is False).
- For user who are privileged users, show additional **search by VIN** button.
- While **Search Button** not pushed, stay on current view.
- To gather the search criteria, when a button is clicked, do the following:
- If the user enters specified *VIN* (\$VIN):
 - Create a predicate: Vehicle VIN must be the same as \$VIN, used for later search
- If the user enters *Keyword* (\$keyword):
 - Create a predicate: keyword must be a substring or entire string of the following Vehicle attributes: manufacturer, model year, model name and description.
- If the user enters *List Price* (\$List Price):
 - Create a predicate: Vehicle list price must equal to \$List Price
- If the user clicks *manufacturer*:
 - **Fetch all manufacturers**, show dropdown for user selection.
 - Wait for user to click one of the options (\$Manufacturer)
 - Create a predicate: Vehicle manufacturer must equal to \$Manufacturer
- If the user clicks *model year*:
 - **Fetch all possible model years**, show dropdown for user selection.
 - Wait for user to click one of the options (\$ModelYear)
 - Create a predicate: Vehicle Model Year must equal to \$ ModelYear
- If the user clicks *color*:
 - **Fetch all possible colors**, show dropdown for user selection.
 - Wait for user to click one of the options (\$Color)
 - Create a predicate: Vehicle Color must equal to \$ Color
- After gathering the search criteria, When the **Search Button** is pushed:
 - For privileged users using lookup vehicle by VIN:
 - **Lookup the vehicle using VIN**, return the Vehicle regardless of Is Sold value.
 - For users who are either Anonymous User or privileged user, **apply all the predicates** created above. Search for the vehicles.
 - If the user is anonymous user, Inventory Clerk, Salesperson:
 - **Search Vehicle** for those whose Vehicle.IsSold is False, with the gathered conditions (predicates).
 - If the user is Service Writer, Manager, Owner:

- **Search Vehicles** regardless of the value of IsSold, with the gathered conditions (predicates).
 - If no vehicles meet the search criteria:
 - **Display an error message:** “Sorry, it looks like we don’t have that in stock!”.
- Once fetched all the Vehicles that met the predicates (criteria), On the **Search Result Page**:
 - If the user is anonymous:
 - Display VIN, Vehicle Type, Model Year, Model Name, Manufacturer, Color(s), List Price.
 - If the user is Manager/ Owner:
 - Show Additional Filter by *Sold Vehicles, Unsold Vehicles, All Vehicles*.
 - If User clicks *Sold Vehicles*:
 - Return Vehicles with Is Sold is ‘True’
 - If User clicks *Unsold Vehicles*:
 - Return Vehicles with Is Sold is ‘False’
 - If User clicks *All Vehicles*:
 - Return all Vehicles satisfy the predicates.
 - If the keyword matches the description, show X mark.
 - Results are sorted by VIN in ascending order.
 - If user click **sort button** on any attribute column, **view sorted table**, display the table sorted by the selected column.
 - If user click **filter button** on any attribute column, **view filtered table**, display the table filtered by the selected column.
 - If user click on an **individual vehicle VIN link**, it will **open a Vehicle Detail Page** of the selected vehicle:
 - **Detail Page** will **load vehicle details**, which includes the VIN, Vehicle Type, attributes for that Vehicle Type, Model Year, Model Name, Manufacturer, Color(s), List Price, and the Description for the selected result.

View Vehicle Detail

Task Decomposition

Lock Types:

Look up read-only Vehicle

Look up read-only Manufacturer

Look up read-only Customer

Look up read-only Sales Transaction details

Look up read-only Repair details

Look up read-only Logged-in User information

Number of Locks: 6

Enabling Conditions:

Triggered by Anonymous Users or privileged user by clicking **View Vehicle Detail**

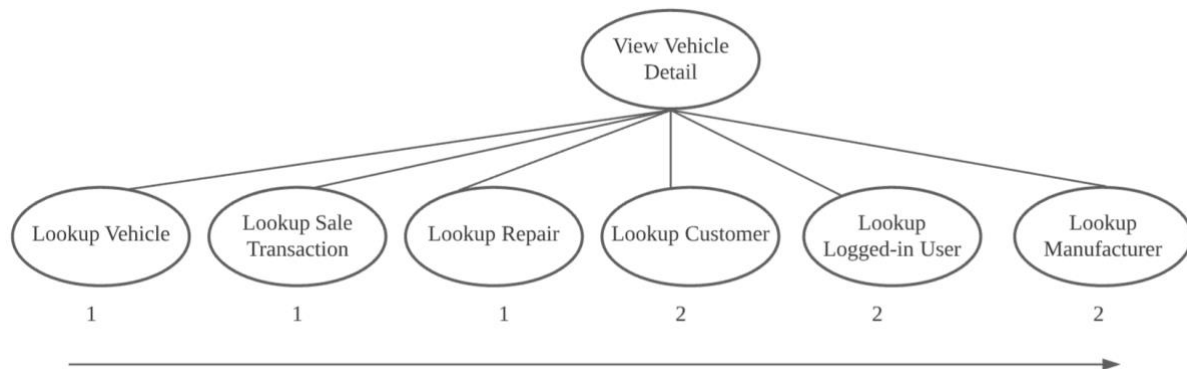
Frequency:

Share the similar frequency. 900 times a day.

Consistency (ACID):

It is important for vehicle because it should not return any vehicle already sold.

Order:



First Batch: Search for Vehicle OR Search for Sales Transaction OR Search for Repair, in any order.

Second Batch: Search for Customer, Logged-in User, Manufacturer, in any order

Subtasks:

Mother Task is needed, sequencing of different tasks is needed.

Abstract Code

- **Search the Vehicle** using the Vehicle VIN
- Select **Vehicle**, Click **View Vehicle Detail Page** for the selected **Vehicle**
 - Defined a Standard Display Set as VIN, Vehicle Type, attributes for that Vehicle Type, Model Year, Model Name, Manufacturer, Color(s), List Price, description.
 - If user is an Anonymous User: **Display** Standard Display Set
 - If user is an Inventory Clerk: **Display** Standard Display Set + Invoice Price.
 - If user is a Manager or Roland Around/Owner: **Display** the Standard Display Set + Invoice Price + Added Date.
 - **Find the Inventory Clerk** for this Vehicle, Display First Name and Last Name
 - If the user is sale person or owner, Show additional **Sell Vehicle** button, which will be directed to **Add Sale Page**.
- If the user is Manager or Roland Around/Owner:
 - **Find the Sale Transaction** for the Vehicle
 - If Sale Transaction cannot be found:
 - **Display Empty sale related information**
 - If Sale Transaction can be found,
 - Display Sold Price, Purchase Date
 - **Find the Sale Person**, display First Name and Last Name.
 - **Find the customers** for the Sale Transaction

- If the Customer is an Individual, display FirstName, Last Name, Phone Number, Email, Address (Street Address, City, State, Postal Code)
 - If the Customer is a Business, display Tax Identification Number, Business Name, Primary Contact Name, Primary Contact Title, Phone Number, Email, Address (Street Address, City, State, Postal Code)
- If the user is Manager or Roland Around/Owner, **Find the Repair** for the Vehicle
 - If Repair cannot be found, **Display empty Repair section.**
 - If Repair can be found
 - **List all Repairs**
 - For each Repair, **Display** Start date, End date, Labor Charges, Part Cost, and Total Cost (by summing all the labor charges and Part costs)
 - **Find the Service Writer** for each Repair
Display First Name and Last Name
 - **Find the Customer** for each Repair
 - If the Customer is an Individual
Display First Name and Last Name
 - If the Customer is a Business
Display Business Name

Add/Update Repair

Lock Types:

Lookup Repair

Write lock Add/update Repair

Look up Vehicle

Lookup Individual or Business

Write Individual or Business

Number of Locks:

5 for new Customer

4 for existing Customer

Enabling Conditions:

Trigger by Logged-in Users (Service Writer and Roland Around/Owner) – Add/Update Repair Button

Frequency:

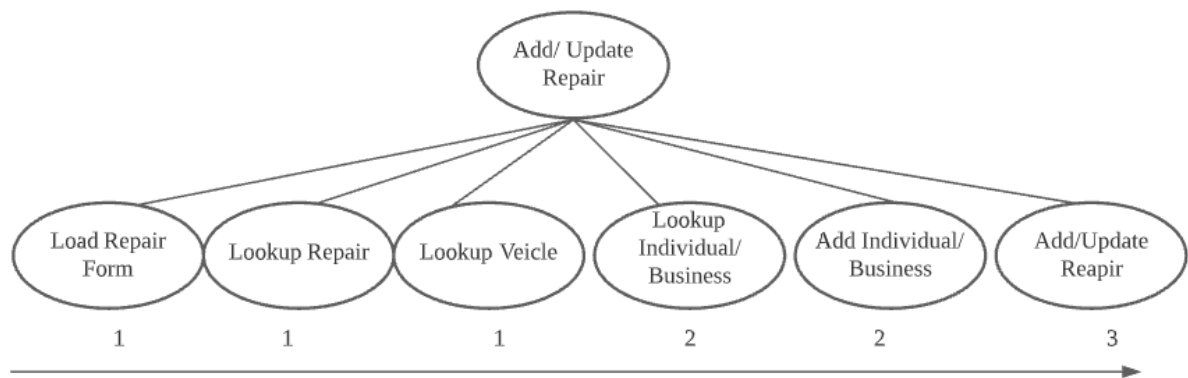
All have the same frequency. 20 times a day.

Consistency (ACID):

Not critical, order is critical (load Repair and Vehicle first, then search for customer or add customer if necessary, finally add/change repairs related information).

Subtasks:

Mother Task is needed to coordinate subtasks. Decomposition is needed.



Abstract Code

- Users who login as Service Writer or Roland Around/Owner click on “*Repair Form*” button to **load the Repair Form**.
- Users enter a *VIN* to **search for Vehicle** on the Repair Form:
 - o **Search for Vehicle** using VIN
 - o If the Vehicle has not been sold (i.e., Is Sold is false)
 - **Display an error message** “This Vehicle is not sold out, should not have any repair”.
 - o If VIN does not match a Vehicle in the database:
 - **Display an error message** “Vehicle with this VIN not found”.
 - o Else (If the Vehicle is not sold and the VIN matches a Vehicle in the database):
 - The Repair Form **display details of the Vehicle**, including VIN, Vehicle Type, Model Year, Model Name, Manufacturer, and Color(s).
- On the **Repair Form**:
 - o **Search Repair** using VIN and Start Date.
 - o If no Repair is found for the Vehicle.
While no button is pushed, do nothing.
When a button is pushed, then do the following:
 - **Start new repair:**
 - Enter Odometer Reading
 - **Store the current date** as repair *start date*
 - Users enters driver’s license (‘\$driver’s license’), or tax ID (‘\$tax ID’) input fields.
 - **Lookup the customer**
 - o If the customer is found:
 - Stay on repair form, and the customer will be populated.
 - o If customer is not found:
 - If the customer is not found:
 - o Show **Add Customer** button to add the customer.
 - o If **Add Customer** is Clicked:

- Wait for user to input the following fields: *Phone Number, City, Street Address, Postal Code, State, Email (optional)*.
 - Show a dropdown to select either Individual or Business
 - If Individual is chosen:
 - Wait for user to input the following: *First Name, Last Name, Driver License Number,*.
 - If Business is Chosen:
 - Wait for user to input the following: *Tax Identification Number, Business Name. Primary Contact Name and Primary Contact Title*
 - o **Save** Button to persist the data.
- Enter other repair information:
 - o Add *Labor Charge*. **Validate** charge is a Float.
 - o Add *Parts* (Quantity, Vendor Name, Part Number, price). **Validate** Quantity is an Int and Price is a Float.
 - ⊕ **Display** Repair details.
- Click **SAVE** button. **Save** the new repair form
- o If there is an unfinished Repair for the Vehicle (indicated by Competition Date is null):

While no button is pushed, do nothing.

When a button is pushed, then do the following:

 - **Update the repair Button:**
 - update *Labor Charges*.
 - o If logged in as Service Writer

If new Labor Charge is equal or less than the previous value, throw error message “Labor Charge cannot be less than previous value”.

Labor Charge must be a Float.
 - o If logged in as the Roland Around/Owner

Add a new Labor Charge, it can be smaller the the preiovus value.

Labor Charge must be a Float.
 - Add Parts. Wait for user to input *Quantity, Vendor Name, Part Number, Price*:
 - o Perform validation :
 - Validate *Quantity* must be integer
 - Validate *Vendor Name* must be STRING
 - Validate *Part Number* must be String
 - Validate *Price* must be Float.
 - o If validation fails,

- **Display an error message**
 - o If all the validations passed:
 - **Populate the fields**
 - Display repair details.
 - **Complete the repair Button:**

The current date is stored on the repair as the completion date
 - Click **“Save”** button to save the updates
 - o If repair is found and finished (indicated by completion date not null):
 - **Display an error message** “Each Vehicle can only have maximum one repair a day”.
 - o If the user clicks **Cancel:**
 - Jump to Open the Repair Form page

Add Vehicle Sale

Task Decomposition

Lock Types:

Search Individual or Business: Read only lock

Add Individual or Business: Write only lock

Add Sale Transaction: Write only lock

Look up Vehicle: Read only lock

Number of Locks: 4

Enabling Conditions:

Logged in as Salesperson or Roland Around/Owner – **Add Sale** Button

Frequency:

Different frequency. About 10 times a day.

Consistency (ACID):

Not critical even if some others are looking up Sales Transaction or customer details.

Order:

Search Customer first, followed by Add Customer if necessary.

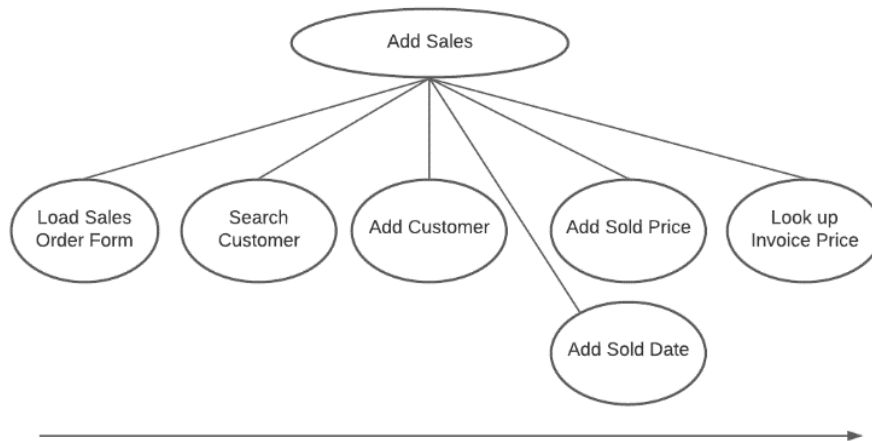
Add Sold Price

Invoice Price Lookup to validate Sold Price.

Subtask:

Master task is needed

Decomposition is needed



Abstract Code

- Users who login as Salesperson and Roland Around/Owner clicked **search vehicle** button to search vehicle by VIN.
- Then load the **View Vehicle Detail form**.
- Click the **Sell Vehicle** button then:
 - **Load the Sale Order Form**
 - Click **Search Customer** button to search customer
 - Users enters driver's license ('\$driver's license'), or tax ID ('\$tax ID') input fields.
 - Lookup the customer
 - If the customer is not found:
 - Show **Add Customer** button to add the customer.
 - If **Add Customer** is Clicked:
 - Wait for user to input the following fields: *Phone Number, City, Street Address, Postal Code, State, Email(optional)*.
 - Show a dropdown to select either Individual or Business
 - If Individual is chosen:
 - Wait for user to input the following: *First Name, Last Name, Driver License Number*.
 - If Business is Chosen:
 - Wait for user to input the following: *Tax Identification Number, Business Name. Primary Contact Name and Primary Contact Title*
 - **Save** Button to persist the data.
 - Else (customer found), load **customer information**:
 - If the customer is an individual, customer details included: customer's first and last names, driver license.
 - If the customer is a business, customer details included: The business's tax identification number, business name, and the name of a primary contact and their
 - Entering the *Sold Price*.

- Entering the *Sold Date*.
- If logged in as Salesperson AND a Sold Price \leq 95% of the Invoice Price
 - The system will reject the sale and show an error message.
- Else logged in as Roland Around/Owner OR a Sold Price $>$ 95% of the Invoice Price
 - The system will accept the sale and save the data to database
- When the **Cancel** button is clicked, the add sales window will close.

Add/Update Manufacturer

Task Decomposition

Lock Types:

View Manufacturer, Read only block

Add/Update Manufacturer, Write only block

Number of Locks: 2

Enabling Conditions:

Logged-in User (Inventory Clerk and Roland Around), enabled by add/edit requests.

Frequency:

Read (Most Frequent) > Add > Update. About 10 times a day.

Consistency (ACID):

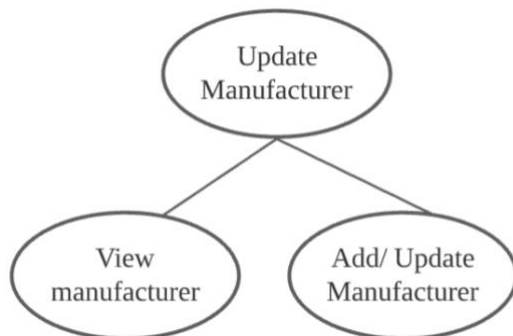
Not critical

Subtasks:

View Manufacturer, Add/Update manufacturer

Order:

All must be executed.



Abstract Code

- **View Manufacturer:**
Populate a list of Manufacturers
- While no button is pushed, do nothing.
- When a button is pushed, then do the following:
 - Wait for user to input *manufacturer name*.
 - If **Add Manufacturer:**
Persist the data.

View Manufacturer

Display manufacturer list with the newly added one.

- If ***Update Manufacturer:***

Persist the data.

Update Manufacturer

View manufacturer with the updated one.

- If ***Cancel***

View Manufacturer for the current list of Manufacturers