

Table of Contents

| | |
|--|----|
| Data Types | 1 |
| Business Logic Constraints | 3 |
| Task Decomposition with Abstract Code (AC) | 5 |
| Search Vehicle | 5 |
| Login | 6 |
| Add Vehicle | 6 |
| Sell Vehicle | 7 |
| View Vehicle | 8 |
| View Repair Form | 8 |
| Update Repair Form | 9 |
| Create a New Repair | 9 |
| Associate Customer | 10 |
| Open Report | 12 |

Data Types

User

| Attribute | Data type | Nullable |
|-----------|-----------|----------|
| Username | String | Not Null |
| Password | String | Not Null |
| LastName | String | Not Null |
| FirstName | String | Not Null |

Customer

| Attribute | Data type | Nullable |
|-----------|-----------|----------|
| Email | String | Null |
| Phone | String | Not Null |
| Address | String | Not Null |

Individual

| | | |
|---------------|--------|----------|
| DriverLicense | String | Not Null |
|---------------|--------|----------|

| | | |
|-----------|--------|----------|
| LastName | String | Not Null |
| FirstName | String | Not Null |

Business

| Attribute | Data type | Nullable |
|---------------------|-----------|----------|
| TaxId | String | Not Null |
| BusinessName | String | Not Null |
| PrimaryContactName | String | Not Null |
| PrimaryContactTitle | String | Not Null |

Repair

| Attribute | Data Type | Nullable |
|-------------------|-----------|----------|
| RelatedVIN | String | Not Null |
| StartDate | Date | Not Null |
| EndDate | Date | Not Null |
| Odometer | Integer | Not Null |
| LaborCharges | Float | Not Null |
| Description | String | Not Null |
| PartNumber | String | Null |
| UnitPrice | Float | Null |
| VendorName | String | Null |
| QuantityUsed | Integer | Null |
| ServiceWriterName | String | Not Null |

Vehicle

| Attribute | Data Type | Nullable |
|-----------|-----------|----------|
| VIN | String | Not Null |

| | | |
|---------------------|--------------|----------|
| Type | String | Not Null |
| Attributes | List<String> | Not Null |
| ManufacturerName | String | Not Null |
| ModelName | String | Not Null |
| ModelYear | Integer | Not Null |
| InvoicePrice | Float | Not Null |
| ListPrice | Float | Not Null |
| Color | List<String> | Not Null |
| DateAdded | Date | Not Null |
| Clerk Name | String | Not Null |
| SoldPrice | Float | Null |
| DateSold | Date | Null |
| CustomerId | String | Null |
| SalespeopleUsername | String | Null |
| Description | String | Null |

Business Logic Constraints

Anonymous Access:

- The only feature accessible without logging in is searching for vehicles (except by VIN).
- If the search returns some results, for each result, the visible attributes are: VIN, vehicle type, attributes for that vehicle type, Model Year, Model Name, Manufacturer, color(s), list price (NOT invoice price), and the description for the selected result.

Privileged Access:

- Privileged users will need to login using their username and password to gain access to appropriate functionality
- All privileged users will have an additional search option to search by VIN
- Inventory Clerk will have access to add new vehicles by given access to “Add Vehicle” functionality after login

- Inventory Clerk will have access to view invoice price on vehicle detail page
- Salespeople will have access to sell the vehicle
- Salespeople can look up customers on sales order form, or add them if customers not found
- Salespeople cannot confirm a sale with sold price less than or equal to 95% of the invoice price
- Service Writer will have access to open a repair form
- If provided VIN not found or the vehicle has not been sold, Service Writer will not be able to create a repair for this vehicle
- Service Writer will have access to start a new repair for current vehicle if no repairs are open for this vehicle
- Service Writer will only be able to update labor charges, add parts, complete repair if current vehicle has unfinished repair
- Updates to labor charges cannot be less than their previous value
- Service writer will be able to search or add customers who will be associated with the repair
- Manager will have the option to filter vehicle search results by sold, unsold or all vehicles
- Manager will see the inventory clerk that added the vehicle, the invoice price, and the date it was added to inventory when viewing vehicle detail page
- Manager will see the buyer's contact information, list price, sold price, sales date, and the salesperson's name if vehicle has been sold
- Manager will see the customer name (first and last name for individuals, or company name for companies), the service writer who entered the repair, and the repair's start date, end date, labor charges, parts cost, and total cost if any repairs have been made for this vehicle
- Roland has access to every functionality mentioned above.
- In addition, Roland can enter sold prices that are less than or equal to 95% of invoice price on a sales order form. Roland can update labor charges on a repair to a value less than their previous value
- Roland and managers have access to view reports

Vehicle

- Each vehicle type has attributes which are specific to that vehicle type.
- The list of manufacturer name can be updated
- Model years cannot exceed the current year plus one.
- Year entered must include century digits

Sales:

- The list price is calculated as 125% of the invoice price.
- If a buyer purchases several vehicles at the same time, they would be handled as separate sales transactions.

Repairs:

- The end date of a repair must be no earlier than its start date.
- A 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.

Task Decomposition with Abstract Code

Search Vehicle



Task Decomposition

Lock Type: 1 read-only lock for Vehicle.

Number of Locks: 1

Enabling Conditions: click on search

Frequency: High (100 search per day)

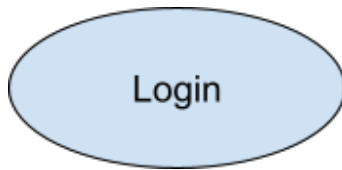
Consistency(ACID): Not critical

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

Abstract Code

- User clicks on **Search Vehicle** button, jump to search page, where initially display total number of vehicles available for purchase.
- User input *Vehicle type, Manufacturer, Model year, Color, List price, Keyword*. Privileged user can search by VIN
- User clicks **Search** button
 - Return unsold vehicles matching all search terms sorted by VIN in ascending order, where display
 - VIN
 - Vehicle type
 - Model year
 - Manufacturer
 - Model
 - Color(s)
 - If an entered keyword matched the description, indicate this with a mark
 - List price
 - If no vehicles meet the criteria, display "Sorry, it looks like we don't have that in stock!"
 - User click any columns header to sort result by the column
 - User click an individual result to open a detail page

Login



Task Decomposition

Lock Types: 1 read-only lock for User

Number of Locks: 1

Enabling Conditions: click on login button

Frequency: Medium

Consistency(ACID): Not critical

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

Abstract Code

- User click **Login** on the **Main Menu**
- User enters username and password
 - If validation for both username and password is correct, then page will update to include access to the appropriate functionality based on the user duty.
 - Else username or password is invalid, back to public-facing page with an error message

Add Vehicle



Task Decomp

Lock Type: 1 writing lock for Vehicle.

Number of Locks: 1

Enabling Conditions: Inventory Clerk logs in and click on add vehicle

Frequency: low

Consistency(ACID): Critical, added vehicle will be available for sale immediately

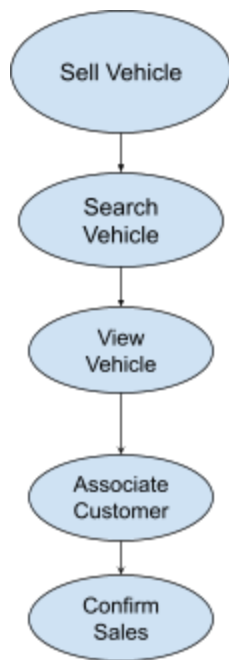
Subtasks: Mother task is not needed. No decomposition needed.

Abstract Code:

- After login, User click **Add Vehicle** to load new vehicle form
- User enters vehicle details, such as VIN, vehicle type, invoice price and date it is added.
- User clicks **submit** button.

- Display vehicle detail page
- Vehicle is available for sale

Sell Vehicle



Task Decomp

Lock Type: 2 read-only lock for Vehicle and Customer, 2 write lock for Vehicle and Customer

Number of Locks: 3 or 4 depends on the existence of a customer

Enabling Conditions: Salespeople log in and click on sell the vehicle

Frequency: low

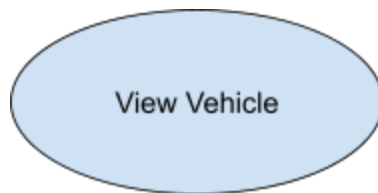
Consistency(ACID): Critical because salespeople must create a sales order form when confirming the sale.

Subtasks: All tasks must be done, but can't be done in parallel. Availability check needs to be done first then click the one to be sold, followed by the confirmation of the sale.

Abstract Code

- Run **Search vehicle** task to check available vehicles
- click a vehicle to view details.
- User clicks the **Sell vehicle** button to sell the vehicle
 - Load the sales order form
 - Associate customer
 - Look up a customer
 - Add a customer if not exist
 - Confirm the sale by entering the sold price and sold date

View Vehicle



Task Decomp

Lock Type: 1 read-only lock for Vehicle.

Number of Locks: 1

Enabling Conditions: Vehicle appears in the list after **Search Vehicle**, click on the vehicle

Frequency: medium

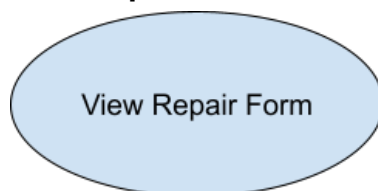
Consistency(ACID): Not critical

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

Abstract Code

- User clicks a vehicle link to open a detailed page
- Display VIN, vehicle type, attributes for that vehicle type, Model Year, Model Name, Manufacturer, color(s), list price and the description.

View Repair Form



Task Decomposition

Lock Type: Read-only on Vehicle, Read-only on Repair

Number of Locks: 2

Enabling Conditions: Service Writer logs in, Service Writer clicks on button to open repair form after logged in

Frequency: Medium

Consistency(ACID): Not critical, order is not critical

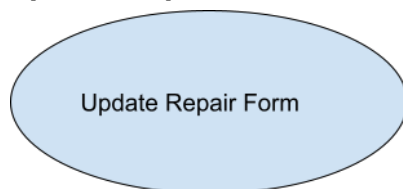
Subtasks: Mother task is not needed. No decomposition needed

Abstract Code

- Service Writer clicks on **Open Repair Form** link / button
- Service Writer will be prompted to enter a VIN
- If VIN does not match a vehicle in vehicle table, then:
 - An appropriate error message will be displayed

- If the vehicle with given VIN has not been sold, then:
 - An appropriate error message will be displayed
- Otherwise, the repair form will show following fields: VIN, vehicle type, Model year, Model Name, Manufacturer and color(s)

Update Repair Form



Task Decomposition

Lock Type: Writing lock on Repair

Number of Locks: 1

Enabling Conditions: Service Writer logs in, service writer opens Repair Form with valid VIN, the vehicle has unfinished active repair

Frequency: Medium

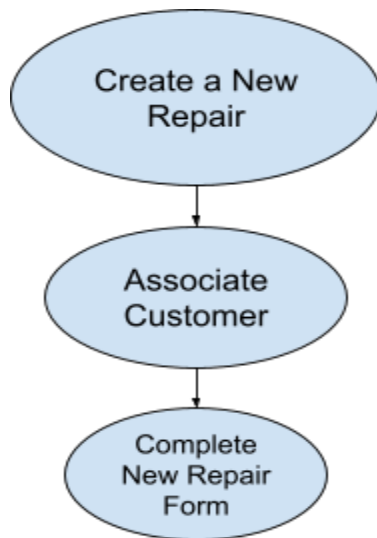
Consistency(ACID): Critical, order important for **Create a New Repair**, given one Vehicle can only have one active repair

Subtasks: Mother task is not needed. No decomposition needed

Abstract Code

- Service Writer can
 - update labor charges,
 - add parts, or
 - complete the repair
- If service write choose to complete the repair, then:
 - Current date will be stored on the repair as completion date

Create a New Repair



Task Decomposition

Lock Type: Write lock on Repair, Write / Read lock on Customer

Number of Locks: 2 (or 3 depends on if **Add Customer** is necessary)

Enabling Conditions: Service writer logs in, service writer opens repair form with valid VIN, the vehicle has no active repairs

Frequency: Medium

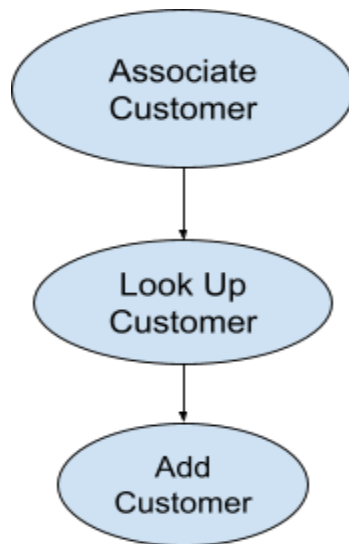
Consistency(ACID): Critical, order important, each vehicle can have at most 1 repair starting on the same date

Subtasks: All subtasks need to be done.

Abstract Code

- Service writer enters odometer reading
- Service writer will need to search for customers associated with this repair, or add new customer
- Service writer enters
 - labor charges,
 - add parts and
 - enter all relevant details(quantity, vendor, part number, price)

Associate Customer



Task Decomposition

Lock Type: Read / Write Lock on Customer

Number of Locks: 1 (or 2 depends on if **Add Customer** is necessary)

Enabling Conditions: Salespeople create a new Sales Order Form (**Sell Vehicle**), or when Service writer **Create a New Repair**

Frequency: Medium

Consistency(ACID): Critical, a customer can be involved in a sales order and an active repair at the same time

Subtasks: Add Customer is optional, mother task is required to trigger subtasks

Abstract Code

- Priviledged User can look up a customer with either
 - Driver's license, or
 - Tax ID
- If no result is found, then:
 - Option to add new customer is provided
- Priviledged user choose a Customer type to add new Customer
- If an individual is to be added, then:
 - Priviledged user needs to record
 - Phone number
 - Address
 - Email is optional
 - First and Last Name
 - Driver's license number
- If a business is to be added, then:
 - Priviledged user needs to record
 - Phone Number
 - Address
 - Email is optional

- Business Tax Identification Id
- Business Name
- Name of Primary Contact
- Contact Title

Open Report



Task Decomposition

Lock Type: 4 read-only locks for Vehicle/Repair/Customer/User lookup

Number of Locks: 1 to 4 depending on the report type

Enabling Conditions: Roland or Managers login and click “Open Report” and select a report type from the dropdown menu

Frequency: Medium (moderate monthly access by managers and Roland)

Consistency(ACID): Not critical

Subtasks: Only one subtask should be done

Abstract Code

- Run **Open Report** task to expand the menu and click the report type to generate.
- Case “Sales by Color”
 - Read all *Vehicle* entities
 - Grouped by *Color* (“multiple for more than 1 color), count the number of vehicles sold, where the *DateSold* is for each “within 30 days”, “within a year”, “any”.
 - Display the form where colors are the rows.
- Case “Sales by Type”
 - Read all *Vehicle* entities
 - Grouped by *Type*, count the number of vehicles sold, where the *DateSold* is for each “within 30 days”, “within a year”, “any”.
 - Display the form where types are the rows.
- Case “Sales by Manufacturer”
 - Read all *Vehicle* entities

- Grouped by *Manufacturer*, count the number of vehicles sold, where the *DateSold* is for each “within 30 days”, “within a year”, “any”.
 - Drop the manufacturers with zero sales and display the form where manufacturers are the rows.
- Case “Gross Customer Income”
 - Read all *Vehicle* entities
 - Grouped by *CustomerId*, sum the *SoldPrice*. Keep all records for each customer as a list.
 - Read all *Repair* entities
 - Grouped by *CustomerId*, sum the total repair cost from each record. Keep all records for each customer as a list.
 - Sum up the 2 results for each customer and keep the vehicle purchase records with repair records together, ordered by total expenses descending and then by last sale/repair start date descending.
 - Read customer’s name (first/last for individuals or business name for business) from *Customer* entities.
 - Select top 15 customers with following attributes: name, the date of the first sale or repair start date, the date of their most recent sale or repair start date, their number of sales, their number of repairs, and the gross income. Display the report. (first part).
 - Click on the link within any customer’s name
 - Expand the 2 lists of records stored priorly into:
 - Vehicle Sales Section: sale date, sold price, VIN, year, manufacturer, model, and salesperson name (Read from *User* entities given *SalespeopleUsername*), for each sale, sorted by sale date descending and VIN ascending.
 - Repairs Section: start date, end date (if the repair is not finished, this should not display any value), the VIN of the repaired vehicle, the odometer reading, parts cost, labor cost, total cost, and the service writer who opened the repair, for each repair, 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 drilldown report with 2 sections (second part).
- Case “Repairs by Manufacturer/Type/Model”
 - Read all *Vehicle* entities and all *Repair* entities and join results where *Vehicle.VIN = Repair.RelatedVIN*
 - Grouped by *Manufacturer*, compute the count of repairs, the sum of all parts costs, the sum of all labor costs, and the sum of total repair costs, including any repairs in progress. Keep the list of repairs for each manufacturer.
 - Display the report using the stats from the last step, including manufacturers whose vehicles do not have any repairs, sorted by manufacturer name ascending (first part).
 - Click on the link within any manufacturer’s name

- Within the list of repairs per manufacturer in the last part, group by *Type*, compute the count of repairs, the sum of all parts costs, the sum of all labor costs, and the sum of total repair costs, including any repairs in progress. Keep the list of repairs for each vehicle type, if the list is empty, exclude the vehicle type.
 - Within the list of repairs per vehicle type, group by *Model*, compute the count of repairs, the sum of all parts costs, the sum of all labor costs, and the sum of total repair costs, including any repairs in progress. If the list is empty for the model, exclude the model. Mark the list generated here as “details” for the vehicle type.
- Display the drilldown report of vehicle type, repair count, parts costs, labor costs, , total costs and details, sorted by repair count descending with vehicle type sorted first, and then detail rows sorted (second part).
- Case “Below Cost Sales”
 - Read all *Vehicle* entities
 - Filter out the entities whose *SoldPrice* < *InvoicePrice*
 - For each, read the salesperson’s name and customer’s name (first/last for individuals or business name for businesses) from *User* and *Customer* entities with key *SalespeopleUsername* and *CustomerId* respectively.
 - For a sale where *SoldPrice* to *InvoicePrice* ratio is less than or equal to 95%, the background of that row should be highlighted red. Display the report by sales date descending and ratio descending.
- Case “Average Time in Inventory”
 - Read all *Vehicle* entities
 - Grouped by *Type*, compute the the average amount of time sold vehicles remain in inventory (*DateSold* - *DateAdded*)
 - Display the report (if a vehicle type has no sales history, the report should display “N/A” for that vehicle type).
- Case “Part Statistics”
 - Read all *Repair* entities
 - Extract *Parts* for each *Repair* where not null
 - Grouped by *VendorName*, sum the *QuantityUsed* and *UnitPrice times QuantityUsed*.
 - Using those stats, display each vendor: the vendor’s name, the number of parts supplied by that vendor, and the total dollar amount spent on parts.
- Case “Monthly Sales”
 - Read all *Vehicle* entities
 - Filter out the sold entities and sorted by *DateSold* descending
 - Using a window side of a year and a month respectively, slice the whole list into a list of windows (each contains a list of sold vehicles).
 - For either yearly and monthly window, drop if empty, count the number, sum the *SoldPrice* for total sales income, the *InvoicePrice* for total invoice price. The net income is total sales income less total invoice price. Keep a “total sales income/total invoice price” ratio.

If the ratio 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.

- Display the report with the total number of vehicles sold, the total sales income, the total net income, and ratio as a percentage (such as 125%), for each yearly and monthly. (first part)
- Using the yearly and monthly window sliced results in the earlier step, given any year or month window, Grouped by *SalespeopleUsername*, count the number of records in each person and sum the *SoldPrice*. Read the salesperson's first and last name from *User* with key *SalespeopleUsername*.
- Display the drilldown report with the salesperson's first and last name, the number of vehicles they sold in that year and month and their total sales for that year and month, sorted by total vehicles descending followed by total sales descending. (second part)