CS 6400

Phase 1 Report

GTSI Team 04

GTSI Team 04
2021-10-4

Table of Contents

Data Types:

User:

Attribute	Data type	Nullable
username	String	Not Null
first_name	String	Not Null
last_name	String	Not Null
user_type	String	Not Null
password	String	Not Null

Vehicle:

Attribute	Data type	Nullable
manufacturer_name	String	Not Null
model_name	String	Not Null
model_year	Integer	Not Null
color	List<String>	Not Null
description	String	Null
invoice_price	Float	Not Null
warehousing_date	Date	Not Null
vehicle_type	Enum(String)	Not Null

Car:

Attribute	Data type	Nullable
number_of_doors	Integer	Not Null

Convertible:

Attribute	Data type	Nullable
roof_type	String	Not Null
back_seat_count	Integer	Not Null

Trunk:

Attribute	Data type	Nullable
cargo_capaticy	Float	Not Null
cargo_cover_type	String	Null
number_of_rear_axles	Integer	Not Null

Van/Minivan:

Attribute	Data type	Nullable
has_drivers_side_back_door	Boolean	Not Null

SUV:

Attribute	Data type	Nullable
drivetrain_type	String	Not Null
number_of_cupholders	Integer	Not Null

Customers:

Attribute	Data type	Nullable
address	String	Not Null
phone_number	String	Not Null
email	String	Null

Individual:

Attribute	Data type	Nullable
drivers_license_number	String	Not Null
first_name	String	Not Null
last_name	String	Not Null

Business:

Attribute	Data type	Nullable
tax_identification_number	String	Not Null
business_name	String	Not Null
primary_contact_name	String	Not Null
primary_contact_title	String	Not Null

Sale:

Attribute	Data type	Nullable
sold_price	Float	Not Null
purchase_date	Date	Not Null

Repair:

Attribute	Data type	Nullable
start_date	Date	Not Null
complete_date	Date	Null

odometer_reading	Float	Not Null
labor_charge	Float	Not Null
description	String	Not Null

Part:

Attribute	Data type	Nullable
part_number	String	Not Null
vendor_name	String	Not Null
Quantity	Integer	Not Null
Price	Float	Not Null

Business Logic Constraints:

Vehicles

- Model years cannot exceed the current year plus one
- The year entered must include century digits
- The invoice price should include dollars and cents

Access

- Anonymous users can only search for vehicles
- Accessing other functionalities requires user to login

Repair

- The date the repair was completed must not before the start date
- A vehicle will never have more than one repair starting on the same date
- A repair must be completed before a new one can be started
- A new repair can only be created if there is no incomplete repair for the vehicle, otherwise only updating labor charges, adding parts, or completing the repair are allowed
- Labor charges should include dollars and cents
- Any updates to labor charges cannot be less than their previous value
- Roland is allowed to update the labor charges on a repair to a value less than their previous value

Sell

- Sold price should include dollars and cents
- If a sold price is less than or equal to 95% of the invoice price, the system will reject the sale
- Roland is allowed to enter sold prices that are less than or equal to 95% of the invoice price
- If a buyer purchases several vehicles at the same time, they will still be handled as separate sales transactions

Login

Task Decomp

Lock Types: Read-only on User table.

Number of Locks: Single

Enabling Conditions: None

Frequency: Around 200 logins per day.

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

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

Abstract Code

- User enters username('$Username') and password('$Password') in input field on Login Form
- If data validation is successful for both username and password, then:
    - When Login button is clicked:
        - If a User record is found using '$Username', and User.password = '$Password', then:
            - Go to **Navigation Panel**
        - Else:
            - Go back to **Login Form**, with Error Message
- If user clicks **Search Vehicle** button, go to Search **Vehicle task**
-

Navigation Panel

Task Decomp

Lock Types: Read-only on User table.

Number of Locks: Single

Enabling Conditions: Triggered by successful login.

Frequency: The same as login.

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

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

Abstract Code

- Based on User.user_type, different buttons are shown on the panel
    - **Search Vehicle** button and **Log Out** button are shown for all user
    - User.user_type = 'Inventory Clerk' - **Add Vehicle** button is shown
    - User.user_type = 'Salespeople' - no extra button is shown because need to search before selling
    - User.user_type = 'Service Writer' - **Search Repair** button is shown
    - User.user_type = 'Manager' - **View Report** button is shown
    - User.user_type = 'Owner' - all buttons mentioned above are shown
- Upon:
    - Click **Search Vehicle** button - Jump to the **Search Vehicle** task
    - Click **Add Vehicle** button - Jump to the **Add Vehicle** task
    - Click **Search Repair** button - Jump to the **Search Repair** task

- Click **View Report** button - Jump to the **View Report** task
- Click **Log Out** button - Invalidate login session and go back to **Login** form

Search Vehicle(Scenario 1: anonymous access)

Task Decomp

Lock Types: Read-only on Vehicle table, Manufacturer table(no Color table since it's immutable)

Number of Locks: 2

Enabling Conditions: None

Frequency: About 1 QPS

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

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

Abstract Code

- User optionally selects vehicle type('$Vehicle Type'), color('$Color'), manufacturer name('$Manufacturer Name') and enters model year('Model Year'), list price('List Price') with selecting '<' or '>' and keyword('$Keyword')
  When **Search** button is clicked:
  - If no unsold vehicles meets the search criteria, then:
    - return an error message, allowing the user to search again
  - Else:
    - Display the VIN, vehicle type, model year, manufacturer, model, color, matching keyword and list price sorted by VIN in ascending order
    - User click **Details** button:
      - Display the VIN, vehicle type with attributes, model year, model name, manufacturer, color(s), list price and description on **Details** page

Search Vehicle(Scenario 2: logged in user)

Task Decomp

Lock Types: Read-only on Vehicle table, Color table, Potentially on Sale table, Repair table, Part table

Number of Locks: 5

Enabling Conditions: With successful log in

Frequency: About 200 times per day

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

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

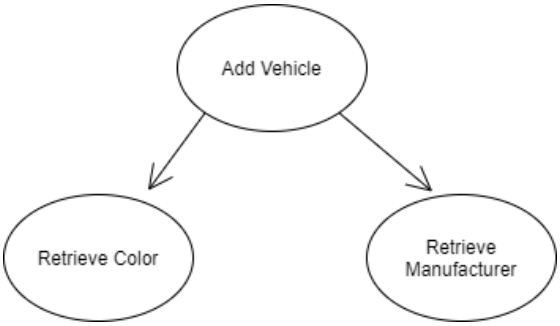
Abstract Code

- If User.user_type = 'Inventory Clerk' or 'Service Writer':
  - Similar to Scenario 1, but with an extra searching criteria VIN, and **Details** page will additionally display invoice price
- If User.user_type = 'Salesperson'':
  - Similar to Scenario 1, but with an extra searching criteria VIN, and **Details** page will additionally display invoice price
  - If **Sell** button on the detail page is clicked, go to **Sell Vehicle** task
- If User.user_type = 'Manager':

        - Similar to Scenario 1, but with an extra searching criteria VIN
        - If the vehicle has been sold, the buyer's information(except driver's license number of tax ID number) , List price, sold price, sale price and salesperson's first and last name are shown on **Details** page
        - If the vehicle has repair records, the customer name, service writer's first and last name, start date, complete date, labor charges, parts cost and total cost of each record are shown in the repair section
- If User.user_type = 'Owner':
        - Similar to Scenario 1, but with an extra searching criteria VIN
        - Can see all the information and buttons mentioned on **Details** page

Add Vehicle

Task Decomp



Lock Types: Write on Vehicle table, read and write on Manufacturer table

Number of Locks: Single read locks and 2 write locks

Enabling Conditions: Logged in and User.user_type='Inventory Clerk' or 'Owner'

Frequency: Around 10 times per day

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

Subtasks: Retrieve Color and **Retrieve Manufacturer**. Mother task is required to store information in Vehicle table.

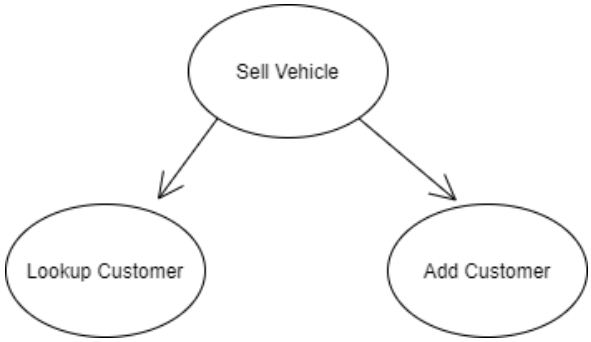
Abstract Code

- Retrieve Color and Manufacturer to display in drop-down lists
- User enters description('$description'), warehousing date('$current date'), model year('$model year'), invoice price('$invoice price') in input field on Vehicle Form.
    - User selects or enters manufacturer name('$manufacturer name'), multiply selects color('$color') and selects vehicle type('$vehicle type') in the input field on Vehicle Form.
        - If Vehicle.vehicle_type = 'Car':
            - User enters number of doors('$Number of Doors')
        - If Vehicle.vehicle_type = 'Convertible':
            - User enters roof type('$Roof Type'), back seat count('$Back Seat Count')
        - If Vehicle.vehicle_type = 'Truck':
            - User enters cargo capacity('$Cargo Capacity'), number of rear axises('$Number of Rear Axles')
        - If Vehicle.vehicle_type = 'Van/Minivan':
            - User enters has drivers side back door('$Has Drivers Side Back Door')
        - If Vehicle.vehicle_type = 'SUV':
            - User enters number of cupholders('$Number of Cupholders'), drivetrain type('$Drivetrain Type')

- User clicks Add button:
  - If data validation succeeds:
    - Store all information in Vehicle table
    - Go to **Details** page to display vehicle information
  - Else:
    - Display an error message, allowing user to edit again

Sell Vehicle

Task Decomp



Lock Types: Read and write on Vehicle table, Customers table

Number of Locks: 2 read locks and 2 write locks

Enabling Conditions: Login in; Search Vehicle and User.user_type='Salespeople' or 'Owner'

Frequency: Around 200 times per day.

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

Subtasks: Lookup Customer and **Add Customer**. Mother task is required to store information in Sale Table

Abstract Code

- User enters VIN('$VIN') to search the target vehicle:
  - If the vehicle has not been sold, or the VIN does not match a vehicle:
    - Display an error message, allowing to search again
  - Else:
    - User enters purchase date('$current date'), sold price('$sold price') in input field on Sales Order Form
    - User clicks **Lookup Customer** button, go to **Lookup Customer** task
    - User clicks **Add Customer** button, go to **Add Customer** task
    - If no customer is selected or created, display an error message
    - Else:
      - If User.user_type = 'Salespeople' and '$Sold Price' < Vehicle.invoice_price * 0.95:
        - Display error message ("Sold price is too low.")
      - Else:
        - User selects customer type('$Customer Type') in input field
        - If Customer.customer_type = 'Individual':
          - User enters driver license number('$Driver License Number') in input field on Sales Order Form
        - If Customer.customer_type = 'Business':
          - User enters tax identification number('$Tax Identification Number') in input field on Sales Order Form

- User clicks Sell button:
    - If data validation succeeds:
        - All information entered stores in Sale table
    - Else:
        - Display an error page, allowing user to edit again

Lookup Customer

Task Decomp

Lock Types: Read-only on Customer table.

Number of Locks: Single

Enabling Conditions: Add repair or **Sell Vehicle**

Frequency: 200 queries per day.

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

Subtasks: Mother task is not required; There is no subtask.

Abstract Code

- User enters driver license number('$Driver License Number') or tax identification number('$Tax Identification Number') and clicks **Search** button
    - If matching record is found, then:
        - The corresponding customer is selected
    - Else:
        - Display an error message, allowing user to look up again

Add Customer

Task Decomp

Lock Types: Read and Write on Customer table.

Number of Locks: single read or write lock

Enabling Conditions:

- **Add repair** or **Sell vehicle**

Frequency: 200 queries per day

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

Subtasks: Mother task is not required; There is no subtask.

Abstract Code

- User selects customer type in drop-down list:
    - If 'Individual' is selected:
        - User enters First name, Last name, Driver license number, along with address, phone number and email
    - If 'Business' is selected:

        - User enters primary contact name, title, tax identification number, business name, along with address phone number and email;

- User clicks **Add** button:
    - If data validation succeeds, then:
        - All information is stored in Customer table
    - Else:
        - An error message is displayed, allowing user to edit again

View Repair

Task Decomp

Lock Types: Read-only on Vehicle table, Repair table and Part table.

Number of Locks: 3

Enabling Conditions: Successful login and User.user_type = 'Service writer' or 'Owner'

Frequency: Around 100 queries per day

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

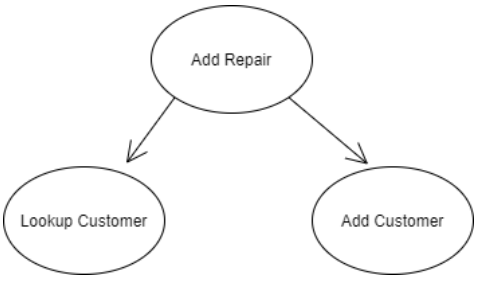
Subtasks: Mother task is not Needed. No decomposition needed.

Abstract Code

- User. user_type = 'Service Writer' or 'Roland Around' - Have a link or button to open the repair form
- User enters VIN('$VIN') in the input field to search the target vehicle
    - the vehicle has not been sold, or the VIN does not match a vehicle - display an error message
    - has been sold and match - the rest of the repair form will be displayed
- If the user want to process repair, based on whether a repair for the vehicle is completed
    - no ongoing repair - go to task **Add Repair**
    - exists ongoing repair - go to task **Edit Repair**

Add Repair

Task Decomp



Lock Types: Read and write on Customer table, Repair table, Part table.

Number of Locks: 3 read locks and 3 write locks

Enabling Conditions: Successful login and User.user_type = 'Service writer' or 'Owner', complete **View Repair**.

Frequency: Around 20 times per day

Consistency(ACID): First Add customer, then Add Part and finally store repair form and parts information in the database

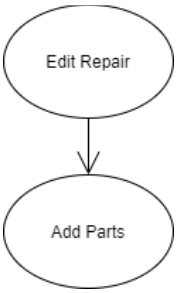
Subtasks: Mother task required for storing all the information for repair. Subtasks are **Lookup Customer** or **Add Custome**r

Abstract Code

- User enters odometer reading('$Odometer'), current date automatically displays as the start date('$StartDate') on Repair Form
- User clicks **Lookup Customer** button, go to **Lookup Customer** task
- User clicks **Add Customer** button, go to **Add Customer** task
- If no customer is selected or created, then:
    - Display an error message
- Else:
    - User enters labor charges('$LaborCharges') and description('$Description') in the input field on Repair Form
- User clicks **Add** button:
    - If data validation succeeds, then:
        - Store the information on Repair form and Parts in the database
    - Else:
        - Display an error message, allowing user to edit again

Edit Repair

Task Decomp



Lock Types: Read and write on Repair table, Part table, read-only on Customer table

Number of Locks: 3 read locks and 2 write locks

Enabling Conditions: View Repair

Frequency: Around 100 times per day

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

Subtasks: Mother task required for repair form. Subtask is **Add Parts**.

Abstract Code

- User updates labor charges('$Labor Charges') in the input field on Repair Form
    - User. user_type = 'Service writer' - any updates to labor charges cannot be less than their previous value
    - User. user_type = 'Owner' - the repair form will allow him to update a value less than their previous value
- User can add parts - go to task **Add Parts**
- User clicks **Complete** button - the current date is stored as the complete date ('$CompleteDate')
- User clicks **Save** button:

- If data validation succeeds, then:
  - All information stored in Repair table and Part table
- Else:
  - Display an error message, allowing user to edit again

Add Parts

Task Decomp

Lock Types: Read and write on Part table

Number of Locks: 2

Enabling Conditions: Edit Repair

Frequency: Around 50 times per day.

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

Subtasks: No decomposition needed.

Abstract Code

- User enters quantity('$Quantity'), vendor name('$Vendor Name'), part number('$Part Number'), price('$Price') in the input field on Repair Form
- User clicks Add button:
- If data validation succeeds, then:
  - All information stored in Repair table
- Else:
  - Display an error message, allowing user to edit again

View Report

Task Decomp

Lock Types: Read-only on Vehicle table, Sale table for sales by color report(No lock on Color table since it's immutable); read-only on Vehicle table, Sale table for sales by type report; read-only on Vehicle table, Sale table for sales by manufacturer report;

read-only on Customers table, Sale table, Repair table, Part table, Individual table, Business table for gross customer income report; read-only on Vehicle table, Repair table, Part table, Manufacturer table for repairs by manufacturer/type/model report; read-only on Vehicle table, Sale table for below cost sales report; read-only on Vehicle table, sale table for average time in inventory report; read-only on Part table for parts statistics report; read-only on Vehicle table, Sale table for monthly sales report.

Number of Locks: See lock types.

Enabling Conditions: Must logged in, and User.user_type = 'Manager' or 'Owner'

Frequency: About 10 times per day

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

Subtasks: Sales by Color, Sales by Type, Sales by Manufacturer, Gross Customer Income, Repairs by Manufacturer/Type/Model, Below Cost Sales, Average Time in Inventory, Parts Statistics, Monthly Sales. Subtasks are not required to be done. Order is neither required. Mother task required to coordinate subtask.

Abstract Code

- Different buttons corresponding to different reports are shown
- User clicks a button - jump to the corresponding subtask
- Subtask will show the corresponding report, click **Back** button to return to mother task
- Clicks **Back** button to **Navigation Panel**