Phase 1 Report | CS6400 – Fall 2019 **| Team 60**

Table of Contents:

Table of Contents

Burdell's Ramblin' Wrecks Data Types

Burdell's Ramblin' Wrecks Constraints

Task Decomposition with abstract code

Data Types:

Vehicle

Attribute	**Data Type**	**Nullable**
Vin	String	Not Null
ManufacturerName	String	Not Null
ModelName	String	Not Null
Model Year	Date	Not Null
VehicleType	String	Not Null
Mileage	Int	Not Null
Color	List<String>	Not Null
Description	String	Null

Customer

Attribute	**Data Type**	**Nullable**
Customer ID	String	Not Null
PhoneNumber	Number	Not Null
EmailAddress	String	Not Null
City	String	Not Null
Street	String	Not Null
State	String	Not Null
postal	Int	Not Null

Privileged User

Attribute	**Data Type**	**Nullable**
Username	String	Not Null
Password	String	Not Null
PrivilegeType	String	Not Null

Vendor

Attribute	**Data Type**	**Nullable**
VendorName	String	Not Null
PhoneNumber	Number	Not Null
City	String	Not Null
Street	String	Not Null
State	String	Not Null
postalCode	Int	Not Null

Part

Attribute	**Data Type**	**Nullable**
PartNumber	String	Not Null
Cost	Double	Not Null
Description	String	Null

Loan

Attribute	**Data Type**	**Nullable**
LoanId	int	Not Null
DownPayment	Double	Not Null
LoanTerm	int	Not Null
MonthlyPayment	Double	Not Null
InterestRate	Double	Not Null
StartMonth	Int	Not Null

IndividualCustomer

Attribute	**Data Type**	**Nullable**
DriverLicenceNumber	String	Not Null
FirstName	String	Not Null
LastName	String	Not Null

BusinessCustomer

Attribute	**Data Type**	**Nullable**
TaxIDNumber	String	Not Null
Businessname	String	Not Null
PrimaryContactName	String	Not Null
PrimaryContactTitle	String	Not Null

PartOrder

Attribute	**Data Type**	**Nullable**
OrderNumber	String	Not Null
SumOfPartsCost	Double	Not Null

Business Logic Constraints

Vehicle

- The model year cannot be one greater than current year and should follow **yyyy** format.
- The SalesPrice of the vehicle cannot be updated or modified.

Customer

- The customer can only be able to see the public page.
- The customer can only view the public information of the vehicle and all the other information should be hidden.
- Only the vehicles which are available for sale and no parts order pending will be visible on the public page.
- A customer cannot log into the system
- A customer cannot search based on the VIN.
- A customer has no access to system reports.

Clerks

- A clerk having privileges can log into the system.
- A clerk cannot view a sold vehicle.
- A newly added car should show 0$ for parts order.
- Once a part status is marked as completed its status can no longer be updated.
- A part status cannot be reverted to the older status.

Sales people

- Can view only vehicle available for sale and with no parts order in pending status.
- The selling price of the car cannot be changed by any sales person.
- SalesDate cannot be older than the PurchaseDate and also cannot be a future date.

Manager

- Manager can only view the information and cannot perform any action.
- The driver license number or the tax id number should not be visible to the manager.

Mr. Burdell

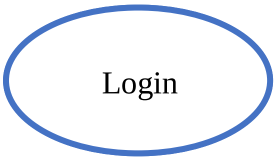
- Cannot add a part to or sell a car which has already been sold.

Reports

- The seller should be marked with red if the cars sold by him has more than 5 parts on the report
- The seller should be marked red if the parts cost of the cars sold by him is more than 500$
- Mr Burdell and only managers will have access to all the reports.

Login

Task Decomposition



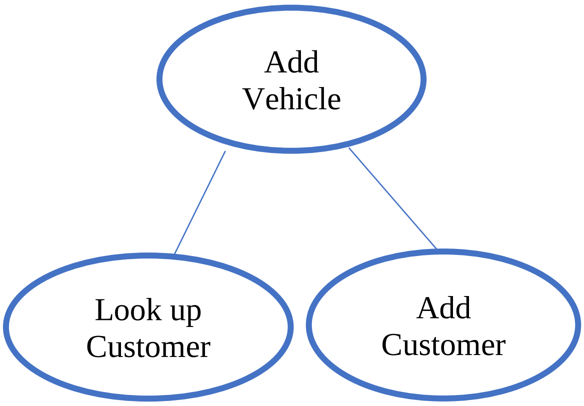
Lock Types: Read only on Privileged User table
Number of Locks: Single
Enabling Conditions: None
Frequency: Around 100 logins per day
Consistency(ACID): not critical
Subtasks: No decomposition is required

Abstract Code

- User enters username ('username'), password ('password') input fields.
- If data validation is successful for both *username* and *password* input fields, then
    - When **Enter** button is clicked:
        - If User record is found but user.password != "password":
            - Go back to **Login** form, with error message.
        - Else:
            - Store login information as session variable 'username'
            - Go to **Main Page**
- Else email and password input fields are invalid, display **Login** form, with error message.

Add Vehicle

Task Decomposition



Lock Types: One read only lookup and on write on Customer table and one write on the vehicle table
Number of Locks: One read, and two write locks
Enabling Conditions: privileged user is logged in.
Frequency: Around 20 vehicles are added per day
Consistency(ACID): The add vehicle transaction is marked as done only when add customer, the linking of customer to vehicle and add vehicle is completed.
Subtasks: Decomposition is required as Add vehicle consists of two logical independent tasks which are **Lookup Customer** and **Add Customer.**

Abstract Code

- Privileged user will click on the add vehicle button.
    - If the *session* for the privileged user is live:
        - Fill in the vehicle details in the add vehicle form.
        - Look up for the customer to be linked to the vehicle.
        - If the customer is found in the lookup :
            - Select the customer.

    - Else:
        - Fill all the details for the customer.
    - Look up the customer list to select the newly added customer.
    - Add the date of purchase
    - The date of purchase will be validated if the date is not from the future.
    - click on the submit button
- Else:
    - Take the privileged user to the **Login Page**.

List Vehicle

Task Decomposition

List Vehicle

Lock Types: Read only on Vehicles Table
Number of Locks: Single
Enabling Conditions: None
Frequency: Around 500 view per day
Consistency(ACID): not critical
Subtasks: No decomposition is required

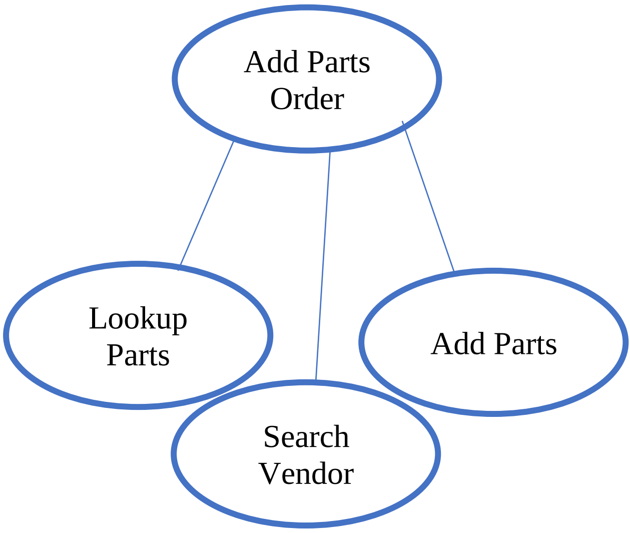
Abstract Code

- Customer will view the **Public Page.**
- Customer will select a filter out of the valid filters from a drop down.
- If there any vehicles which match the filter:
    - List all the vehicles.
- Else:
    - Print an appropriate *error* message.

Add Parts Order

Task Decomposition

Lock Types: One write lock on Parts Order,
One read-write lock on parts, one read lock on
parts
Number of Locks: One write lock, One read-write lock
One write lock
Enabling Conditions: Privileged user is logged in,
Request to add parts order.
Frequency: Search vendor and lookup parts have
same frequency
Consistency (ACID): Consistency is not critical,
order is not critical
Subtasks: Mother task in needed to coordinate subtasks



Abstract Code

- **View Public Search page** for privileged user. List parts and status (Use **List Parts**)
- While no button is clicked do nothing
- When ***Add Part*** button is clicked then:
    - **Lookup Parts.** Look for existing parts
    - If part exists then:
        - Select part.
    - Else:
        - **Add Parts.** Enter Status, Cost, Description and Part Number.
    - If ***Add Another Part*** button is clicked then
        - **Add Parts.** Enter Status, Cost, Description and Part Number.
    - **Search Vendors.** Look for existing vendors.
    - Select the vendor from the list and add it to the **Add Parts Order** form.
- If ***Save Order*** button is clicked then
    - View **Public Search page** for privileged user.

Search Vehicle

Task Decomposition

Lock Types: One read lock on Vehicle
Number of Locks: One read lock
Enabling Conditions: None
Frequency: Very frequent
Consistency (ACID): Consistency is not critical
Subtasks: Mother task is not needed, no decomposition needed

Abstract Code

- View total number of cars available for purchase;
- If user enters a keyword:
    - find entries that match the keyword and display details;
    - if no entries match:
        - display error
- else
    - Populate search criteria drop downs.
    - While no options are selected do nothing
    - If option is selected then **list vehicles** that match selected criteria:
        - Display VIN, vehicle type, model year, manufacturer, model, colours, mileage and sales price of all matched vehicles in search result
        - If a car is selected from the list :
            - Display VIN, vehicle type, model year, manufacturer, model name, colors, mileage, sales price and description of car (Use **view vehicle details**)

Lookup Customer

Task Decomposition

Lock Types: One read lock on Customer
Number of Locks: One read lock
Enabling Conditions: Privileged user's login
Frequency: Frequent
Consistency (ACID): Consistency is not critical
Subtasks: Mother task is not needed, no decomposition required

Abstract Code

- Privileged user clicked on ***Sell Car*** button on **Public Search Page.**
- This loads **Sales Order** form.
- Upon clicking ***Lookup Customer*** button on **Sales Order** form
    - Search for the customer
    - If the customer is found:
        - Select the customer
    - Else:
        - Add a new customer (Use **Add Customer**)
- Click on ***Save*** button to save the changes.

Update Sales Date

Task Decomposition

Lock Types: One write lock on Vehicle table
Number of Locks: One Write lock
Enabling Conditions: Privileged user's login
Frequency: Frequent
Consistency (ACID): Consistency is not critical
Subtasks: Mother task is not needed, no decomposition required

Abstract Code

- On **Sales Order** form, update customer details (Use **Lookup Customer**).
- Add Sales Date on **Sales Order** form.
- Click on ***Save*** button to save the changes.

Update Loan Details

Task Decomposition

Lock Types: One write lock on Loan table
Number of Locks: One write single lock
Enabling Conditions: Privileged user's login
Frequency: Frequent

Consistency (ACID): Consistency is not critical
Subtasks: Mother task is not needed, no decomposition required.

Abstract Code

- On **<u>Sales Order</u>** form, check if customer has applied for loan
- If customer has applied for loan:
    - Enter loan details on **<u>Sales Order</u>** form. Enter Loan Term, Down Payment, Interest Rate, Loan ID, Monthly Payment and Start Month.
- Else:
    - Do nothing
- Click on ***Save*** button to save the changes.

<u>View Seller History</u>
Task Decomposition
Lock Types: Read-only on Customer, Vehicle and Parts Order tables
Number of Locks: Three read locks.
Enabling Condition: User should have access to look for seller history
Frequency: Around once or twice a day
Consistency: not very critical
Subtasks: Mother Task is not needed. Decomposition is not needed.

Abstract Code

- On **<u>Display Reports Page</u>** when ***View Seller History*** *is clicked*
- Validate if user has access to reports
- Initialise a **seller_history** report with zero rows and five columns seller_detail, total_vehicles, average_purchase_price, average_parts and average_parts_cost
- For every customer
    - If he has sold at least one vehicle
        - Create a display row -> seller_row
        - Add first column to seller_row first name and last name of customer or company name in case customer is business
        - Find list of all the vehicles sold by them as vehicle_list and
        - Add second column to seller_row **total_vehicles ->** size of vehicle_list

        - Calculate the average_purchase_price for all those vehicles and add third column to seller_row **average_purchase_price**

- For every vehicle in the vehicle_list
    - Find all the parts order
    - Calculate **average_parts and average_parts_cost**

- Add fourth column to seller_row **average_parts**
- Add fifth column to seller_row **average_parts_cost**

- If average_parts_cost > $500:
    - Highlight the row background as red
    - Add seller_row to the report

- Sort all rows by **total_vehicles** descending
- Sort all rows by **average_purchase_price** ascending
- Display the **seller_history** report.

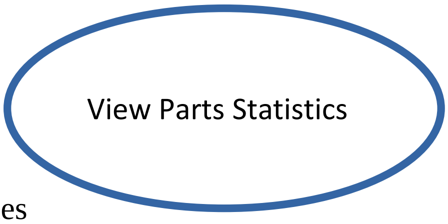
<u>View Monthly Loan Income</u>

Task Decomposition
Lock Types: Read-only on Loan table and Vehicles table
Number of Locks: Two read locks
Enabling Condition: User should have access to look for Loan Details and Vehicle details
Frequency: Frequent
Consistency: not critical
Subtasks: Mother Task is not needed. Decomposition is not needed.

Abstract Code

- On **<u>Display Reports</u>** page, when ***View Monthly Loan*** *is clicked*
- Validate if user has access to reports
- Initialise a **view_monthly_loan** report
- Initialise start month as start_month = current_month - 1 - 12
- Initialise end month as end_month = current_month - 1
- For current_month in range of start_month to end_month
    - initialise total_monthly_payment = 0
    - Find all the vehicles with sellDate in current_month, find date and year
    - For every vehicle find the corresponding loan if exists
        - If startMonth of loan is one less than current month then
            - add MonthlyPayment to total_monthly_payment.

- Calculate 1% of total_monthly_payment -> Mr Burdell's share.
- Display year, current_month, total_monthly_payment, Mr Burdell's share.
- Display **view_monthly_loan** report.

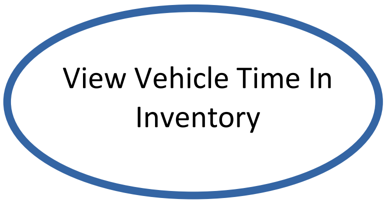


View Parts Statistics

Task Decomposition
Lock Types: Read-only on Vehicles, Parts Order and Vendor tables
Number of Locks: Three read locks.
Enabling Condition: User should have access to look for Loan Details and Vehicle details
Frequency: Very Frequent
Consistency: not very critical
Subtasks: Mother Task is not needed. Decomposition is not needed.

Abstract Code

- On **Display Reports** page, when ***View Monthly Loan*** *is clicked*
- Validate if user has access to reports
- Initialise **view_parts_statistics** report.
- Create a report with zero rows and three columns vendor, total_noOf_parts and total_dollars_spent.
- Find all the vehicles in the inventory -> vehicle_list
- For every vehicle in the vehicle_list
    - For every part order
        - For every part in the order
            - Find the corresponding vendor
                - Add or update the cost of part to value of column total_dollars_spent for the vendor
                - Update the column total_noOf_parts by 1 for the vendor
- Display the **view_parts_statistics** report.



View Vehicle Time in Inventory

Task Decomposition
Lock Types: Read-only on Vehicles
Number of Locks: 1
Enabling Condition: User should have access to look for Vehicle details
Frequency: Frequent
Consistency: not critical
Subtasks: Mother Task is not needed. Decomposition is not needed.

Abstract Code

- On **<u>Display Reports</u>** page, when ***View Vehicle Time In Inventory*** *is clicked*
- Validate if user has access to reports
- Find all the vehicles in the inventory -> vehicle_list
- Group all the vehicles by vehicle type - > vehicle_list_by_type
- For every vehicle_type_group in vehicle_list_by_type
    - initialise total_time = 0
    - initialise total_vehicle_count_in_type = size of vehicle_type_list
    - If total_vehicle_count_in_type is 0 then Display “N/A"
    - Else
        - For every vehicle in vehicle_type_group
        - Calculate time spent in inventory time_in_inventory using purchaseDate and sellDate and update total_time
        - Calculate average_time per vehicle_type using total_time and total_vehicle_count_in_type
        - Display average_time per vehicle_type

<u>View Price Per Condition</u>

Task Decomposition
Lock Types: Read-only on Vehicles
Number of Locks: One read lock
Enabling Condition: User should have access to look for Vehicle details
Frequency: Frequent
Consistency: critical
Subtasks: Mother Task is not needed. Decomposition is not needed.

Abstract Code

- On Display Reports page, when ***View Price Per Condition*** *is clicked*
- Validate if user has access to reports
- Initialise a report **view_price_per_condition** with vehicle types and conditions, types in rows and conditions in columns
- Find all the vehicles in the inventory -> vehicle_list
- Group all the vehicles by vehicle type - > vehicle_list_by_type
- For every vehicle_type_group in vehicle_list_by_type
    - Group all the vehicles by vehicle condition - > vehicle_list_by_condition
    - For every condition_type in vehicle_list_by_condition
        - Find total_vehicle_count and purchase_price for all vehicles
    - If total_vehicle_count == 0 then

        - Update the cell in row for type and column for condition to $0
    - Else
        - Calculate average_price for all vehicles for a type and condition
        - Update the cell in row for type and column for condition with average_price

- Display the **view_price_per_condition** report.

View Monthly Sales

Task Decomposition
Lock Types: Read-only on Vehicles, Privileged User, Parts Order tables
Number of Locks: Three read only locks
Enabling Condition: User should have access to View Vehicle details, Privileged User and Parts order tables
Frequency: Very Frequent
Consistency: critical
Subtasks: Mother Task is not needed. Decomposition is not needed.

Abstract Code

- **On Display Reports page, when *View Monthly Sales is clicked***
- Validate if user has access to reports
- Initialise Monthly_Sales_Report and Monthly_Sales_DrillDown_Report
- Find all the vehicles which have a non empty SalesDate column
- Group all these vehicles by year and then by month and sort in reverse chronological order
- Initialise total_sales_income, total_vehicles, total_net_income for every month, salespeople_list
- For every vehicle in every month
    - Find the PurchasePrice, SalesPrice and sum of SumOfPartsCost from all parts orders for this vehicle
        - add SalesPrice to total_sales_income
        - calculate net income using SalesPrice - (PurchasePrice + sum of SumOfPartsCost)
        - add net_income to total_net_income
    - Find the corresponding SalesPerson and add id to salespeople_list
    - Display total_sales_income, total_vehicles, total_net_income and add to Monthly_Sales_Report
- For every month
    - Find details of PrivilegedUser using ids from salespeople_list

    - From the vehicles table, group them using no_of_vehicles sold and by the total_sales of vehicles
    - For every user in salespeople_list
        - Display FirstName, LastName, no_of_vehicles and total_sales and update Monthly_Sales_DrillDown_Report
    - Sort by no_of_vehicles descending and update Monthly_Sales_DrillDown_Report
    - Sort total_sales descending and update Monthly_Sales_DrillDown_Report
    - Display Monthly_Sales_DrillDown_Report
- Display Monthly_Sales_Report

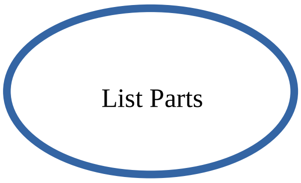
View Vehicle Details

Task Decomposition
Lock Types: Read-only on Vehicles
Number of Locks: One read only
Enabling Condition: User should have access to View Vehicle details
Frequency: Frequent
Consistency: not critical
Subtasks: Mother Task is not needed. Decomposition is not needed.
Abstract Code

- After login when user searches for vehicle by using either of vehicle type, manufacturer, model year, color or any keyword.
- If the vehicle matches to any of the vehicle's field**:**
    - List out all the vehicles matching the search criteria used by a user in ascending order of VINs.
    - Display following attributes for each of the vehicle:

      VIN, vehicle type, model year, manufacturer, model, color(s), Mileage, SalesPrice.
    - Provide user an option for sorted list of vehicle as per users choice of attribute.
    - If user selects any specific vehicle result:
        - Open details page for of the vehicle
        - Under the details page, display following vehicle details:
            - VIN, vehicle type, model year, model name, manufacturer, color(s), mileage, sales price, description of the car.

List Parts

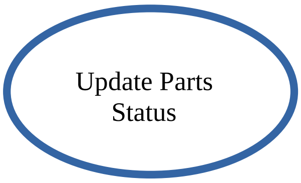


Task Decomposition
Lock Types: Read-only on Parts
Number of Locks: One read only lock
Enabling Condition: User should have access to list parts for a vehicle.
Frequency: Frequent

Consistency: not critical
Subtasks: Mother Task is not needed. Decomposition is not needed.
Abstract Code

- Show parts details of a vehicle for an Inventory clerk in view vehicle details page.
- If the car is newly added:
  - Show $0 total for parts.
- If user clicks on Parts section in the vehicle details page:
  - For each part show details like part number, description, vendor, purchase order, cost and status.
- After login when user searches for vehicle by using either of vehicle type, manufacturer, model year, color or any keyword.
- if the vehicle matches to any of the vehicle's field:
  - List out all the vehicles matching the search criteria used by a user in ascending order of VINs.
  - Display following attributes for each of the vehicle:
    - VIN, vehicle type, model year, manufacturer, model, color(s), Mileage, sales price.

- Provide user an option for sorted list of vehicles as per users choice of attribute.
- If user selects any specific vehicle result:
  - Open details page for of the vehicle
  - Under the details page, display following vehicle details:
  - VIN, vehicle type, model year, model name, manufacturer, color(s), mileage, sales price, description of the car.



Update Part Status

Task Decomposition
Lock Types: Read-Write on Parts
Number of Locks: One read-write lock
Enabling Condition: User should be able to update part-status for a vehicle.
Frequency: Frequent
Consistency: critical
Subtasks: Mother Task is not needed. Decomposition is not needed.

Abstract Code

- If user clicks onto view parts section on view vehicle details page.
  - For each part show details like part number, description, vendor, purchase order, cost and status.

- If user clicks on edit status button:
  - User should be able to edit the status and save it.