**Flow chart Design**

Start

Choose Operation region

finish Fueling

Start Fueling

Go the next stop

Choose the next stop

False

True

Refill

fuel level < 100%

Check the fuel level

Calculate how many Gallons was delivered

True

false

End

True

Check the current time

During work hours

Check the fuel level

fuel level <= 17%

False

**Fuel pseudo-code design**

* Start

1- Check the fuel level before moving from the base.

* + If the fuel level is less than 100% of the total capacity of the truck
* Instruct to refill.
  + Else
    - Go to step2

1. Choose operating region.
2. Select the next stop.
3. Go to the next stop.
4. Start fueling.
5. Finish fueling.
6. Calculate how many Gallons was delivered to this stop.
7. Check the time
   * If the time during work hours
     + - Go to step 9
     + Else
       - End.
8. Check the fuel level

* If the fuel level less than 17% of the total capacity of the truck
  + Return to the base to refill
* Else
  + Go to step 3

1. Go to step 3

**Requirements**

* + - **List class definitions for any objects needed in the dispatcher application.**
  + **﻿** public class **Truck**

{

public int truckId { get; set; }

public string driverName { get; set; }

public int plateNumber { get; set; }

}

* ﻿public class **OperatingRegion**

{

public int regionId { get; set; }

public string regionName { get; set; }

}

* public class **Customer**

{

public int customerId { get; set; }

public int regionId { get; set; }

public string customerName { get; set; }

public string customerAddress { get; set; }

public double maxNoOfGallons { get; set; }

}

* ﻿public class **DeliveryEvent**

{

public int deliveryEventId { get; set; }

public int truckId { get; set; }

public DateTime eventDate { get; set; }

}

* ﻿public class **DeliveryStop**

{

public int deliveryStopId { get; set; }

public int customerId { get; set; }

public int deliveryEventId { get; set; }

public double noOfGallons { get; set; }

}

* **List web service method signatures that the truck will call to get or post information to the dispatcher.**
* Method for Average fuel consumption per delivery, listed by “Stop” for the month.

**﻿**public IEnumerable<dynamic> GetAvgFuelConsumptionPerCustomer()

{

using (var Connection = new SqlConnection(ConfigurationManager.ConnectionStrings["FuelDelivery"].ConnectionString))

{

Connection.Open();

var result = Connection.Query("Select C.customerName as 'Customer Name',"+

"AVG(DS.noOfGallons) as 'Average Fuel Consumption'"+

"from FuelDelivery.dbo.DeliveryStop DS "+

"left Join FuelDelivery.dbo.customer C "+

"ON DS.CustomerID = C.customerId "+

"left join FuelDelivery.dbo.DeliveryEvent DE "+

"ON DS.deliveryEventId = DE.deliveryEventId "+

"WHERE MONTH(DE.eventDate) = Month(GETDATE()) AND YEAR(DE.eventDate) = YEAR(GETDATE()) "+

"GROUP BY C.CustomerName "+

"ORDER BY 'Customer Name'");

return result.ToList();

}

}

* Method to add the data for a new stop. The dispatcher will use this method to add the data to the system after each stop,

﻿ public int AddNewStopSata(DeliveryStop deliveryStop)

{

using (var Connection = new SqlConnection(ConfigurationManager.ConnectionStrings["FuelDelivery"].ConnectionString))

{

Connection.Open();

var result = Connection.ExecuteScalar<int>(@"INSERT INTO[dbo].[DeliveryStop] ([customerId],[deliveryEventId],[noOfGallons])

VALUES (@customerId, @deliveryEventId, @noOfGallons)"

, deliveryStop);

return result;

}

}

* + - **Provide a high-level outline for how these classes will be used to maintain a current “state”.**

**﻿** o the ‘DeliveryStop’ class will have all the details that we need to know in each stop, like the customer name, how many Gallons delivered to this customer. Also, it will take the truck information from the ‘*DeliveryEvent*’ class, like the truck plate number and service date.

o We will use the ‘DeliveryStop’ table join with the other tables to get the information about the consumption for the customers in certain months

**- Describe how we will store and retrieve information from a SQL database for a recorded fuel stop. Provide a minimal two table definition of “Stops” and “Fuel delivery event”**

* We have five tables:
  + *Truck*: has data about the truck, like plate number, and driver name.
  + *OperatingRegion*: has data about the different regions like the region name.
  + *Customer*: has data about the customer like the name, belong to which region and the max capacity.
  + *DeliveryEvent*: has data about which truck will go to deliver the gas and the date.
  + *DeliveryStop*: has data about each stop, like the customer name, how many gallons it delivered to this customer and the DeliveryEventId used.
* After the truck finishes fueling at the first stop, the dispatcher will calculate how many gallons delivered to this stop- we will insert a new row in the ‘DeliverStop’ table. So, here we will have a record that has data about the name of the customer, how many gallons, and the date of delivery.
* When we need to retrieve information about which customer(stop) delivered and how many gallons at which date we will select from the ‘DeliverStop’ table.
* **Demonstrate delivery reporting capability by writing two SQL queries to show the following:**
* **Top 10 fuel consuming “Stops” over the past 12 months**.

﻿

Select Top 10

C.customerName as 'Customer Name',

SUM(DS.noOfGallons) as 'Total fuel consuming'

from

FuelDelivery.dbo.DeliveryStop DS

left Join

FuelDelivery.dbo.customer C

ON

DS.CustomerID = C.customerId

left join

FuelDelivery.dbo.DeliveryEvent DE

ON

DS.deliveryEventId = DE.deliveryEventId

WHERE

DATEDIFF(MONTH,DE.eventDate, GETDATE()) <= 12

GROUP BY

C.CustomerName

ORDER BY

'Total fuel consuming' DESc

* **Average fuel consumption per delivery, listed by “Stop” for the month.**

﻿

Select

C.customerName as 'Customer Name',

AVG(DS.noOfGallons) as 'Average Fuel Consumption'

from

FuelDelivery.dbo.DeliveryStop DS

left Join

FuelDelivery.dbo.customer C

ON

DS.CustomerID = C.customerId

left join

FuelDelivery.dbo.DeliveryEvent DE

ON

DS.deliveryEventId = DE.deliveryEventId

WHERE

MONTH(DE.eventDate) = Month(GETDATE()) AND YEAR(DE.eventDate) = YEAR(GETDATE())

GROUP BY

C.CustomerName

ORDER BY

'Customer Name'