Semester Term: **Fall 2021**

**Team Project:**

**Team Name: Group 4**

**Team Members with Roles:**

* Aaron Caldwell – Designer/Developer - Product
* Stephanie Delgado – Designer/Developer - Client
* Raul Gutierrez – Designer/Developer – Sales Rep
* Michael Nguyen – Designer/Developer – SalesHistory, Sales, Main menu

**Project Design:**

**Project Scope:** *(Review how you will satisfy the program requirements)*

Program will read client, sales representatives, product, and sales history from files. This new information will be saved into the read file. The program will be able to list specific or all information from respective classes. Sales bonus and monthly sales report will be calculated from given information. Ability for client to purchase a product will be included.

**Program Organization**: *(How are you organizing your program and data to make the program work?)*

Clients, Sales Representatives, Product/Service, and Sale are classes. Main Program will be used to test all the classes and class functions, including reading and writing client sales files.

**Project Details:** *(What data are you storing and what functions will you use to make the program work?)*

Company Name: Dinder Mufflin Product: office supplies.

Stored Data:

Client information (name, short address & sales to date).

Sales representatives information (name, short address, sales to date, monthly sales, sales bonus).

Product (name, sales, etc.).

Sales history (sales, sales from client).

Functions:

Clients - GetClientName, GetClientAddress, GetClientSales, PrintAllClients, PrintClientInfo, AddNewClient, UpdateClient, SaveClient, SetClientName, SetClientAddress, SetClientSales, ClientMenu

Sales Representatives - SetRepName, SetRepAdress, SetSalesToDate, GetRepName, GetRepAddress, GetSalesToDate, MonthlySalesList, NewSalesRep, SalesRepList, UpdateRepInfo, SalesRepMenu, SaveRepInfo

Product – PrintAllProducts, GetProductSales, PrintProductInfo, AddNewProduct, ProductMonthlySalesReport, UpdateProduct, SaveProduct, GetProductInfo, SetProductInfo, GetProductName, SetProductName, SetProductSales, ProductMenu

Sales – SalesHistory, SalesHistoryMenu, PurchaseProduct, PrintSaleHistory, PrintSaleHistoryTitles, PrintAllSalesHistory, PrintClientSalesHistory, UpdateSales, CheckNumber, CheckDate, GetSaleInfo, Sales, SetSaleID, GetSaleID, SetDate, GetDate, SetClientID, GetClientID, SetProductID, GetProductID, SetProductSales, GetProductSales, SetSalesRepID, GetSalesRepID

**Client View:** *(What does the client see when he/she runs to the program?)*

Can see options to look at client, sales representatives, product, and sales information menus.

Client Information – list of clients, specific client information, option to add or update a client

Sales Representatives – list of sales representatives, specific sales representative information including

sales bonus, option to add or update a sales representative

Product – list of products, specific product information, option to add or update a product, monthly

sales report

Sales – option to purchase a product, all sales information, sales from a specific client,

option to update a sale

**Algorithm** (flowchart and/or pseudo-code):

Main:

loops through menu options for client, sales representatives, sales, and products

executes a respective menu function depending on user input

end program when user enters ‘q’

Clients Class:

GetClientName: return clientName

GetClientAddress: return clientAddress

GetClientSales: return clientSales

PrintAllClients: loop through clients in file, PrintClientInfo

PrintClientInfo: GetClientName, GetClientAddress,GetClientSalesToDate, print them

AddNewClient: creates new client from user input (name, address, sales)

UpdateClient: SetClientName, SetClientAddress, SetClientSalesToDate, SaveClient

SaveClient: saves new data to file

SetClientName: set clientName

SetClientAddress: setClientAddress

SetClientSales: set ClientSales

ClientMenu: prints options that users can do, executes function from chosen option

Sales Representatives Class:

SetRepName: set repName

SetRepAddress: set repAddress

SetSalesToDate: set SalesToDate

GetRepName: return repName

GetRepAddress: return repAddress

GetSalesToDate: return SalesToDate

MonthlySalesList: print sales depending on month

NewSalesRep: creates new sales rep object with info from user input

SalesRepList: loop through representatives file, print info

UpdateRepInfo: SetRepName, SetRepAddress, SetSalesToDate, SaveRepInfo

SalesRepMenu: prints options that users can do, executes function from chosen option

SaveRepInfo: saves new data to file

Product Class:

PrintAllProducts: loops through product file, PrintProductInfo

GetProductSales: return productsSales

PrintProductInfo: GetProductInfo, GetProductName, prints them

AddNewProduct: create new product object with info from user input

ProductMonthlySalesReport: Calculates and prints monthly sales report

UpdateProduct: SetProductInfo, SetProductSales, SetProductName, SaveProduct

SaveProduct: saves new data to file

GetProductInfo: return productInfo

SetProductInfo: set productInfo

GetProductName: return productName

SetProductName: set productName

SetProductSales: set productSales

ProductMenu: prints options that users can do, executes function from chosen option

Sales History Class:

SalesHistory: constructor, opens salehistory file and puts data into vector

SalesHistoryMenu: loop menu, prints options for purchasing, updating, looking at all sales, looking at client sales. executes function from chosen option

PurchaseProduct: GetSaleInfo(), writes new info into sale history file

PrintSaleHistory: Prints sale from file with formatting

PrintSaleHistoryTitles: Print titles for each sale information category with formatting

PrintAllSalesHistory: PrintSaleHistoryTitles(), loop PrintSaleHistory(currentSale)

PrintClientSalesHistory: check if client exists, PrintSaleHistoryTitles(), PrintSaleHistory(currentSale)

UpdateSales: check if sale exists, GetSaleInfo(), write new info into sale history file

CheckNumber: loop for checking if user input is a number

CheckDate: loop for checking if user input is a date

GetSaleInfo: Gets user input for each sale category by using checkNumber()

Sales Class:

Sales: constructor for setting saleID, date, clientID, productID, productSales, SalesRepID

SetSaleID, SetDate, SetClientID, SetProductID, SetProductSales, SetSalesRepID: sets respective data

GetSaleID, GetDate, GetClientID, GetProductID, GetProductSales, GetSalesRepID: gets respective data

**Project Code/Implementation:**

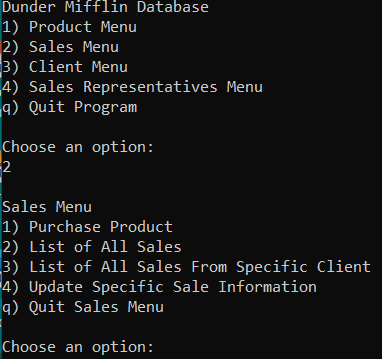
**Source Code Listing:** *(Text listing not image of source code)*

**Project Test Plan:** *(How can you prove your program works all the time?)*

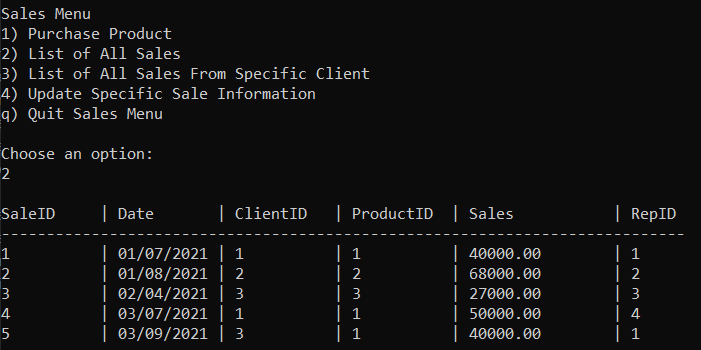
Test each function by seeing if entered input correctly outputs data.

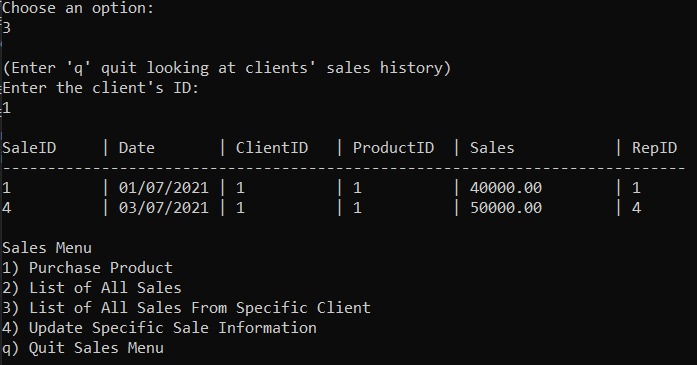
**Screen Shots of Running Program:** *(Images of the program output very to completion.)*

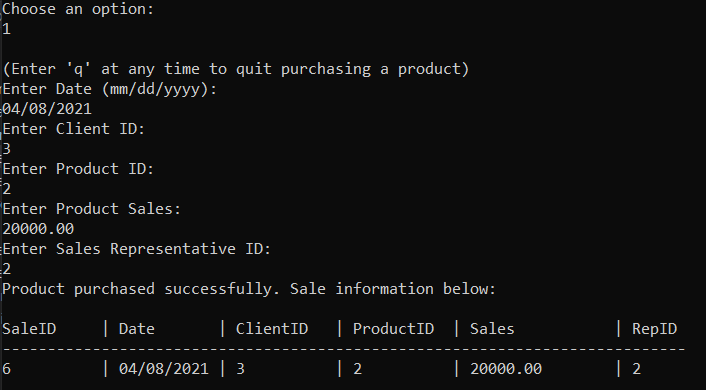
Main Menu Screenshots:

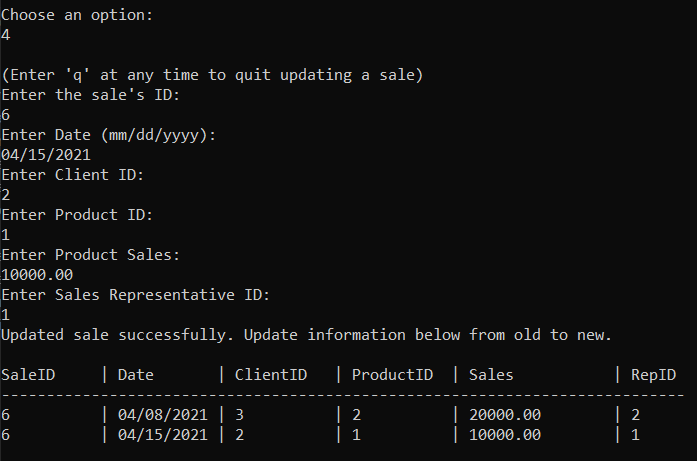


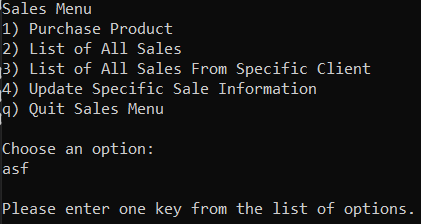
Sales Menu Screenshots:

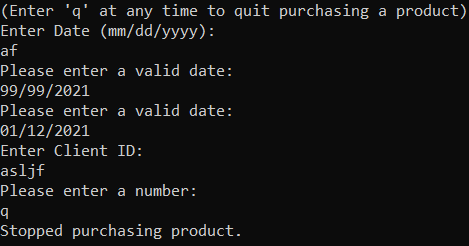


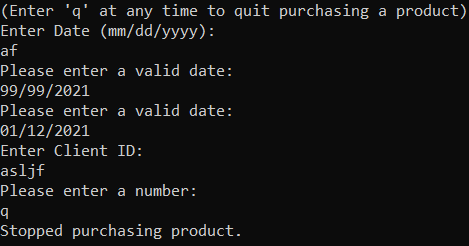


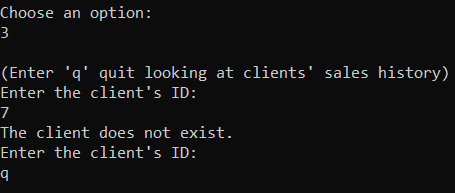


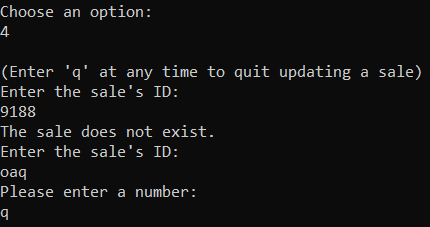




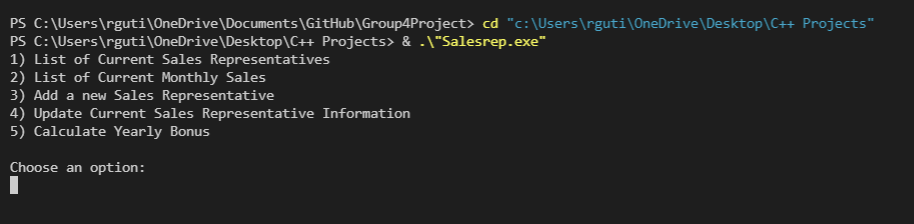


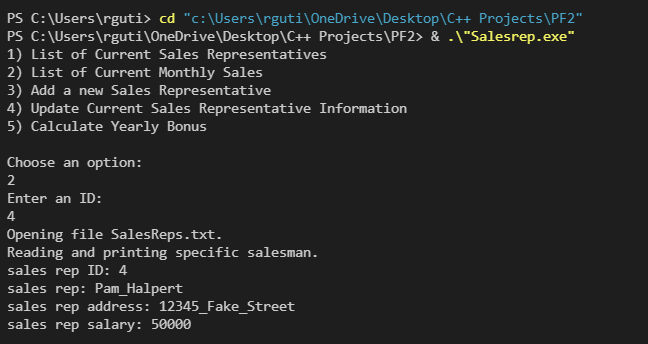


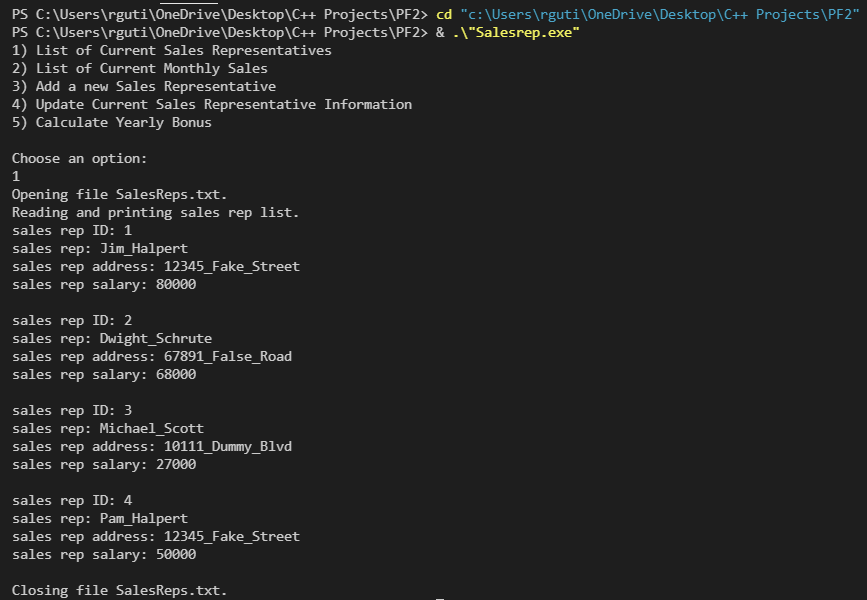


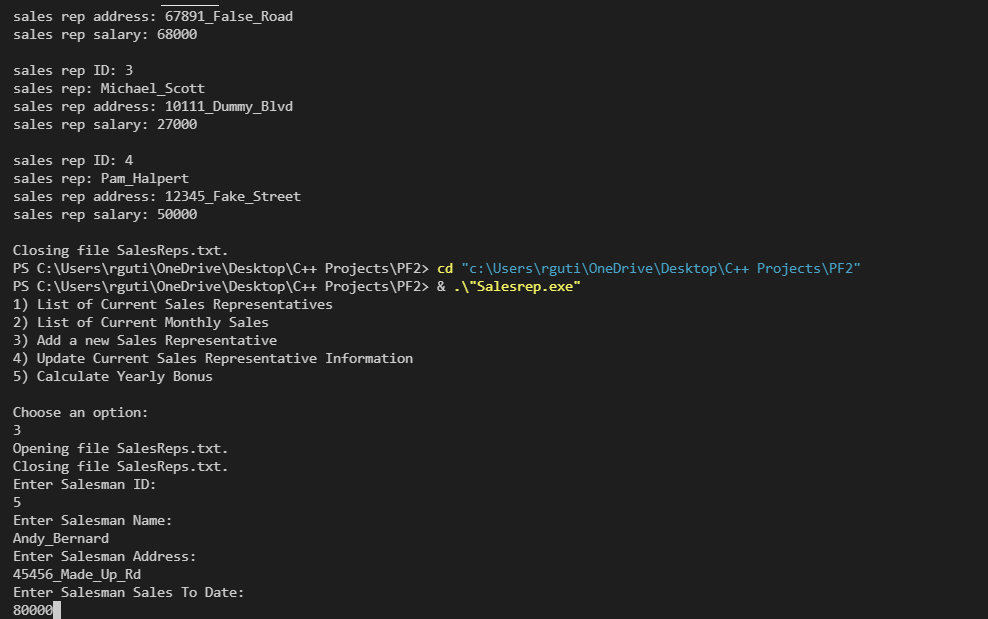


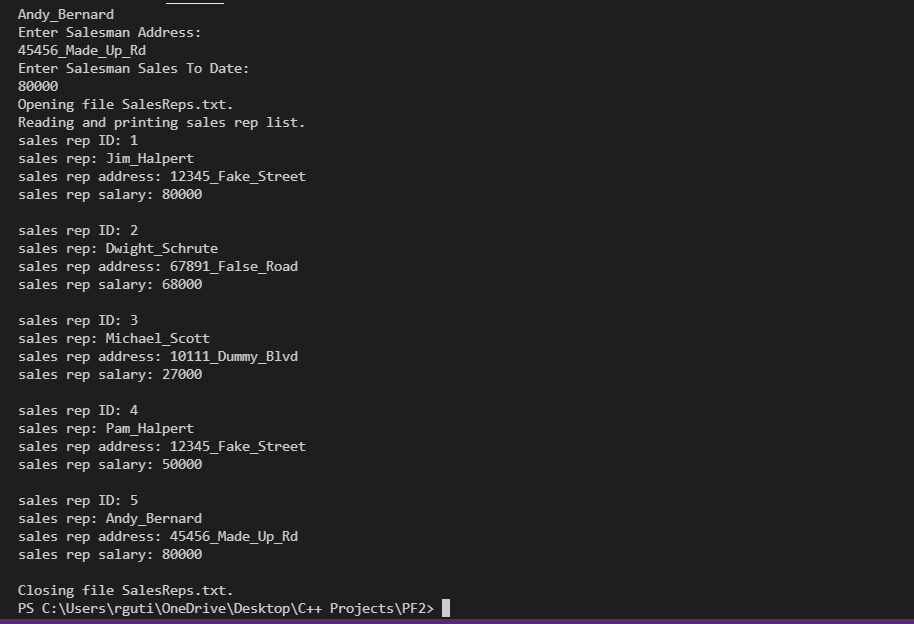
Sales Rep Menu Screenshots:











Product Menu Screenshots:

