Statement of Completeness (Template): Mates Rates Rent-a-Car

This statement of completeness will need to *accurately* state the functionality which has been implemented. Use this checklist of program functionality as you complete your project.

Student Name: John Nguyen **Student ID:** N10132767

Functionality Checklist

In the following section, you are required to mark which functionality you have implemented. In the column on the right please mark 'Y' where you have completed this functionality, and 'N' where you have not. Please fill in any additional text boxes requested, and please note any limitations or bugs in the box at the end of each section. You may expand the table if you need more room for comments.

File I/O		
File location	*If your solution does not load files from the specified location in the specification, please note here where the files are stored.* **Customers:** Data/customer.csv **Fleet:** Data/fleet.csv **Rentals:** Data/rentals.csv	
Read	CRM	Y
	Fleet	Y
	Rentals (must meet above criteria first)	Y
Write	CRM (must be able to read CRM first)	Y
	Fleet (must be able to read CRM first)	Y
	Rentals (must meet above criteria and read Rentals first)	Y
Comments	When passing in command line arguments to load the files, the data folder is 3 directory nests relative to the executable program. E.g. parse fleet.csv will require "../../../Data/fleet.csv" as one of the three command line arguments	

Please note that your assignment must be able to load the files in the same format as provided in the example files.

CRM And Fleet Functionality		
CRM	View customers	Y
	Add customer - With validation – all fields are valid options, no repeated customer ID	Y Y
	Remove customer - With validation – only if not renting vehicle	Y Y
	Edit customer - With validation – same as add	Y Y
Fleet	View vehicles	Y
	Add vehicle - With validation – all fields are valid options, and there is no repeated registrations	Y Y
	Remove vehicle - With validation – only if not already being rented	Y Y
	Edit vehicle - With validation – same as add	Y Y
	Extra marks – implemented vehicle classes with inheritance Please include a brief explanation of the inheritance here if implemented: For the vehicles class. There is 4 derived class named FamilyVehicle, EconomyVehicle, LuxuryVehicle and CommercialVehicle. These classes take in a default grade enumeration, and provides specific defaulted attributes in the constructor of sub-class vehicle type.	Y
Renting	View rented vehicles report. This should show the vehicle, customer, and the daily cost of the rental.	Y
	Rent vehicle. This should show the total cost of the rental. - With validation – only if vehicle and customer exist, vehicle is not already being rented, and customer is not already renting	Y Y
	Return vehicle. - With validation – only if vehicle was being rented by the customer	Y Y

Comments	Small bug when displaying fleet table in the console using default console window dimensions. When modifying the Registration or ID of a vehicle or customer, the operator can still choose to retain the new registration/ ID fields as the old value, as it does not compromise uniqueness constraints. E.g., customer old ID 10 - > new ID 10 won't have any duplicates in the customers list.
Changes from spec.	There is an additional table and border menu class which allows the operator to choose a different table and border style when displaying the csv files to the console.

Search Functionality

Possible to Query - Simple	Any - Be able to see any unrented vehicle in the fleet	Y

	Single attribute query -
	    Example:
	        o `Red`
		Y

	A choice between two attributes (disjunction) -
	    Example:
	        o `Family OR Luxury`
		Y

	A combination of two attributes (conjunction) -
	    Example:
	        o `Family AND Luxury`
		Y

Intermediate	A choice of any number of attributes (disjunction) -
	    Example:
	        o `Family OR Luxury OR Red`
		Y

	A combination of any number of attributes (conjunction)
	   - E.g.:
	        o `Luxury AND Red`
		Y

Advanced	A combination of any number of attributes, using both `AND` and `OR`, where the operators `AND` and `OR` should have the same priority (precedence) - E.g.: o `Economy OR Family AND 4-Cylinders`	Y

Ninja	A combination of any number of attributes, using `AND` and `OR`, with `AND` having a higher priority than `OR`, supporting parenthesis to resolve priority - E.g.:
	        o `((GPS AND Sunroof) OR (Red OR Green))`
	          `AND Commercial OR Luxury`
		Y

Comments		

Program Transcript

You are also required to attach a full transcript demonstrating all of the features of your program (you can use the checklist above as a basis). You will practice this in a future workshop. You must include this transcript as part of your submission. Please state the name of the transcript file.

Transcript File Name: n10132767Transcript.txt