README

Project: *Restaurant Management System*, using *"Hello World"* starter code.

Additional libraries: None

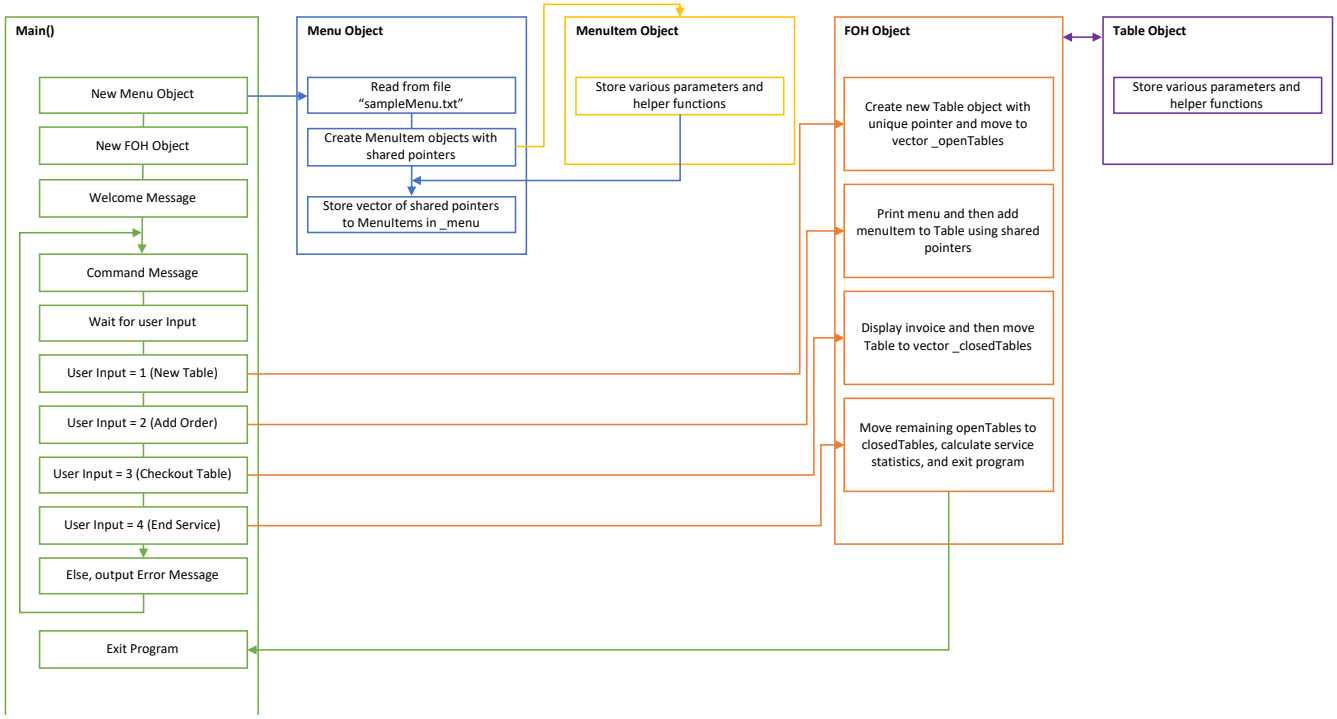
Building/Running:
Enter the following into the console:

mkdir build
cd build
cmake ..
make
./RMS

File and Class Structure || Expected Behavior

Overview:
This program utilizes object oriented programming and memory management. *Restaurant Management System* allows the user to add new guests to a table, add orders from a menu to tables, and checkout tables (which displays the invoice and frees up that table). When service is ended, a summary of the service statistics is shown.

Code Structure:



Rubric Points

Category	Criteria	Code Location(s)
Loops, Functions, I/O	The project demonstrates an understanding of C++ functions and control structures.	*FOH.cpp, Menu.cpp, MenuItem.cpp, Table.cpp, Main.cpp*
	The project reads data from a file and process the data, or the program writes data to a file.	*Menu.cpp*, Line 11 - 42
	The project accepts user input and processes the input.	*Main.cpp*, Line 57
Object Oriented Programming	The project uses Object Oriented Programming techniques.	*FOH.h, Menu.h, MenuItem.h, Table.h*
	Classes use appropriate access specifiers for class members.	*MenuItem.h,* Line 8 - 18
	Class constructors utilize member initialization lists.	*MenuItem.h*, Line 9
	Classes abstract implementation details from their interfaces.	*MenuItem.h*, Line 13
	Classes encapsulate behavior.	*Table.h,* Line 13 - 39
Memory Management	The project makes use of references in function declarations.	*FOH.cpp*, Line 91
	The project uses move semantics to move data, instead of copying it, where possible.	*FOH.cpp*, Line 179
	The project uses smart pointers instead of raw pointers.	*FOH.cpp*, Line 110