**Screenshots:**

**Part 1**

Inputting and displaying five customers



**Part 2**

**1. A high-level narrative description of the algorithms you will use to determine the grid**

**selection when it is your turn (step #4 above):**

The algorithm will need to incorporate both random selection as well as intelligent selection. Initially, selections will be more random as the opponents grid is explored. However, once a ship is hit, the algorithm should make more intelligent decisions and prioritize the neighboring spaces for future turns. The algorithm should be able to adapt based on the results of previous turns.

**2. Include more detailed pseudocode that describes specific parts of the algorithm. Note**

**that all players are informed of all guesses and results (step #5 and #6 above) and**

**you may (should?) store and use that information as part of your algorithm.**

One part of the algorithm would be to ensure that previous results are analyzed so that more educated guesses can be made as the game continues:

function makeGridSelection():

if initialTurn:

return randomGridSelection()

else:

return strategicGridSelection()

function strategicGridSelection(Grid opponentGrid)

if previousMove = “Hit”

if neighboringSpots = clear

return x, y coordinates

function randomGridSelection()

return random x,y coordinates

**3. Describe the data Structures you will need to store any data as part of those algorithms.**

We will need a 2D array that represents the player’s grid, a way to record each move’s grid selection as well as the outcome, and a way to maintain the status of each ship.

**4. Include a high-level OO design for the classes you will need to incorporate the**

**algorithms and data structures. A class diagram will be acceptable as will an outline**

**of the class definitions in a Visual Studio project (i.e., empty classes with no code**

**inside the methods).**









**5. Define the APIs/Entry points for the code in specifications 4/5/6. Specifically, what**

**information would you expect to be passed, returned for you to be able to implement**

**your algorithm. Note that this may require you to add some data structures to item**

**#2. Justify why you need any parameters passed in and why you think the return**

**values would be needed.**

The grid selection function would be called from the main game method, from inside a loop that continues the game until it is over. The grid selection method is part of the player class which means that it has access to all of the properties of the player class, such as the previous coordinates and the player grid. In the function we would use the previous coordinates and the result of the previous coordinates to determine what the next move would be for the algorithm. This would have to return the coordinate selected by the algorithm.