COSC 330 Project 1 Network Battleship Game Due: 10/7/2022

You will be developing a Battleship game that can be played by two players across the network. See the instructor's web page for the detailed functional requirements in the game instruction manual for the game and the networking Java code. A non-functional requirement is that in your design of the game, MVC design pattern must be used.

Besides supporting all game play rules, your game need also to satisfy the following functional requirements:

1. Colours and sound must be used in your application (e.g., red for attack shot and white for missed shot on the target grid, red for attack on ocean grid, cheers sound for a hit and crying sound for a miss, use your imagination here)
2. You need to use different symbol or shape for each ship in the game.
3. You should allow a player to set up the battleship manually or automatically on the ocean grid. Manual set-up of the ships must be done using drag and drop.
4. You should allow the player to fire a shot by clicking on a grid of the target grid and update the grid with red/white after you receive the result (hit/miss) from your opponent.
5. Instead of letting player to decide whether it is a miss or hit, or a ship has been sunk as in a traditional battleship game, the program should automatically detect a miss or a hit or a ship has been sunk, and send the result back to the opponent.
6. Display the current status of the game (which battleships get sunk for both you and your opponent).

Helpful tips:

1. If you have any questions regarding the given networking code, you can read Section 24.4, 24.5, and 24.6 from *Java How to Program* by Deitel and Deitel.
2. If you have any questions regarding the drag and drop functionality, check out the instructor's web page.

Deliverables

The project is to be delivered (submitted) in three stages:

Stage I: Design of your application using UML diagrams (to be submitted via myClass) before 9/18/2022.
Stage II: Complete the implementation of networking and drag and drop functionalities (to be submitted via myClass and demonstrate to the instructor) by 9/25/2022.
Stage III (final stage): All your source code tested (to be submitted via myClass and demonstrate to the instructor) by 10/7/2022.