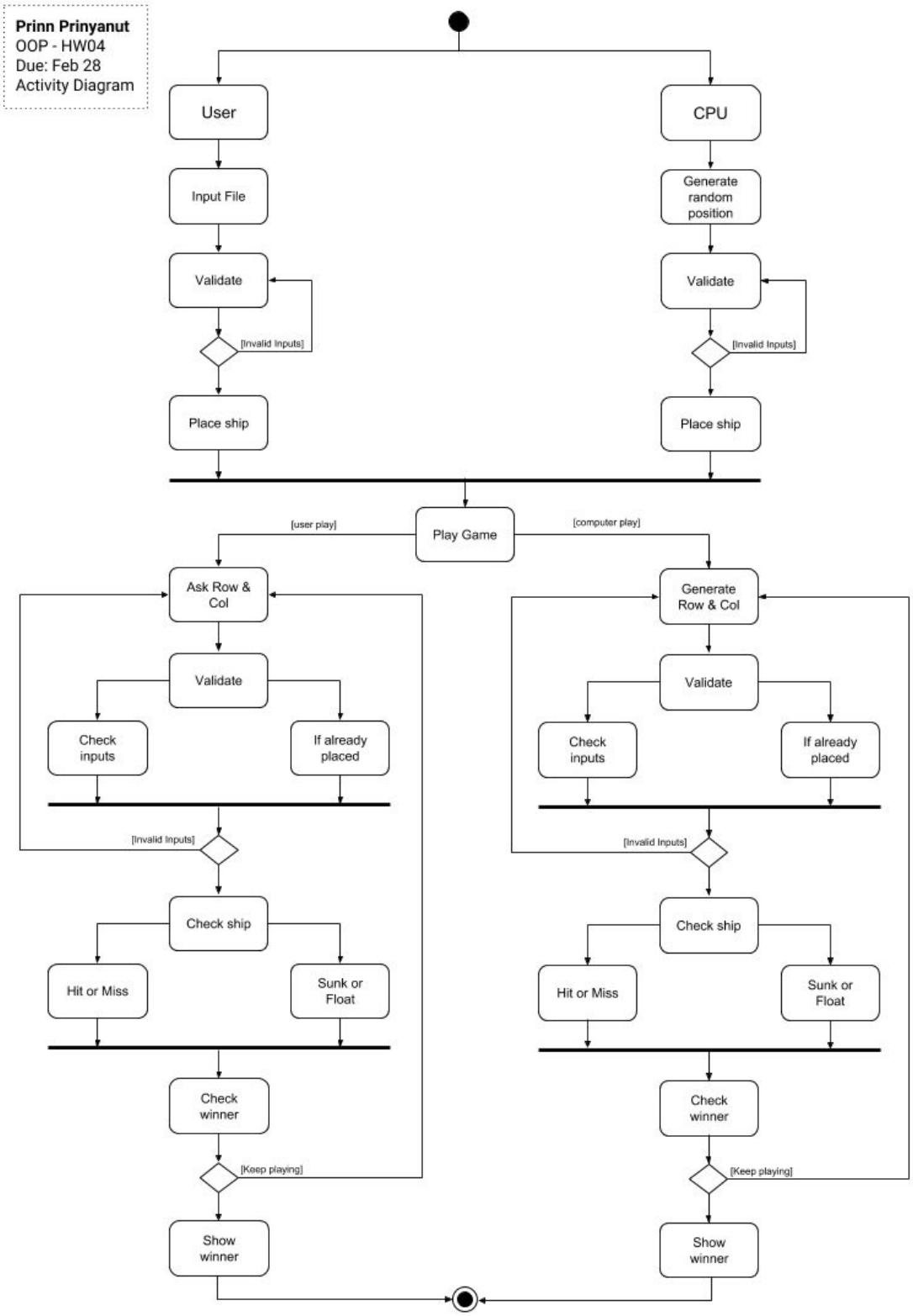
CSCI 2312 - Design Document

1. **Title**
    a. Battleship Game
    b. Prinn Prinyanut
    c. Feb 28 2018

2. **Program short description**
    a. This is a game of battleship where user is playing against a computer. User will place ships by reading data from file. The computer will generate random ship placement. Once both ship placements are complete the game will start. Each play has to take turn to drop their torpedo shots. Whoever sinks all of their opponent ships first is the winner.

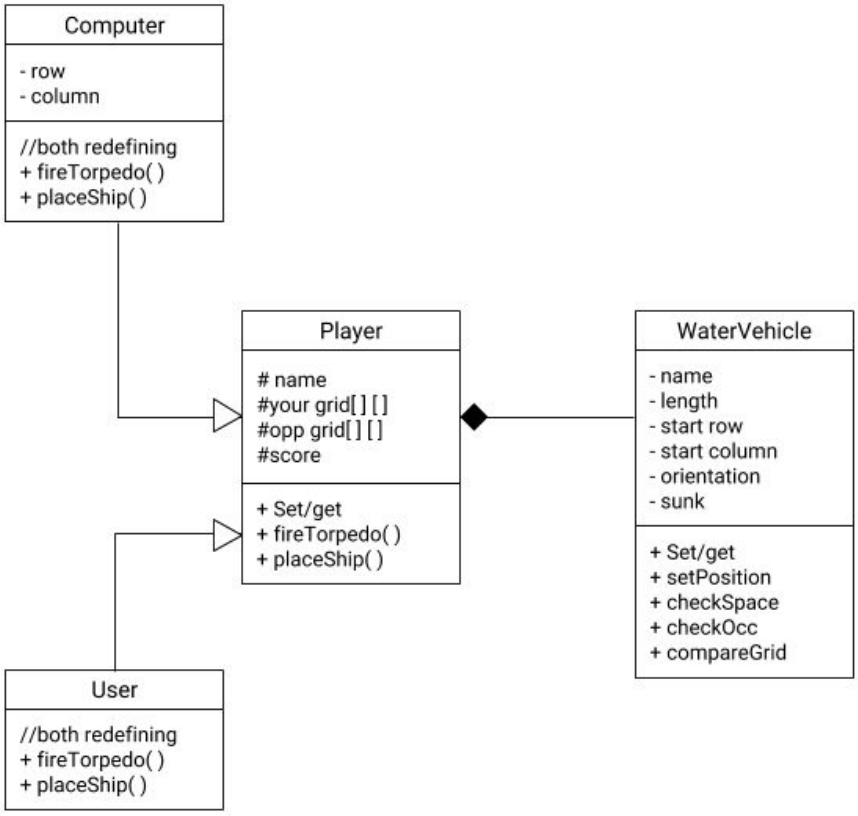
3. **Overall Software Architecture**
    a. In main, 2 objects will be created from both player1 and computer class. Function for placing ships will be called for both of them. Then print a grid for computer and another for player. Function that asks user for torpedo placement will be called. Then in the background it will check weather there's a ship there or not. Then behind the scene the computer will generate random torpedo shots. The checking if hit or miss will also be called for computer. Then the function that checks for winner will be called. If there's no winner the game continues. Lastly the print grid function for both will be called and print result to the console. Also, the user always have the option to exit the game before placing any torpedo shots.

Figure 1 – ACTIVITY (Flow) DIAGRAM



a. The program starts off with user and computer
b. User
    i. Need to read in from input file to place ship
c. Computer
    i. Need to generate random position on grid to place ship
d. Start game
    i. User
        1. Need to ask for row and column input for torpedo shots
        2. Check on computer's grid if it's a hit or miss
        3. Check if user sinks all computers' ships
    ii. Computer
        1. Need to generate random row and column for torpedo shots
        2. Check on user's grid if it's a hit or miss
        3. Check if computer sinks all users' ships
    iii. Check for winner
e. Game end
    i. Ask if want to play again

Figure 2 - Class Diagram



a. Base class is called Player
    i. It has a WaterVehicle class
b. User Class is a Player
c. Computer Class is a Player
d. Player Class will have all of the members that both user and computer will inherited from.
e. User will redefine both fireTorpedo and placeShip functions
f. Computer will redefine both fireTorpedo and placeShip functions

4. **Input Requirements**
    a. CSV files
        i. Range
            1. Row: 1 - 10
            2. Column: A - J
            3. Ship length <= 10
    b. User's torpedo shots input
        i. Range
            1. Row: 1 - 10
            2. Column: A - J
    c. If user wants to quit the program before game ends
        i. Press: q

5. **Output Requirements**
    a. Print out 2 grids
        i. User grid
            1. User ship placement
            2. Computers' torpedo shots
                a. O for hit
                b. X for miss
        ii. Attack grid
            1. Users' torpedo shots
                a. O for hit
                b. X for miss

    b. Ask if user wants to quit

6. **Problem Solution Discussion**
    a. First the program will start with user player and computer player. The user will be reading in ship positions from a csv file, and the computer will be generating random ship positions. Each player will then place ships on their grid. Once that is complete, the game will start. Each player will be taking turns firing torpedo shots, and every user's turn he/she will have an option to quit the game or continue to play. Every turn, 2 grids will display on the console. One for user grid and another is the attack grid. Each turn both grids will update, with O as hit ship and X as miss ship. Every turn the program will check if all ships has been sunk and determine the winner.

7. **Classes, Inheritance, and Data Structures**
    a. I will have Player as my base class because there will be 2 players playing this game. One is user and another is the computer.
    b. Player does not inherited WaterVehicle class but it will have WaterVehicle as a member. I choose to do this because both player will eventually need its own WaterVehicle. Since both user and computer will inherit player, this is the best option to go with.
    c. The reason why I didn't want each ship to have its own class and each inherit WaterVehicle class is because it's unnecessary. The only difference between all ship are just its name and length, which is why I think it is not worth it to create 5 new classes just for that.
    d. For the grid itself I will be using 2d array because I know that the grid is 10x10. Which is why vector or any other data structure won't be necessary.