# Battleship

Architectural Design

1. **Introduction and Context**

*The project that we built is a multiplayer form of the board game battle ship consisting of two players. Essentially, players use a lobby to join a game then each player in a game take turns trying to shoot down other players boats on a specified sized board by choosing different coordinates to shoot at.*

1. **Users and their Goals**

*Users are players of the game, and their goal is to find all their opponents ships locations.*

1. **Major Components and Their Relationship**

The following outlines how three different applications operate and their purposes and the communication sub-system layer.

**Game Manager:**

The Game manager waits for the Lobby to send it two players. It communicates with players to change the game data. Each player communicates to the players via TCP.

**User Client:**

The User Client allows players to connect with the lobby and receive game results from the game manager. The user client handles the user interface.

**Lobby:**

The Lobby connects with the User Clients and Game Managers then the Lobby connects two players ready to play a game with a designated game manager.

***Communication Sub-System:***

*There are three active treads; The dispatcher, UDP communicator, and TCP communicator.*

* + *UDP Communicator:* 
    - *Receives messages from remote application, put those messages in an envelope and put that envelope in the systems queue.*
  + *TCP Communicator:* 
    - *Receives messages from remote application, put those messages in an envelope and put that envelope in the systems queue.*
  + *Dispatcher:*
    - *It takes envelopes from the incoming queue and checks the conversation dictionary if it belongs to a conversation that has been already made. If a conversation has not been made, then the dispatcher will use a conversation factory to create a new conversation.*

*UML of Communication Sub System*

