**Requirement 3**

**Network Diagram**

C:\Users\40173513\Downloads\Player Diagram.png

No changes were made to the actual design of the system in our approach so this diagram is the most accurate one.

# Channel Interaction Sequence

C:\Users\Tomek\Documents\Parallel-Systems\Challenge\Interaction Diagram.png

**Interactions**

This implementation behaves in the same way as the original implementation but a turn system was successfully added. When the player receives GameDetails, they are also given the current turn value which is a simple integer. Player enters the valid point loop and then it checks if its ID matches the turn value, if it does then the player can pick cards, otherwise the player can only withdraw from the game.

Whenever a player picks 2 cards, player manager sends UpdateBoard to controller and controller updates the board for all players in the game.

Once the player picked 2 cards and they did not match, player manager sends a TurnManager data type over to the Controller which increments the turn to the next player and subsequently sends this new turn value to all players so they know whether it’s their turn.

**Datatypes**

GameDetails data type was changed to include a turn value that is sent to each player when they enter enrolled loop.

A new data type called TurnManager was added which contains the currentPlayer value that is pretty much the turn value. This data type has its own algorithm inside the ControlManager that decides which player moves next. After the next player, has been decided, TurnManager is sent back to all players to inform them of this change.

UpdateBoard type which takes in the chosen pairs and sends these to all players who then update the board using this data.