**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 much like the original implementation with the addition of a turn system. When the player receives GameDetails they are also given the current turn value which is stored as an integer. PlayerManager enters the valid point loop and then it checks if its ID matches the turn value, if it does then the player can pick two cards, otherwise the player is only able to withdraw from the game.

Whenever a player picks a card the PlayerManager sends UpdateBoard to controller and the controller updates the board for all players in the game.

When the player picks two cards which do not match, PlayerManager sends a TurnManager datatype 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 or not.

**Datatypes**

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

A new datatype, TurnManager, was added which contains the currentPlayer value that contains the turn value. This datatype has an algorithm in the ControlManager that dictates which player is able to make a move. After the next player has been decided the TurnManager is sent back to all players to inform them of this change.

The UpdateBoard type takes in chosen cards and then sends these to all players who then update their board using this data.

**Reasons for change in approach**

The main reason for moving away from the initial design of mobile agents was the time constraint. We overestimated the time we had to implement such an approach and as we worked on it we realised that it would be much easier on us to go with a much simpler approach of just expanding the original system by using various datatypes to run algorithms in the controller manager.

This approach is prone to deadlocks however, as there is just a single pair of channels between Controller and player. Furthermore, sending multiple things over a single channel isn’t that good of an idea. Still this was much easier to implement than mobile agents.