The first decision made was that the runnable interface was used instead of extending the thread class, to avoid multiple inheritance problem that extending the thread class provides.

We also decided that the min team size was 2 players and max is 3 to avoid overcrowding the console with clients.

As for the game, we decided that the number of incorrect attempts for each player is 3 attempts, so if the team has 3 players then their total attempts of the team would be 3\*3=9.

When implementing the teams formation, we let one player create a team by entering the team name and the team size, then we create an object of the Team class by calling the parametrized constructor and passing in the team name, the logged in player and the team size, then a team code is generated for that team that other players can use to join, then that player enters a waiting state using a while loop that waits until other players join the team to make the team full.

Meanwhile, other players can join a team by entering the team code generated as mention previously.

The linking of these players is done by the database class which is used to hold player entities and team entities by returning specified objects and storing them in a static array list. For example, when a player logs in, the database class is called to create a player object and store it in the players static array list.

Finally, for the scores criteria, we decided that if players get scores based on the number of correct guesses they made regardless of if they won or not. So if player1 in team1 guessed 2 letters correctly but eventually lost, then his score is saved in the history file as “player1 multi 2”, and if he plays another game and a history record already exists, then the new score is simply added to the end of the line as follows “player1 multi 2 3”.