**Project Summary**

Our team plans on implementing the well-known game of “Connect Four” by means of a web-based application developed using the C# programming language. The game will include a start menu that will allow user input for difficulty, game modes, and game piece color selection. The game will consist of an interactive gameboard that will allow the user to click to drop their game piece in the desired location.

**Game modes**

1. AI vs AI
2. Player vs AI
3. Player vs Player

**Difficulty**

1. Easy
2. Medium
3. Hard
4. Don’t Even Try

**Hardware & Software Specifications**

There are very few specific hardware and software requirements for this project. To ensure proper use and satisfaction, the following are needed:

1. A stable internet connection
2. A desktop, or non-mobile device capable of web-services

**Interface Design**

**Start Menu**

* Play Button
* Game-mode dropdown list
* Difficulty dropdown list
* Game piece color selection dropdown list

**Gameboard**

* 7 column x 6 row grid
* Grid will be drawn using graphics and brushes features
* Grid will be blue
* Player pieces will be yellow and red unless another color is specified
* A textbox will indicate player turn
* Another textbox will indicate turn count
* A dialog box will appear when a game ending state is met (WIN, TIE, and LOSS)
* Dialog box will allow access to main menu, rematch, or exit

**Program Design**

**Classes**

* Gameform
  + Responsible for graphics and other UI features
  + Responsible for running gameloop
* Gameboard
  + Responsible for gameboard behavior

**Methods**

Note: Methods are still subject to change as the project progresses.

* GameForm\_Paint()
  + Will implement graphics of the gameboard
  + Use of the Graphics class built-in to Visual Studio
* GameForm\_MouseClick()
  + Event-handler for a mouse-click will be defined here
  + Records mouse-click location which will be passed to playerMove() along with many others
  + Calls the playerMove() method
* playerMove()
  + Takes care of human player turn
  + Will call the DropPiece()
  + Will call the MovePainter() method
  + Will call the displayLastMove() method
* displayLastMove()
  + Responsible for keeping track of player turns
  + Will display output using a textbox
  + “Player 1’s last move was column X”
  + “Player 2’s last move was column X”
* DropPiece()
  + Ensures that the game piece falls to the lowest open row in the specified column
* MovePainter()
  + Checks turn
  + Draws the game piece in the desired location
  + Calls displayWinner() if a winner is found
* displayWinner()
  + Display winning player as a string in a message box
* WinnerPlayer()
  + Checks for four in a row vertical, horizontal, and both diagonals
* FullBoard()
  + Checks for tie condition
  + If turn count is == 42 the method will return true
* Win()
  + Implements scoring function that will be used in MiniMax
* MiniMax()
  + Returns score at terminal conditions (Full board, WIN, or when depth of search tree is 0)
  + Tests each game state possible until depth is 0
  + Calculates move that returns the best score for the AI
* turnSelector()
  + Simply alternates turns after each move
  + Code preview:
    - If (turn == 1)

{ turn = 2; }  
else

{ turn = 1; }

* AiMoveSelector()
  + Responsible for placing the piece for the AI