|  | Method | What needs to be done |
| --- | --- | --- |
| Ashanti | Question | * Prints game rules * Ask users names * Assign a symbol for each user |
| Mihlali | Board | * Design the board * Just basic for now * Make sure the board is mutable |
| Mihlali | Print board | * Prints the board with the new added symbols |
| Chris | Game | * Loop that allows players to move alternatively and continues until the game has ended * Asked user for their move * Calls all the other methods |
| Ashanti | Move Checker | * Checks if move player wants to make is valid * Checks if the first move of each player is one of the outside squares |
| Alisha | Placement | * Place Symbol in correct place on board |
| Mthobisi | Win Checker | * Check if someone won * Dont need to check for draw * End game |

public static void question()

{

board();

System.out.Println(print\_board);

game();

}

public static array board()

{

Row 1 = {“ “,” “, “ “, “ “};

Row 2 = {“ “,” “, “ “, “ “};

Row 3 = {“ “,” “, “ “, “ “};

Row 4 = {“ “,” “, “ “, “ “};

Return {row1. Row2, row3, row4)

Return an array of four string arrays.

}

public static xxx print\_board()

{

Nested loop (first for loop goes through the array of arrays and pulls out one row array at a time)

Loop (goes through each element in the called array and prints it with the board)

Print “----”

Proint “|”

Print each element of 1st array;

}

public static xxx game()

{

For int i=0; i< (possible number of moves+1);i++)

If i%2=0

Then it is x’s turn

Else

ITS O’S TURN

n = User moves

If (move\_checker(n)==true)

{

placement();

print\_board();

if(win\_check()==true);

{

Break;

}

}

}

public static boolean move\_check()

{

}

public static array placement()

{

}

public static boolean win\_check()

{

}