Game scenarios and how it reflects on the server

1- Login page

when the user press on the login button it will call the Xointerface constructor

```
public XOInterface(String typeOfOperation,Player player)
   {
      this.typeOfOperation = typeOfOperation;
      this.player = player;
   }
```

With **typeOfOperation="login"** And this interface will call the Player constructor

```
     public Player(String _userName,String _passwd)
   {
      userName=_userName;
      passwd=_passwd;
   }
```

And the server will reply by calling getOperationResult constructor

```
     public Boolean getOpearationResult()
   {
      return operationResult;
   }
```

and the operationResult var will be True or False
If True: it will show the next screen(Game mode selection).
if False: it will show (wrong user id or pass) message.

2- Sign up page

when the user press on the Sign up button it will call the Xointerface constructor

```
public XOInterface(String typeOfOperation,Player player)
    {
        this.typeOfOperation = typeOfOperation;
        this.player = player;
    }
```

With **typeOfOperation=”Register”** And this interface will call the Player constructor

```
public Player(String _userName,String _passwd,String _fName,String _lName)
    {
        userName=_userName;
        passwd=_passwd;
        fName=_fName;
        lName=_lName;
    }
```

And the server will reply by calling getOperationResult constructor

```
         public Boolean getOpearationResult()
    {
        return operationResult;
    }
```

and the operationResult var will be True or False
If True: it will show the next screen(login page).
if False: it will show (User name is already in use) message.

3- Game mode selection page

when the user press on the **(single mode)** button it will call the Xointerface constructor

```
public XOInterface(String typeOfOperation,Gamelog gamelog)
    {
        this.typeOfOperation = typeOfOperation;
        this.gamelog = gamelog;
    }
```

With t**ypeOfOperation=”playingSingleMode”** And this interface will call the Player constructor

```
         public Gamelog(String homePlayer, String opponentPlayer)
    {
        this.homePlayer = homePlayer;
        this.opponentPlayer = opponentPlayer;
    }
```

And the server will reply by calling getOperationResult constructor and Get Game log

```
         public Boolean getOpearationResult()
    {
        return operationResult;
    }
public Gamelog getGameLog()
    {
        return gamelog;
    }
```

and the operationResult var will be True or False
If True: it will show the next screen(Game in single mode).
if False: it will show (Server Error/can’t be reached) message.

4- Playing with the computer

when the game(single mode) finish it will call the Xointerface constructor

```
public XOInterface(String typeOfOperation,Player player,Gamelog gamlog)
    {
        this.typeOfOperation = typeOfOperation;
        this.player = player;
        this.gamelog = gamelog;
    }
```

With **typeOfOperation=”singleModeFinished”** And this interface will call the Player constructor

```
         public Player (String userName, boolean status,int score)
    {
        this.userName = userName;
        this.status = status;
        this.score = score;
    }
```

And the server will reply by calling getOperationResult constructor

```
         public Boolean getOpearationResult()
    {
        return operationResult;
    }
```

and the operationResult var will be True or False
If True: it will show (Status updated) message.
if False: it will show (Server Error/can’t be reached) message.

5- Playing in Multi mode

when the user press on (multi mode) button in (Game selection page) it will call the Xointerface constructor

```
public XOInterface(String typeOfOperation,Player player)
   {
      this.typeOfOperation = typeOfOperation;
      this.player = player;
   }
```

With typeOfOperation=”getPlayers” And this interface will call the Player constructor

```
       public Player(String _userName)
   {
      userName=_userName;
   }
```

And the server will reply with Players vector (**excluding my username)** with Player constructor

```
       public Player (String userName,String fName,String lName,boolean
                              status,int score,boolean isPlaying,int gameId)
   {
      this.userName = userName;
      this.fName = fName;
      this.lName = lName;
      this.status = status;
      this.score = score;
      this.isPlaying = isPlaying;
      this.gameId = gameId;
   }
```

and the reply will be the Players list

6- Selecting an opponent.

when the user press on any player username to invite them to play it will call the Xointerface constructor

```
public XOInterface(String typeOfOperation,Gamelog gamelog)
    {
        this.typeOfOperation = typeOfOperation;
        this.gamelog = gamelog;
    }
```

With **typeOfOperation="invite"** And this interface will call the Game.log constructor

```
         public Gamelog(String homePlayer, String opponentPlayer)
    {
        this.homePlayer = homePlayer;
        this.opponentPlayer = opponentPlayer;
    }
```

And the server will send the same message to client-2 (opponent player),

Then the server will reply client-1 with operrationResult message

the operationResult var will be True or False
If True: it will show (Invitation sent) message.
if False: it will show (Server Error/can't be reached) message.

7- Receiving an invitation.

when the server sends me an invitation

A popup message will be appeared which contains the username of the invitation sender, and has (Accept/Decline) options

If Accept: Multi mode game screen will be shown, and a meesage will be sent to the server **typeOfOperation=”accept”** with a constructor

```
public XOInterface(String typeOfOperation,Gamelog gamelog)
    {
        this.typeOfOperation = typeOfOperation;
        this.gamelog = gamelog;
    }
```

and Game.log constructor

```
         public Gamelog(String homePlayer, String opponentPlayer)
    {
        this.homePlayer = homePlayer;
        this.opponentPlayer = opponentPlayer;
    }
```

if Decline: the popup message will disappear. And it will send to the server **typeOfOperation=”decline”** with a constructor

```
public XOInterface(String typeOfOperation,Gamelog gamelog)
    {
        this.typeOfOperation = typeOfOperation;
        this.gamelog = gamelog;
    }
```

and Game.log constructor

```
         public Gamelog(String homePlayer, String opponentPlayer)
    {
        this.homePlayer = homePlayer;
        this.opponentPlayer = opponentPlayer;
    }
```

8- making a move within the game.

when the user make any move, it will send to the server
and call the Xointerface constructor

```
public XOInterface(String typeOfOperation,Gamelog gamelog,int
fieldNumber,char signPlayed)
    {
        this.typeOfOperation = typeOfOperation;
        this.gamelog = gamelog;
        this.fieldNumber=fieldNumber;
        this.signPlayed=signPlayed;
    }
```

With **typeOfOperation="playMove"** And this interface will call the Game.log constructor

```
        public Gamelog(int gameId ,String _homePlayer,String _opponentPlayer)
{
        this.gameId = gameId;
        homePlayer=_homePlayer;
        opponentPlayer=_opponentPlayer;
    }
```

And the server will send the same message to client-2 (opponent player),

The server will reply client-1 (home user) by operationResult

If True: it will show any indication that the opponent player received your move.
if False: it will show (Server Error/can't be reached) message.

9- Receiving a move within the game.

when the other user make any move, the server will send its move to you

Then you should send to the server a confirmation message by using this constructor:

```
public XOInterface(String typeOfOperation,Boolean moveReceived)
    {
       this.typeOfOperation = typeOfOperation;
       this.moveReceived=moveReceived;
    }
```

The server will reply by operationResult

If True: it will show any indication that you can do your move
if False: it will show (Server Error/can’t be reached) message.

10- Finishing the game.

After finishing the game, The winner will send to the server using this interface

```
public XOInterface(String typeOfOperation,Player player,Gamelog gamlog)
    {
       this.typeOfOperation = typeOfOperation;
       this.player = player;
       this.gamelog = gamelog;
    }
```

With **typeOfOperation=”multiModeFinished”** And this interface will call the player and GameLog constructors

```
public Player (String userName,int score)
    {
       this.userName = userName;
       this.score = score;
    };

public Gamelog (int gameId)
    {
       this.gameId = gameId;
    }
```

The server will reply by operationResult

If True: it will show any indication that you are the winner
if False: it will show (Server Error/can’t be reached) message.