xGame Chess

Player

- Color

- forfeitGame()
- takeTurn()

Game

- startTime:Time
- endTime:Time
- creator:User
- opponent:User
- winner:User = null
- turn:Player
- players: Player[]
- moveHistory:ArrayList<MovePiece>

- createMove():MovePiece
- invitePlayer(User)
- declareWinner():User

User

- email:String
- password:String
- activeGames:Game[]
- finishedGames:Game[]

+ getEmail():String
+ processUsernamePassword(JSON Obj):Boolean
+ getGame(int Gameid):Game
+ registerUser():Boolean
+ authenticateUser():Boolean
+ deleteUser()
+ getUserData()
- startNewGame(int Gameid)

Board

- board:Square[][]

+ getState():Square[][]
+ getSquare(Position):Square
+ toString():String
- initalize()

MovePiece

- board:Board
- player:Player
- isValidMove:Boolean
- startSquare:Square
- endSquare:Square
- currentPiece:boardPiece
- capturedPiece:boardPiece

+ getStartSquare():Square
+ getEndSquare():Square
+ getIsValidMove():Boolean
+ executeMove()
+ promotePawn()
+ castle()
+ undoMove()
- isValidMove():Boolean

Square

- position
- piece:BoardPiece

+ getPiece():BoardPiece
+ setPiece(Piece)
+ getPosition():Position

<<BoardPiece>>

- position
- pieceType
- color

+ getPosition():Position
+ setPosition(Position)
+ getColor():Color
+ setColor(Color)
+ getName():PieceType
+ compareTo(PieceType)
+ getOpposite():Position
+ getPossibleMoves():ArrayList<int[]>
+ getPossibleMovePositions(ArrayList<int[]>):Position[]

Pawn

Rook

Knight

Bishop

Queen

King

1
2
1..*
0..*
1
1
0..*
1
64
2
1..*
1
1
1