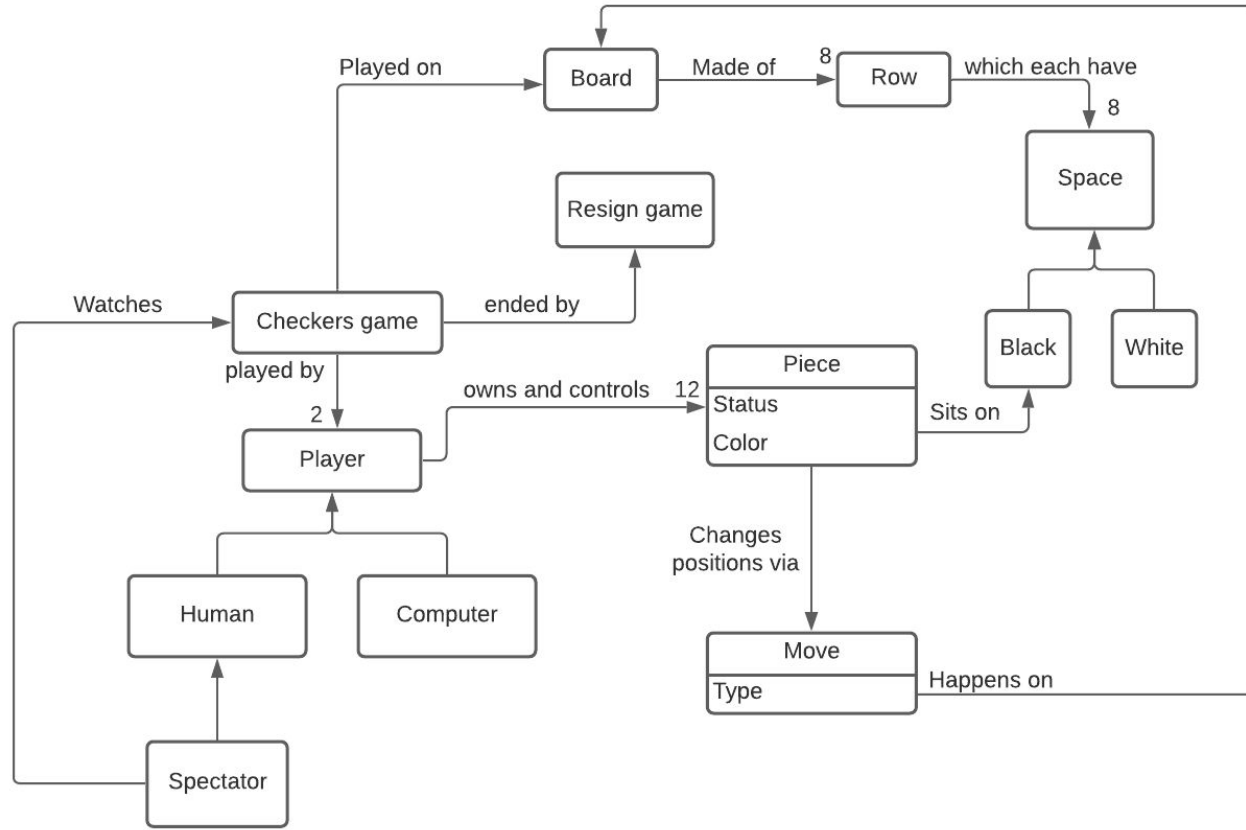# *Sprint 4: Transition Phase*
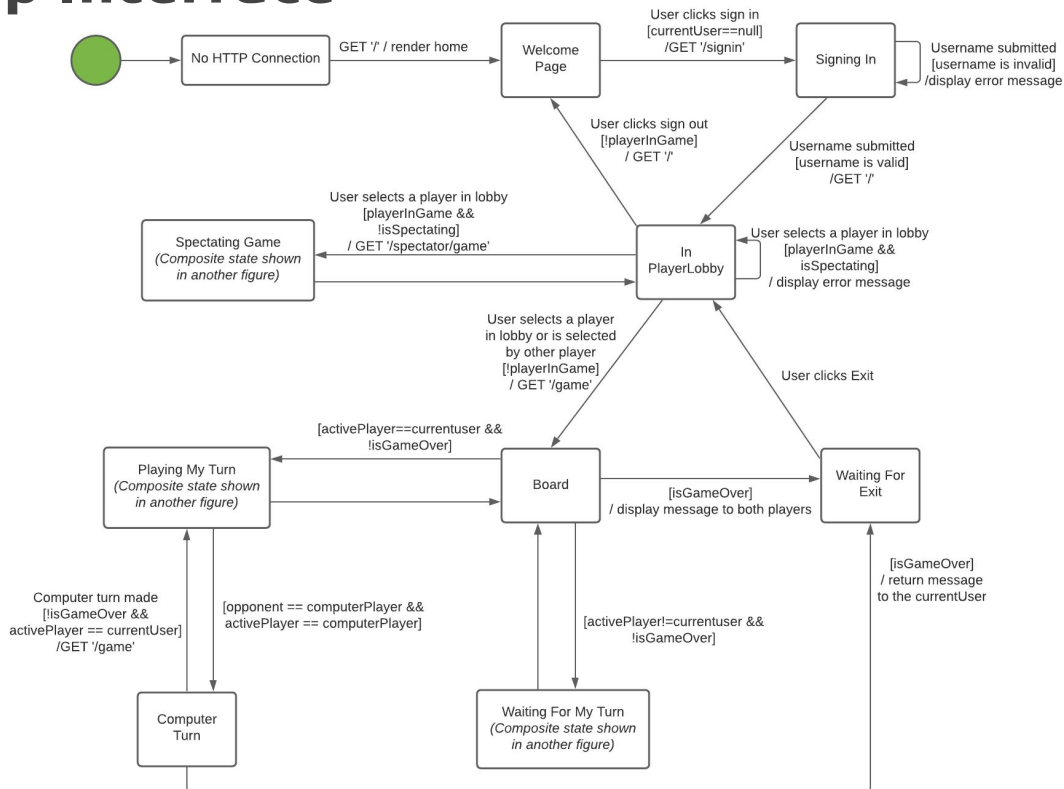
Team 2C: Alanna L, Alex I, Benson Y, Julio C, Joshua S

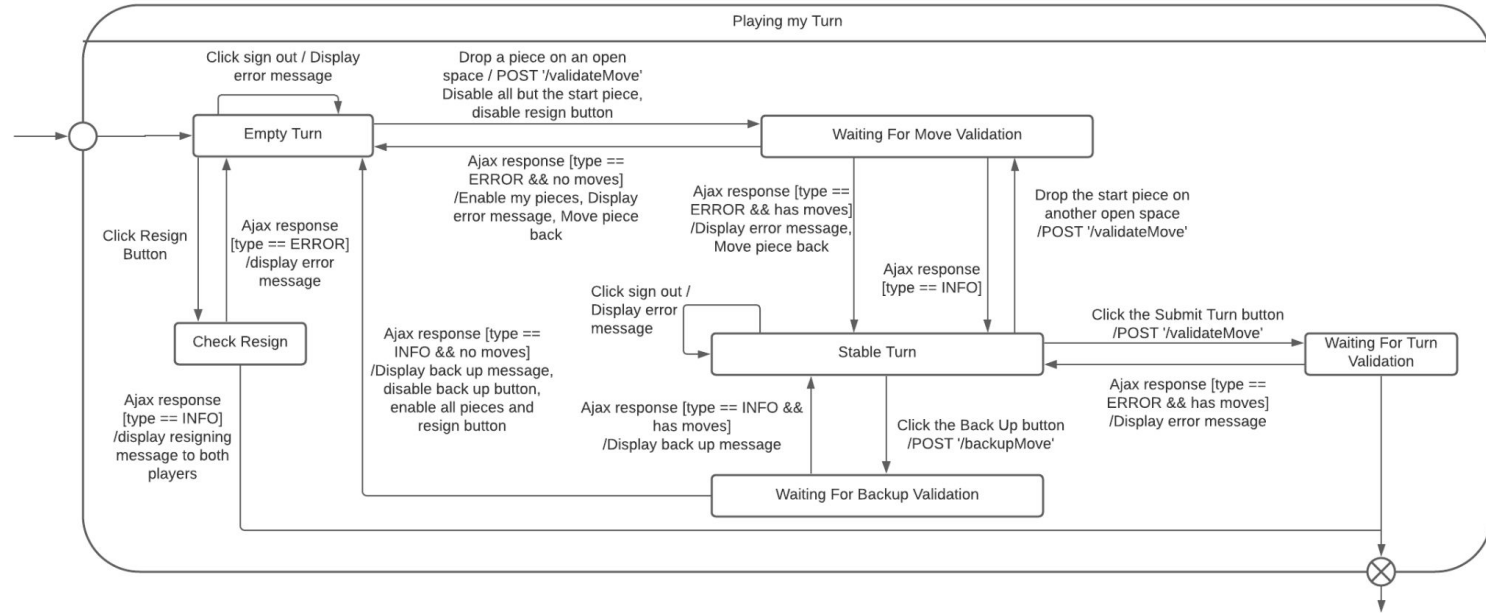# Domain Model

Project Design

Web App Interface

```
( ● ) ──→ No HTTP Connection ──GET '/' / render home──→ Welcome Page
```

No HTTP Connection ── GET '/' / render home ──→ **Welcome Page**

Welcome Page ── User clicks sign in [currentUser==null] /GET '/signin' ──→ **Signing In**

Signing In ── Username submitted [username is invalid] /display error message (self loop)

Signing In ── Username submitted [username is valid] /GET '/' ──→ **In PlayerLobby**

In PlayerLobby ── User clicks sign out [!playerInGame] / GET '/' ──→ **Welcome Page**

In PlayerLobby ── User selects a player in lobby [playerInGame && !isSpectating] / GET '/spectator/game' ──→ **Spectating Game** *(Composite state shown in another figure)*

In PlayerLobby ── User selects a player in lobby [playerInGame && isSpectating] / display error message (self loop)

In PlayerLobby ── User selects a player in lobby or is selected by other player [!playerInGame] / GET '/game' ──→ **Board**

Playing My Turn *(Composite state shown in another figure)*

Board ── [activePlayer==currentuser && !isGameOver] ──→ **Playing My Turn**

Board ── [isGameOver] / display message to both players ──→ **Waiting For Exit**

Waiting For Exit ── User clicks Exit ──→ **In PlayerLobby**

Playing My Turn ── Computer turn made [!isGameOver && activePlayer == currentUser] /GET '/game' ──→ (from) **Computer Turn**

Computer Turn ── [opponent == computerPlayer && activePlayer == computerPlayer] ──→ **Playing My Turn**

Board ── [activePlayer!=currentuser && !isGameOver] ──→ **Waiting For My Turn** *(Composite state shown in another figure)*

Waiting For My Turn ── [isGameOver] / return message to the currentUser ──→ **Waiting For Exit**

Web App Interface: Playing Turn

Playing my Turn

Click sign out / Display error message

Empty Turn

Drop a piece on an open space / POST '/validateMove' Disable all but the start piece, disable resign button

Waiting For Move Validation

Ajax response [type == ERROR && no moves] /Enable my pieces, Display error message, Move piece back

Ajax response [type == ERROR && has moves] /Display error message, Move piece back

Drop the start piece on another open space /POST '/validateMove'

Click Resign Button

Ajax response [type == ERROR] /display error message

Ajax response [type == INFO]

Check Resign

Ajax response [type == INFO && no moves] /Display back up message, disable back up button, enable all pieces and resign button

Click sign out / Display error message

Stable Turn

Click the Submit Turn button /POST '/validateMove'

Waiting For Turn Validation

Ajax response [type == ERROR && has moves] /Display error message

Ajax response [type == INFO] /display resigning message to both players

Ajax response [type == INFO && has moves] /Display back up message

Click the Back Up button /POST '/backupMove'

Waiting For Backup Validation

Web App Interface: Waiting Turn

Waiting for my Turn

Click sign out /
Display error
message

Waiting For 5
Second Delay

5 Second timeout
/POST '/checkTurn'

Checking For
My Turn

Ajax response [type ==
INFO && text == "false"]

Ajax response [type ==
INFO && text == "true"]
/GET '/game'

Click Resign
Button

Ajax response
[type == ERROR]
/display error message

Check
Resign

Ajax response [type == INFO]
/ display resigning message to both players

Web App Interface: Spectating Game

UML Diagrams

Application Tier

- This tier handles the behavior of the web application once the user interacts with it through the UI.
- Main components are the GameCenter and the PlayerLobby.
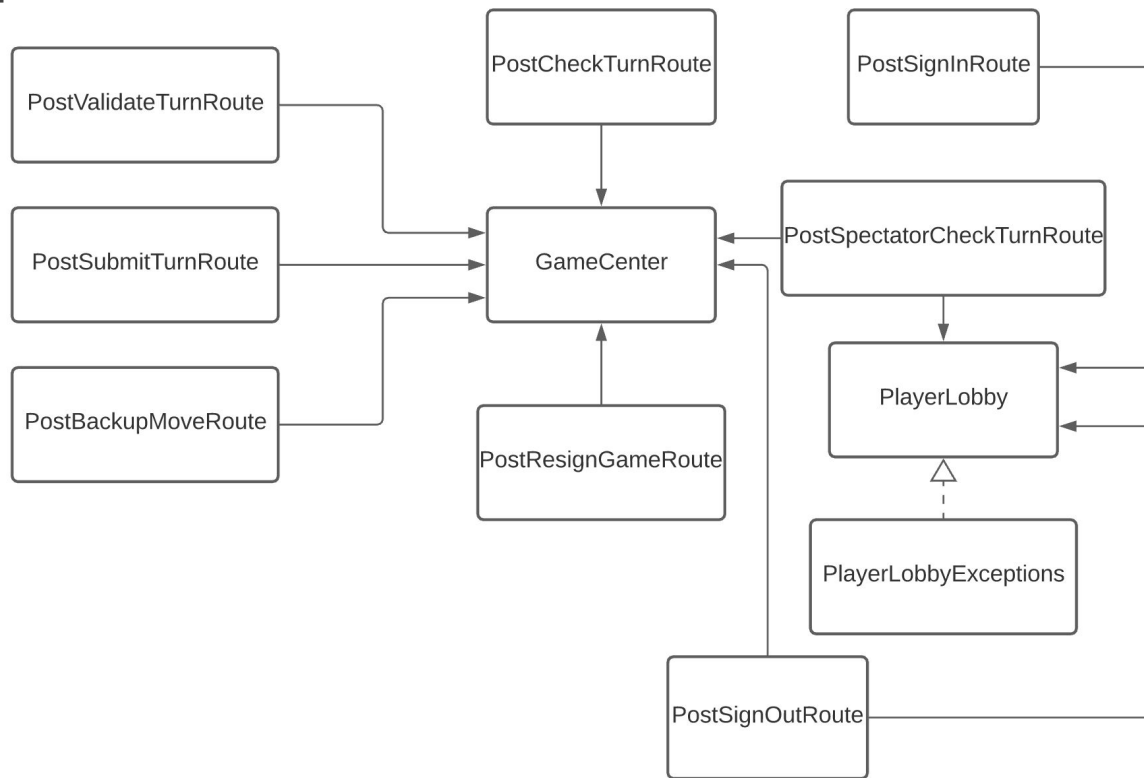
User Interface Tier

- The User Interface is what is presented to the user once the application is running.

- The main components are the GetHomeRoute and the GetGameRoute.
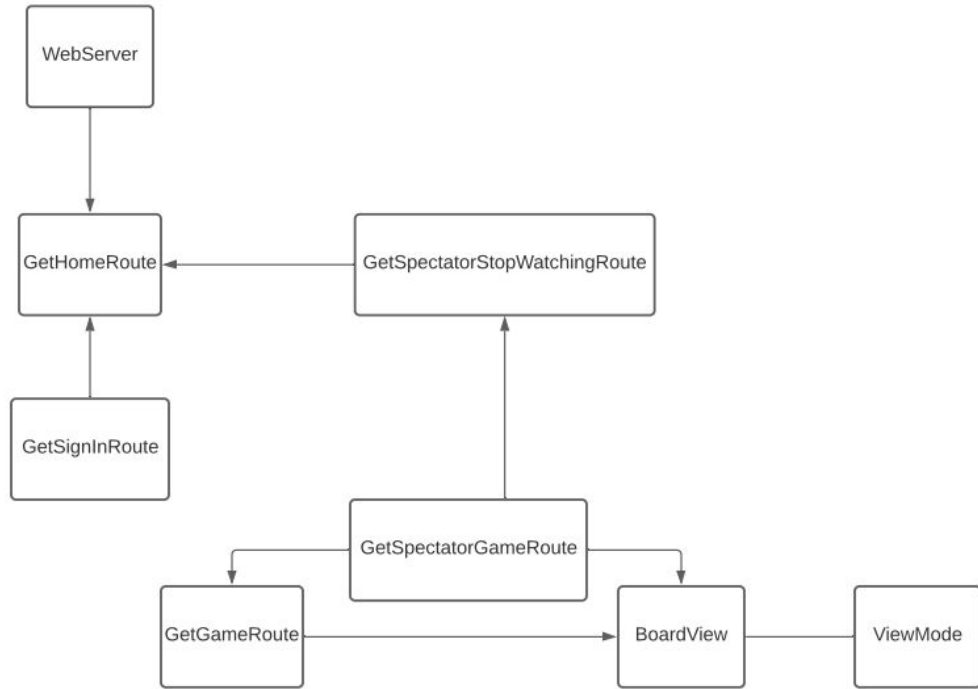
Model Tier

- The model tier is in charge of handling the behavior of the game and how the rules are implemented.
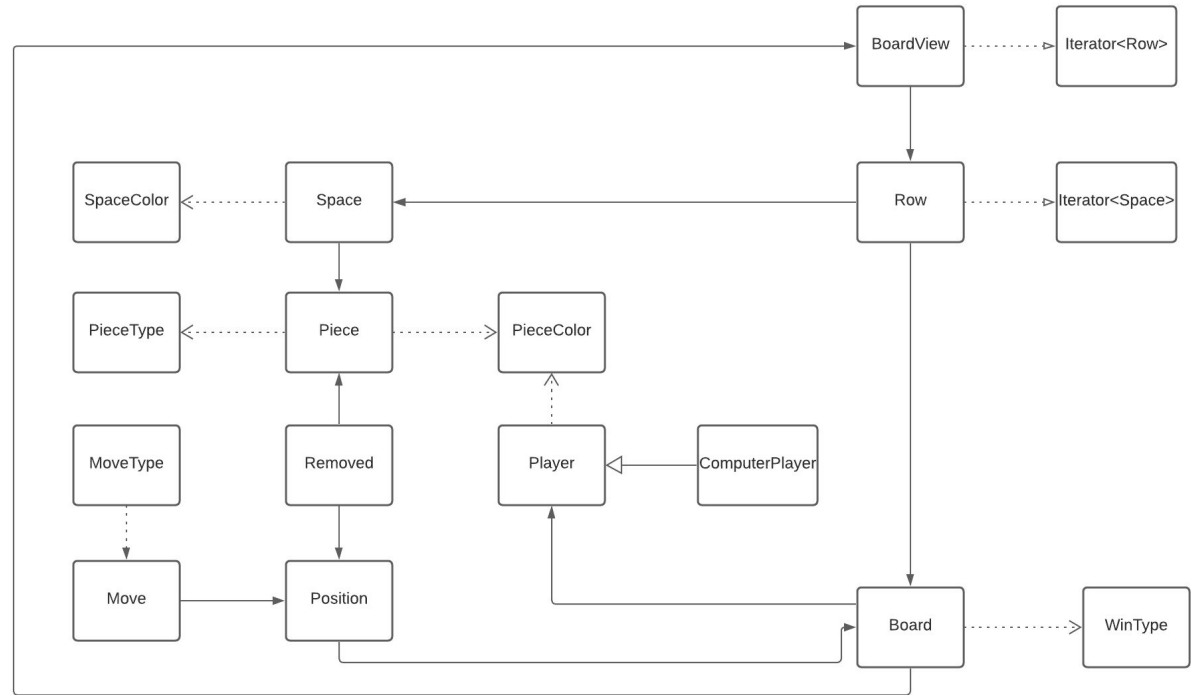
Code metrics

Major Changes: Lines of Code & Complexity Metrics

Before

com.webcheckers	122	5,284	122	2,353	0
	26	8,739	26	5,808	0
public.js	5	2,605	5	2,605	0
com		5,284		2,353	
public		3,242		3,242	
public.js.game.modes		1,544		1,544	
spark		187		187	
spark.template		187		187	
Total	**8,739**		**5,808**		**2,931**
Average	546.19		363.00		183.19

After

public.img	375	375	375	375	0
public.js.game.modes.replay	335	335	335	335	0
public.js.game.modes.spectator	284	284	284	284	0
public.css	262	262	262	262	0
public.js.game.model	199	199	199	199	0
spark		183		183	
spark.template		183		183	
spark.template.freemarker	183	183	183	183	
Total	**12,965**		**7,222**		**5,743**
Average	810.31		451.38		358.94

com.webcheckers.util.Message.Type		0
com.webcheckers.util.Validate	3.73	41
com.webcheckers.util.ValidateTest	1.00	32
Total		**435**
Average	1.39	8.37

com.webcheckers.util.MessageTest	1.00	4
com.webcheckers.util.Validate	4.32	82
com.webcheckers.util.ValidateTest	1.03	61
Total		**790**
Average	1.51	11.62

Major Changes: Chidamber-Kemerer Metrics

Before

com.webcheckers.ui.PostSubmitTurnRouteTest	14	1	1	0	42	9
com.webcheckers.ui.PostValidateMoveRoute	8	1	1	0	23	13
com.webcheckers.ui.PostValidateMoveRouteTest	11	1	1	0	32	7
com.webcheckers.ui.TemplateEngineTester	5	1	1	0	14	7
com.webcheckers.ui.ViewMode	4	0			0	0
com.webcheckers.ui.WebServer	15	1	1	0	17	2
com.webcheckers.ui.WebServerTest	3	1	2	0	10	3
com.webcheckers.util.Message	11	1	3	0	11	7
com.webcheckers.util.Message.Type	4	0			0	0
com.webcheckers.util.Validate	12	1	1	0	33	41
com.webcheckers.util.ValidateTest	10	1	1	0	67	32
Total						**435**
Average	7.85	1.07	1.83	0.00	18.58	8.37

After

com.webcheckers.ui.PostSignOutRouteTest	5	1	1	0	20	4
com.webcheckers.ui.TemplateEngineTester	5	1	1	0	14	7
com.webcheckers.ui.ViewMode	4	0			0	0
com.webcheckers.ui.WebServer	21	1	1	0	22	2
com.webcheckers.ui.WebServerTest	3	1	1	0	9	2
com.webcheckers.util.Message	21	1	3	0	15	11
com.webcheckers.util.Message.Type	8	0			0	0
com.webcheckers.util.MessageTest	2	1	4	0	17	4
com.webcheckers.util.Validate	12	1	1	0	44	82
com.webcheckers.util.ValidateTest	12	1	2	0	108	61
Total						**790**
Average	9.32	1.07	1.94	0.02	23.07	11.62

Design Improvements
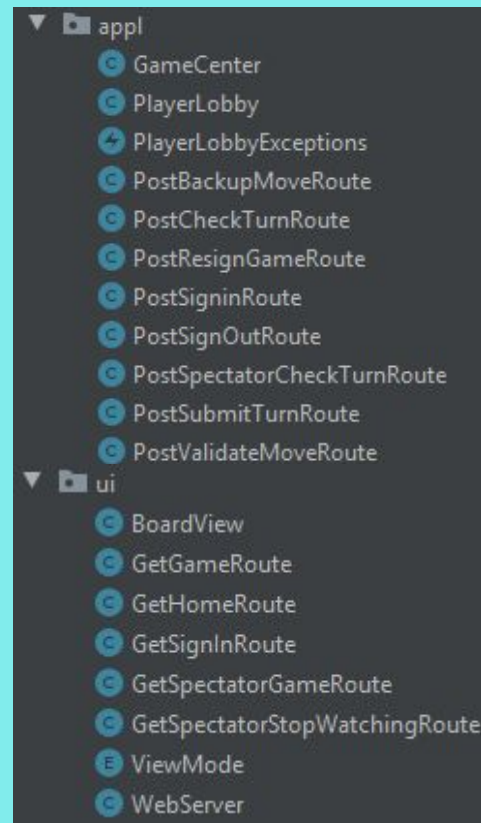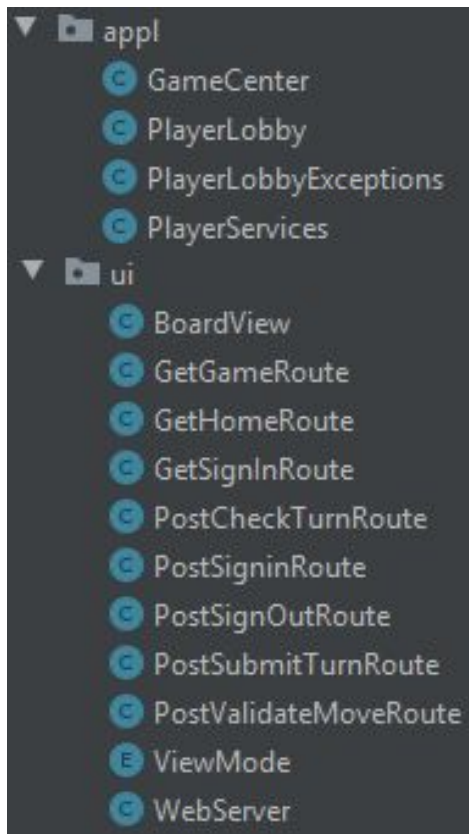
Improvements We Made

Polymorphism

Reduced Cross-tier Dependencies

Tier Restructuring

Reduced Cross-tier Dependencies and Restructured

Recommendations for Improvement

What We Could Work On

Spectator Class	Validate Classes	Javadoc Content
Highlight Previous Move	Home Page	Computer Difficulty

What We Could Work On

Spectator Class

Validate Classes

Javadoc Content

Highlight Previous Move

Home Page

Computer Difficulty

What We Could Work On

Spectator Class	Validate Classes	Javadoc Content
Highlight Previous Move	Home Page	Computer Difficulty

What We Could Work On

Spectator Class	Validate Classes	Javadoc Content
Highlight Previous Move	Home Page	Computer Difficulty

What We Could Work On

Spectator Class

Validate Classes

Javadoc Content

Highlight Previous Move

Home Page

Computer Difficulty

What We Could Work On

Spectator Class	Validate Classes	Javadoc Content
Highlight Previous Move	Home Page	Computer Difficulty

Thanks!

Does anyone have any questions?

Call 248 - 434 - 5508

Our representatives are ready to answer any questions