

The Jungle Game

Cheshire Coders

- Angélica Fallas
- Taner King
- Adam Gundem
- Alexander Hennings
- Cameron Ackerman

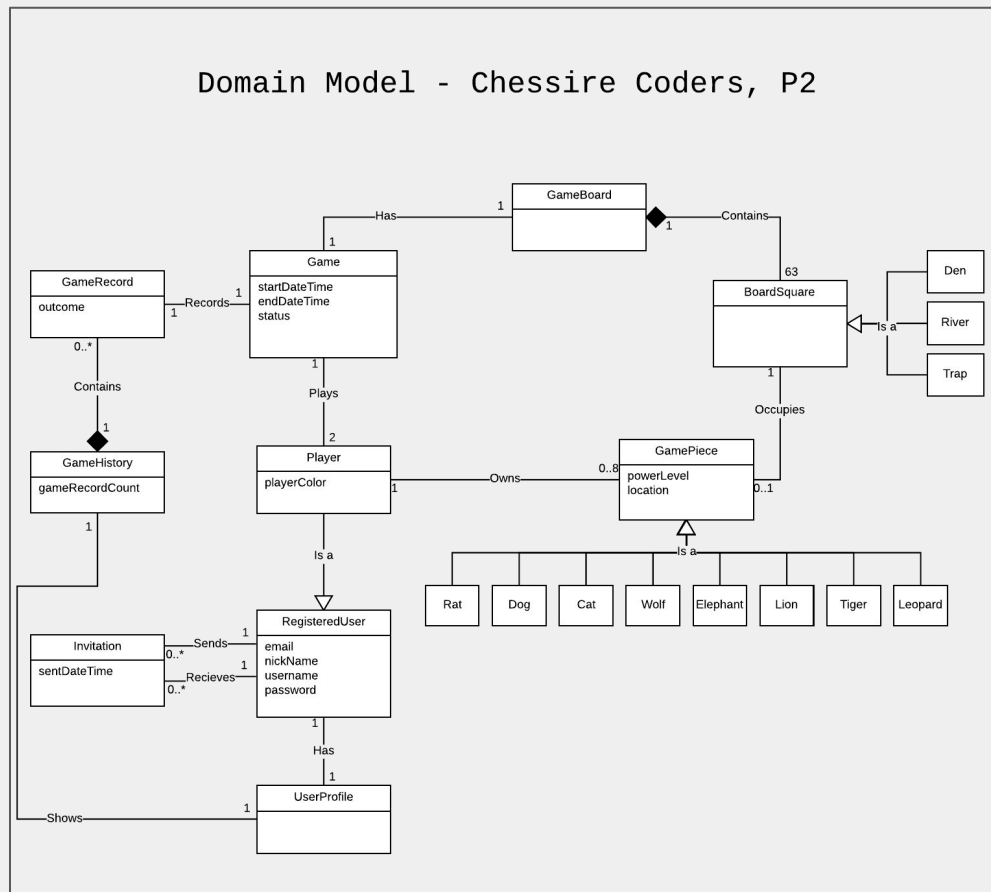


Presentation Overview

- Changes from last iteration
- Sequence diagrams
- Class diagrams
- Test Case Summary
- Project Tools
- Traceability Matrix & Use Case Progress
- Demo

Domain Model Changes

Removed *piece* from **BoardSquare** and
players from **Game**.



Glossary Changes

- Added entries for attributes.
- Sorted items alphabetically.

Glossary

BoardSquare: A representation of a single square on the Jungle board. A square has an attribute piece.

Game: An instance of a game of Jungle.

-**endDateTime:** Date and time when a game ended.

-**startDateTime:** Date and time when a game started.

-**status:** Status of a specific game (ongoing, completed, abandoned, etc)

GameBoard: A representation of the Jungle board that contains the current state of a game. The game board contains the different squares of Jungle, and any uncaptured Jungle pieces.

GameHistory: The game history is shown on each registered user's profiles. It includes a brief synopsis of each game played by that user.

-**gameRecordCount:** Represents the average score for a certain player.

GamePiece: A representation of a single Jungle piece. It is required that a game piece must be one of its eight different specialization types (i.e. if GamePiece were a Java class, it would be abstract). And there may be no more than one of each piece type per player.

-**location:** represents where a GamePiece is located withing a GameBoard.

-**powerLevel:**Represents the current level that a certain GamePiece has in a given state of the game.

GameRecord: A game record is the outcome of a single game of jungle.

-**outcome:** represents the final result of a certain GameRecord.

Invitation: An invitation is a request for another registered user to play a game with the sending user. Each invitation has one sender and one receiver.

-**sentDateTime:** specific date time value that represent when an invitation was sent.

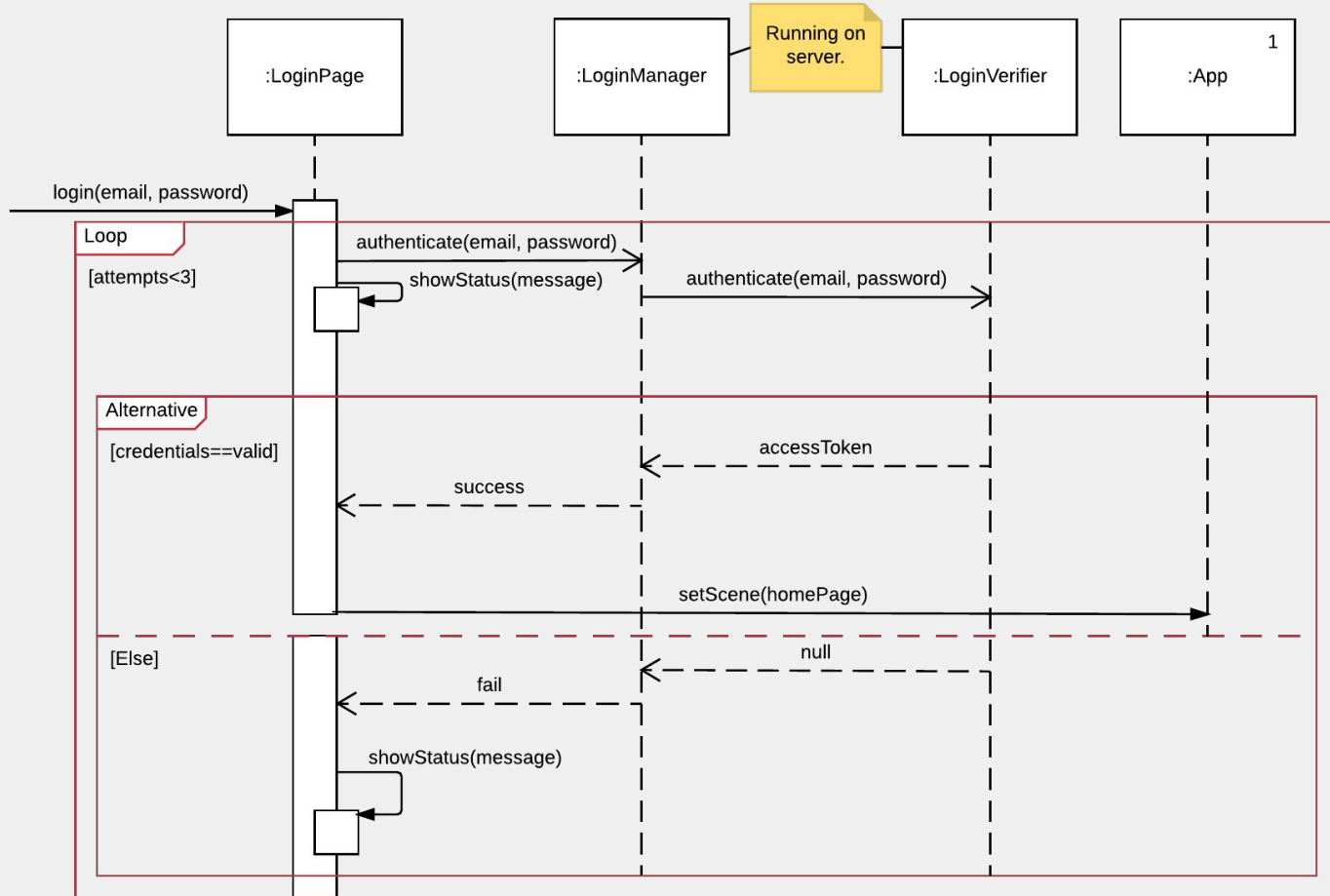
Player: An extension of a registered user. They may make moves, capture pieces, and perform other actions that the registered user entity cannot. Each player owns 0-8 game pieces(depending on how many have been captured by an opposing player) that they may control.

-**playerColor:** indicates what team the player is on.

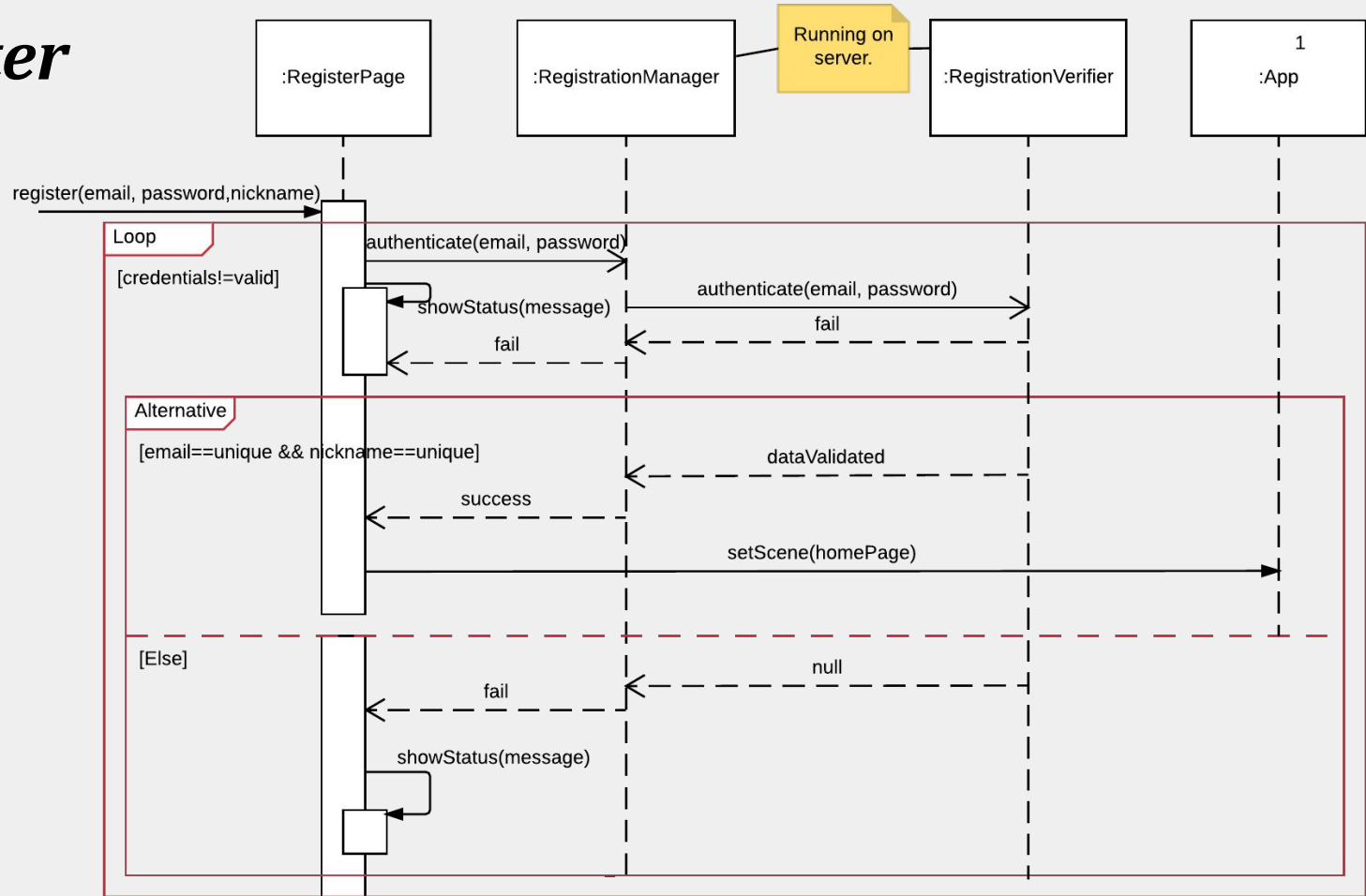
Sequence diagrams

- Login
- Game Invite
- Game Create
- Register

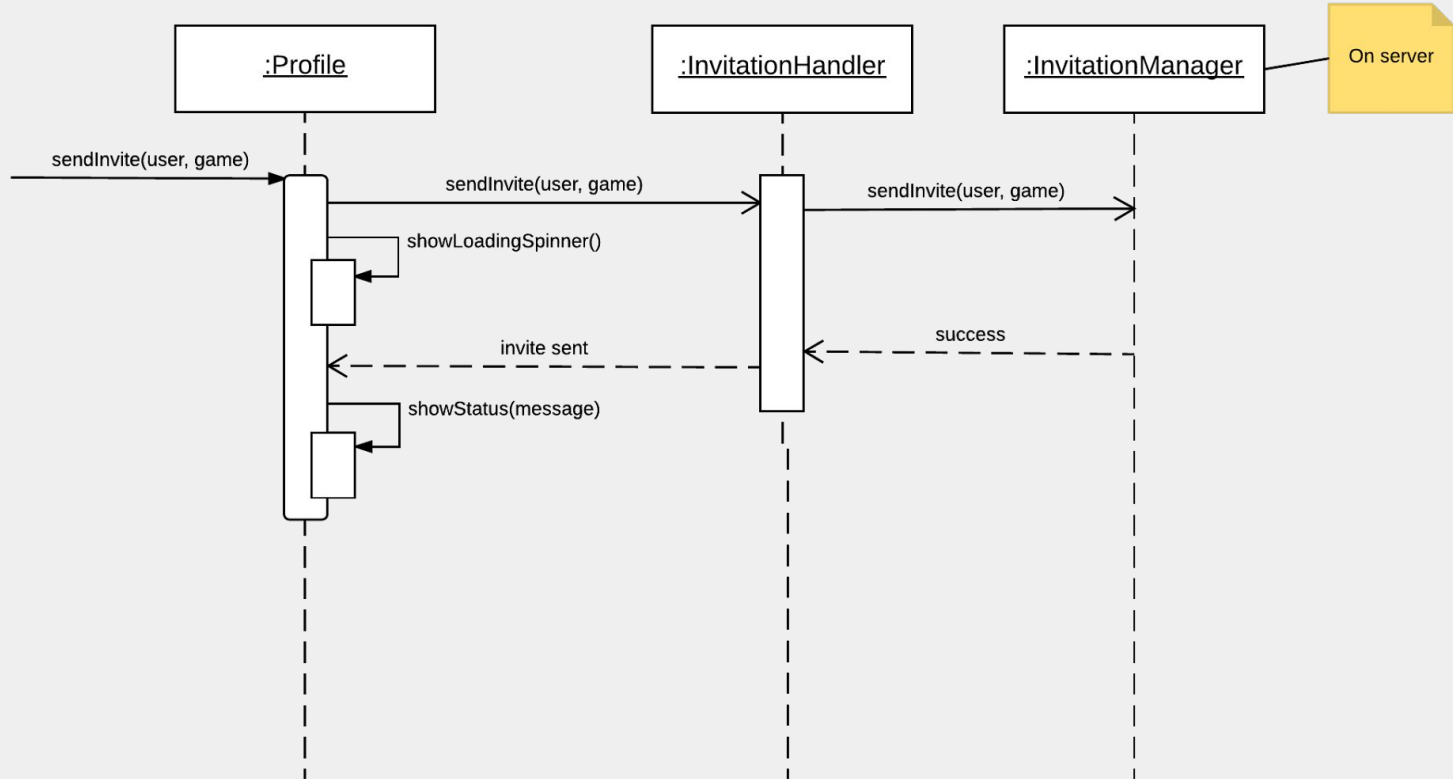
Login



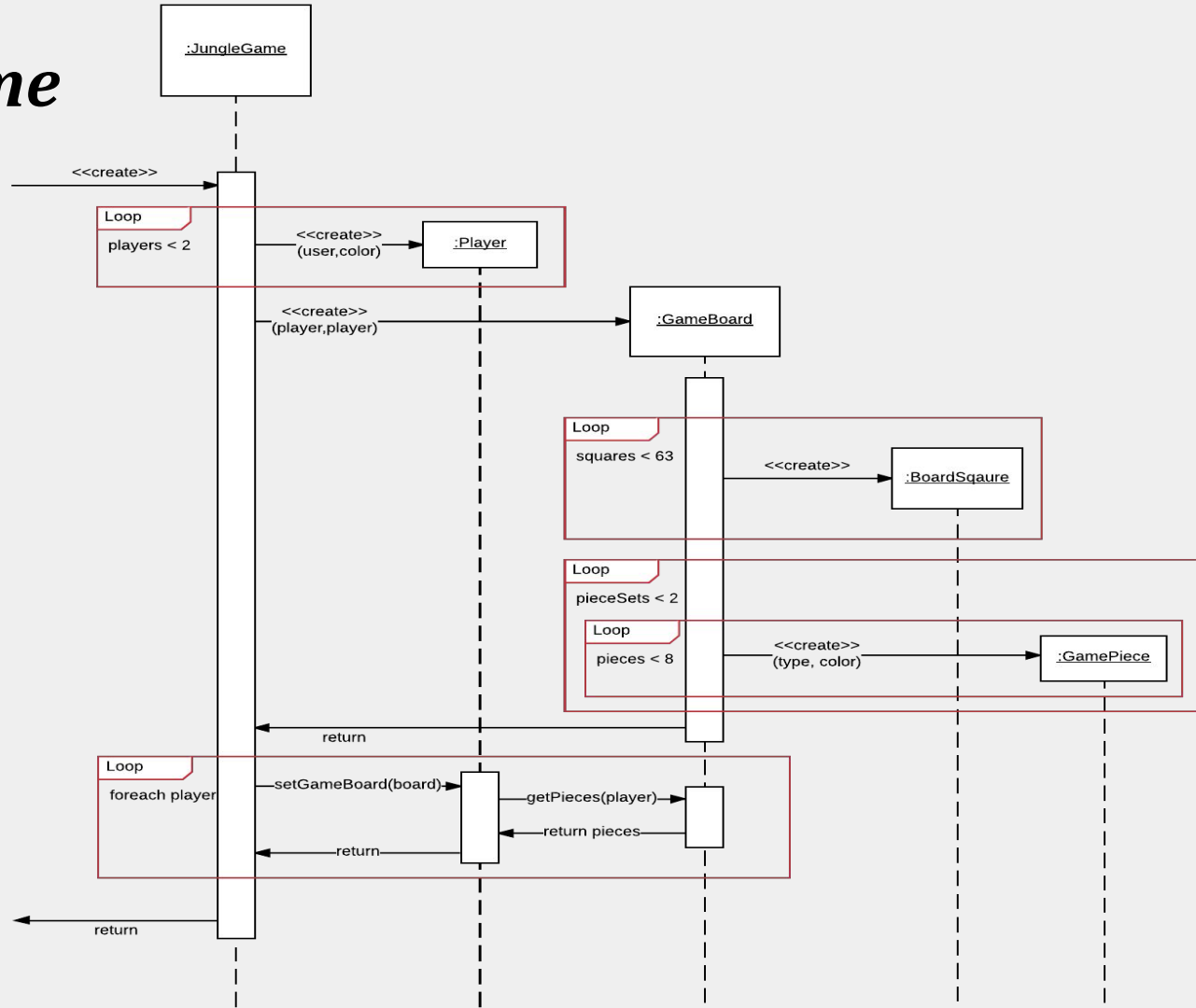
Register



Game Invite

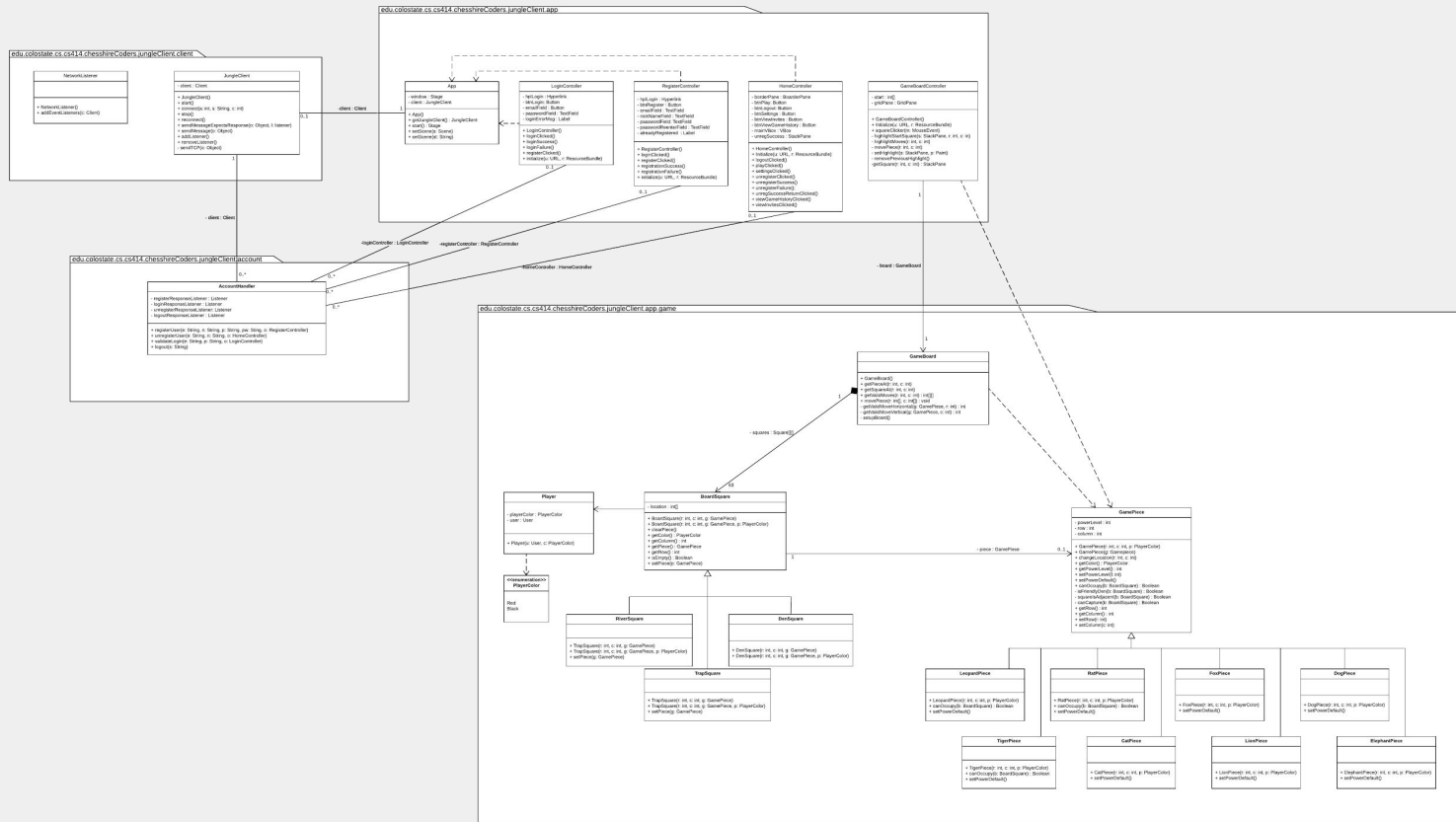


Create Game



Class Diagram

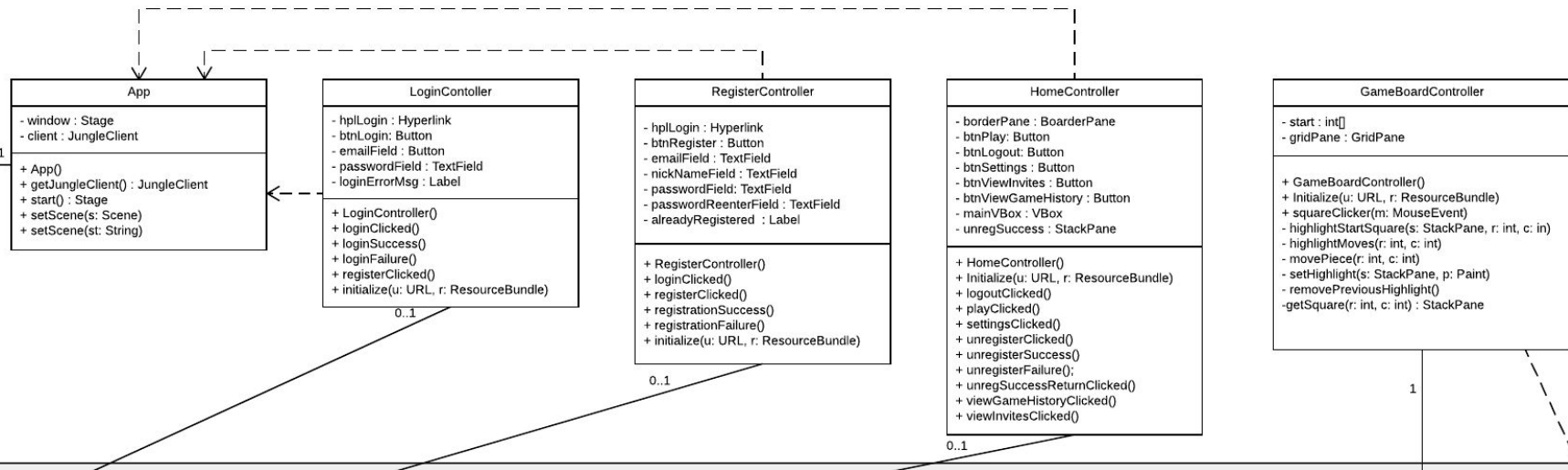
Client Logic - Overview



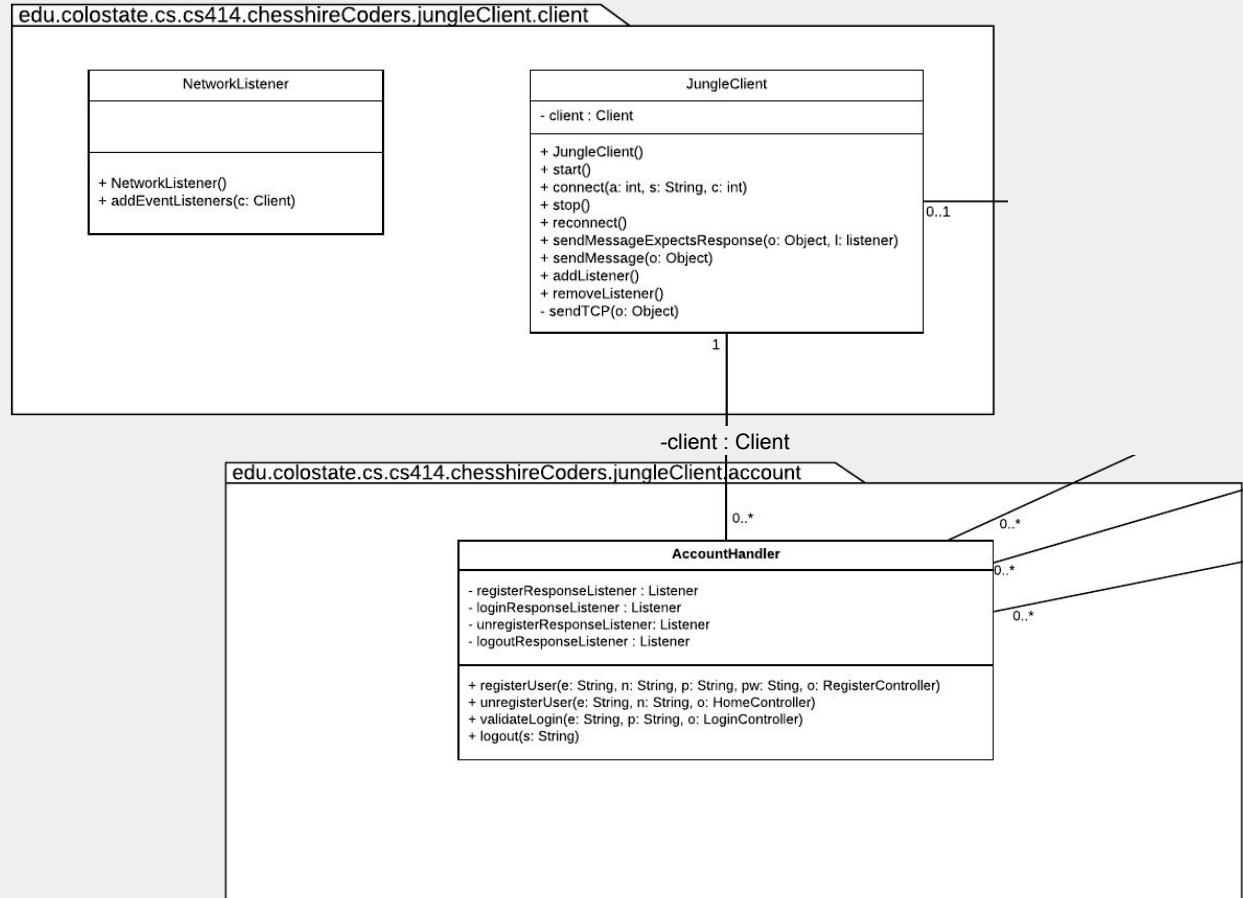
Class Diagram

Client Logic

edu.colostate.cs.cs414.cheshireCoders.jungleClient.app



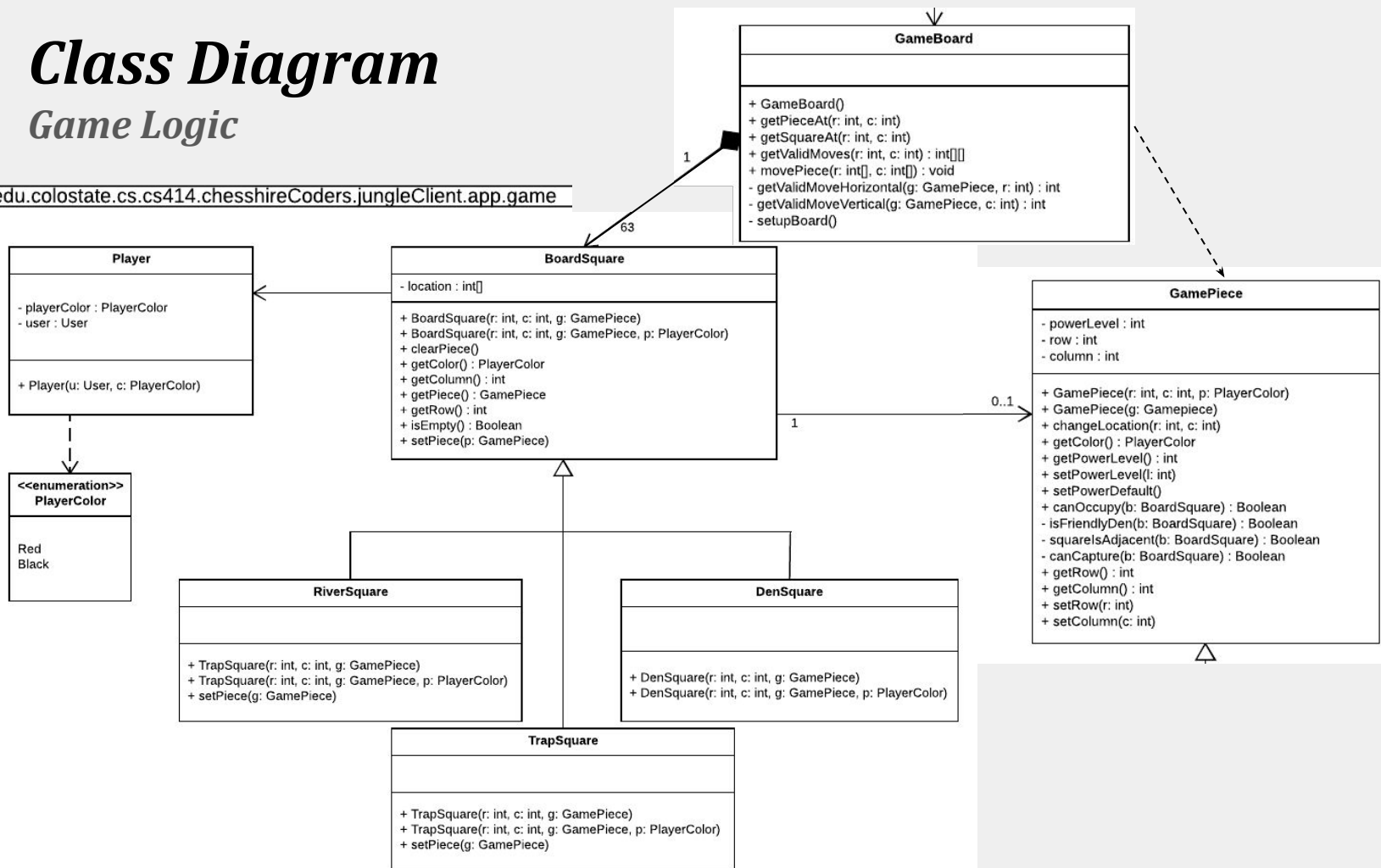
Class Diagrams - Client cont.



Class Diagram

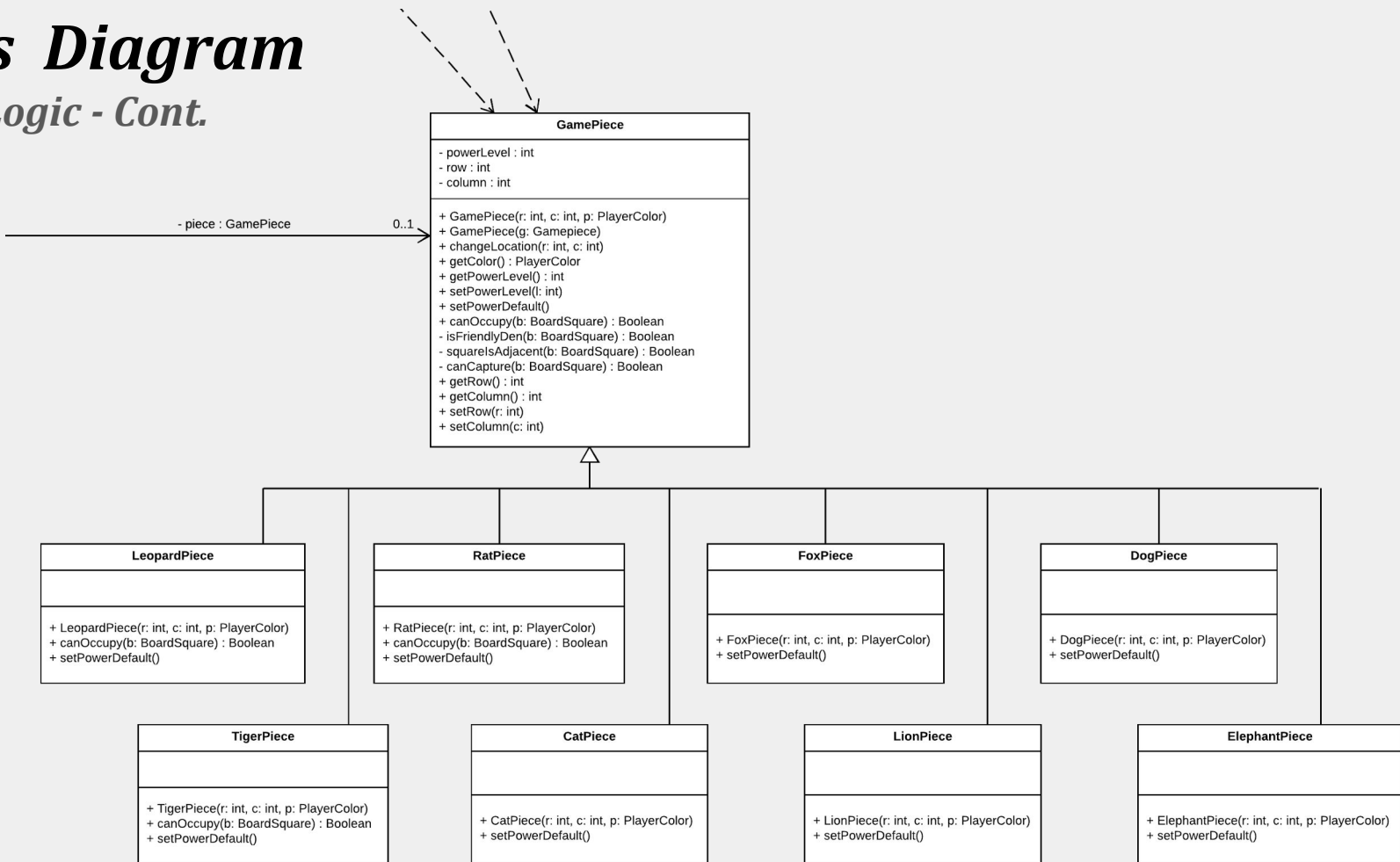
Game Logic

edu.colostate.cs.cs414.cheshireCoders.jungleClient.app.game

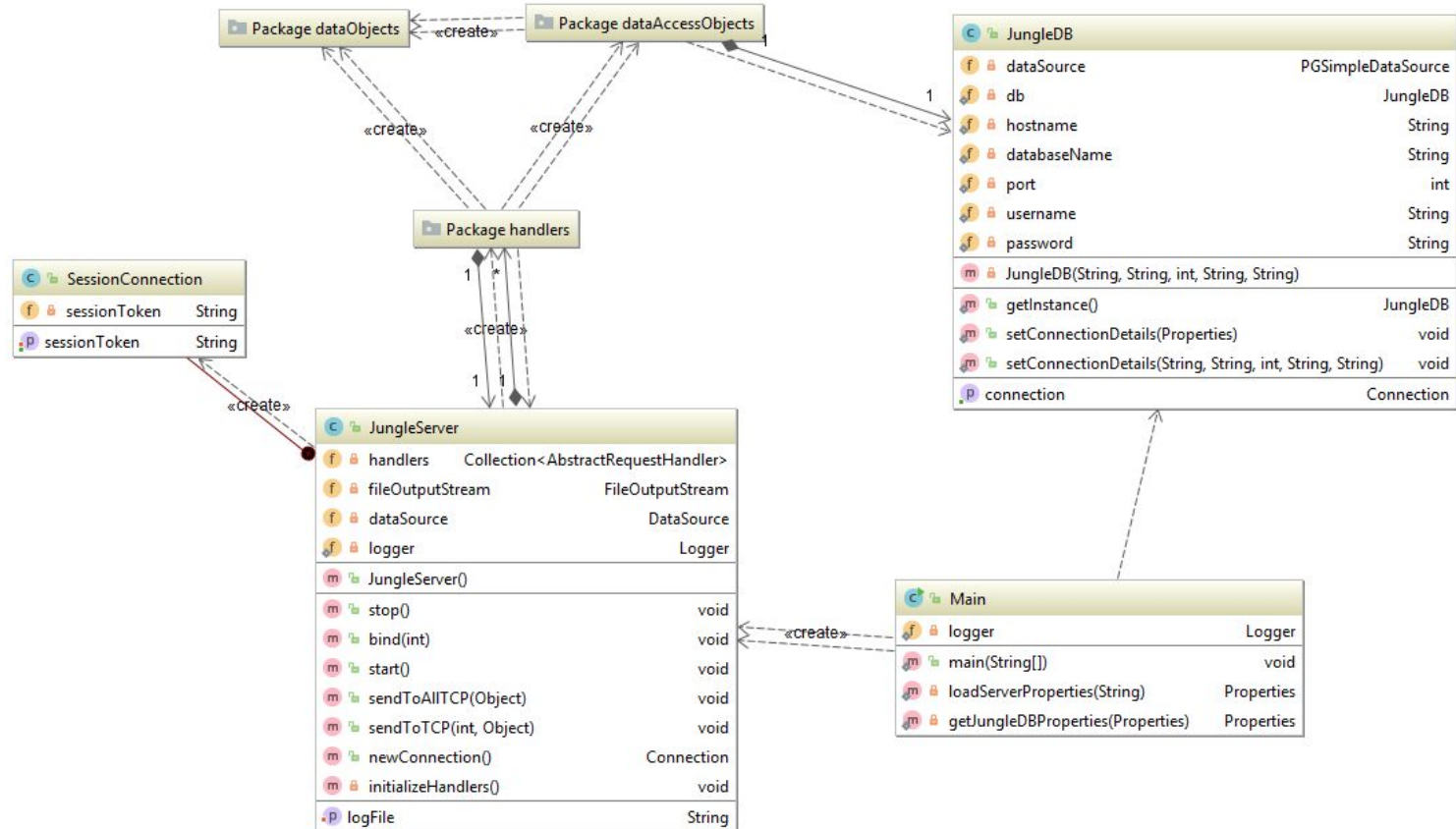


Class Diagram

Game Logic - Cont.

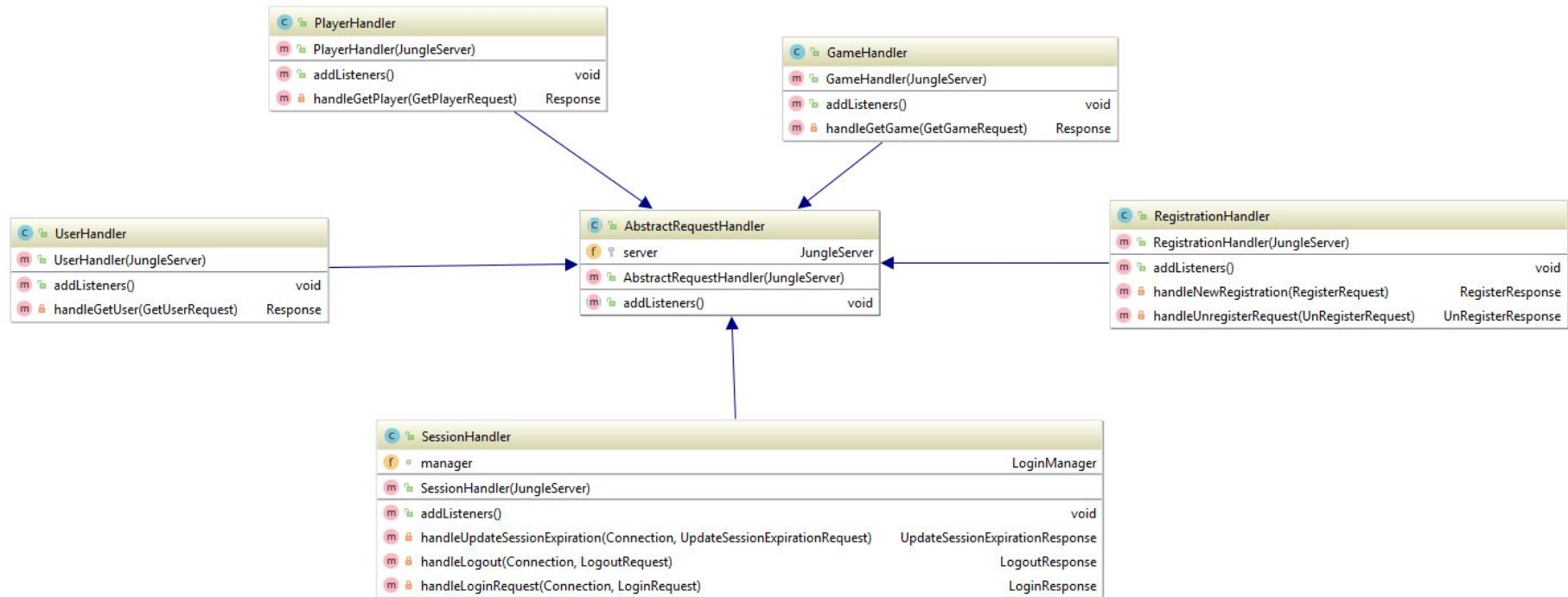


Class Diagrams - Server



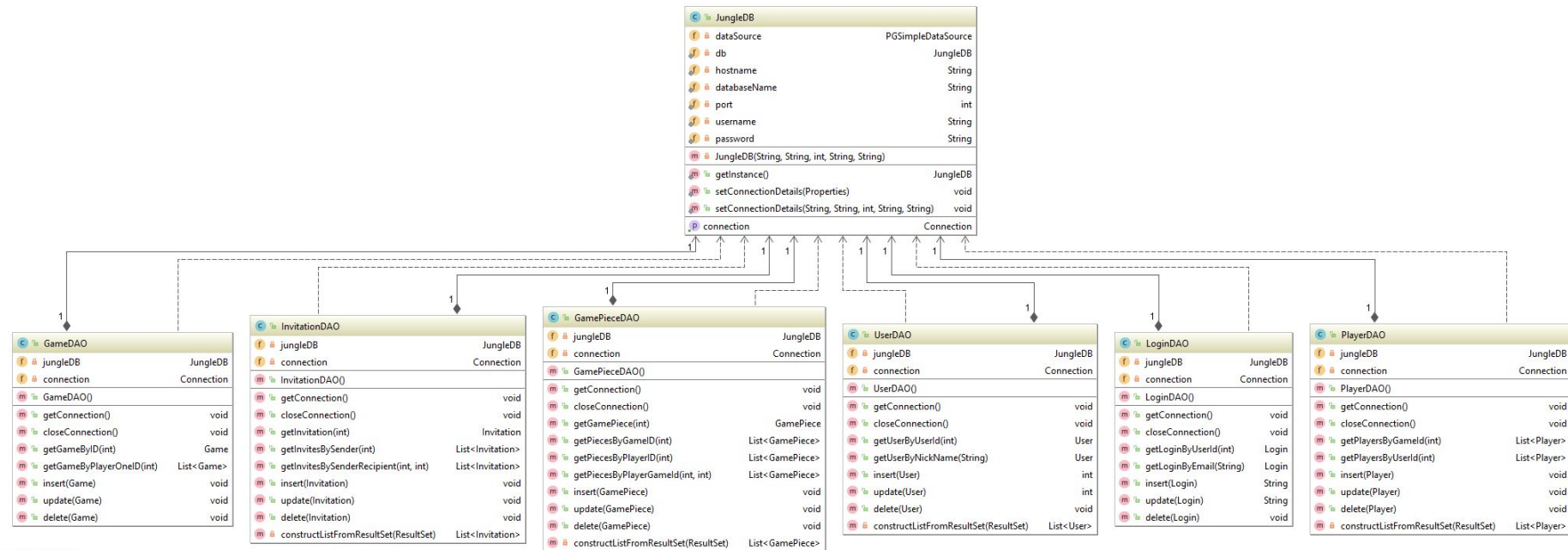
Class Diagrams - Server cont.

Handlers



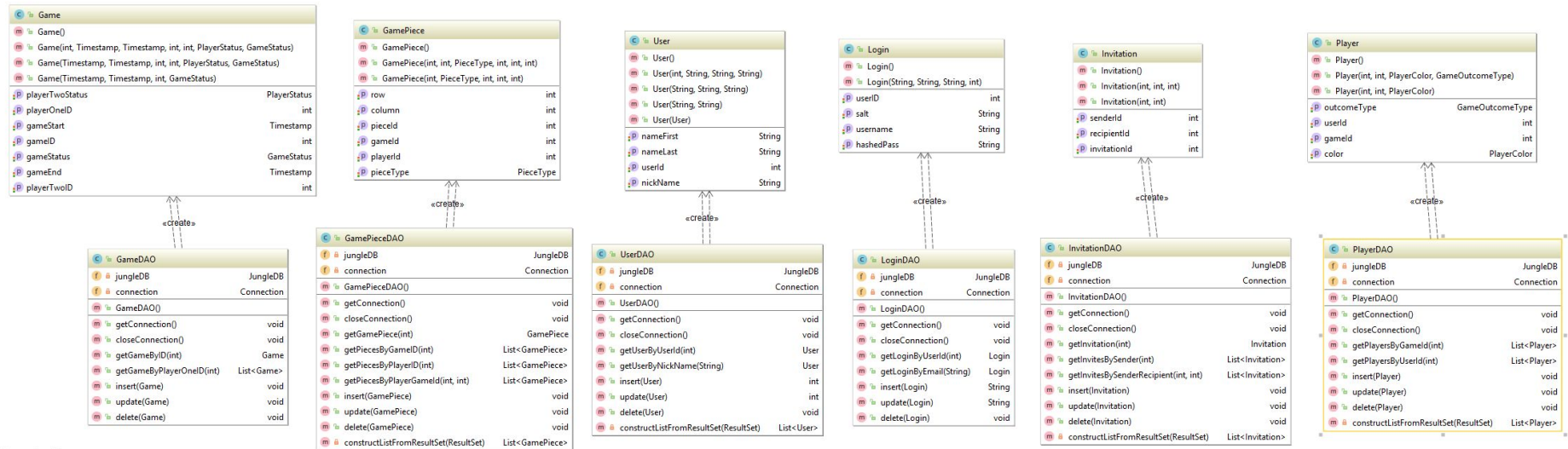
Class Diagrams - Server cont.

Data Access Objects



Class Diagrams - Server cont.

Data Objects/Data Access Objects



Test Case Summary

- **TestBoardSquare:** 16 tests
- **TestGameBoard:** 15 tests
- **TestAccountHandler:** 6 tests
- **TestBoardSquare:** 6 tests

Project Tools

Development Tools

- Eclipse
- IntelliJ IDEA
- Gluon Scene Builder
- Maven
- Git & Github
- Lucidchart

Libraries & Frameworks

- JavaFX
- Mockito
- JUnit
- Kryonet

Other

- DigitalOcean Cloud Hosting
- PostgreSQL
- Docker
- Slack
- Waffle.io
- Travis CI

Traceability Link Matrix

	Game	Game Piece	Invitation	Login	Player	User	Jungle Game	Board Square	Game Board
#1: Register to the system			X	X		X			
#2: Create a new game	X		X	X					X
#3: Invite other users to a game	X		X	X					
#4: Respond to Game Invitation	X		X	X	X				X
#5: Quit Game	X			X	X				X
#6: Unregister from System				X		X			
#7: View Player Profile				X	X				
#8: Log in to System				X		X			
#9: Log out of System				X		X			
#10: Move Game Piece		X		X	X			X	X
#11: Switch Game	X			X	X				X

Use Case Completion

Use Case	Progress Notes
#1: Register to the system	Most server-side logic in place, validation, redirection to new page
#2: Create a new game	Client and GUI logic completed.
#3: Invite other users to a game	Data objects and network handlers in place.
#4: Respond to Game Invitation	Server configuration ready to handle invite send/receive events
#5: Quit Game	Game logic functional, working on server implementation.
#6: Unregister from System	GUI elements done, some server side logic in place
#7: View Player Profile	
#8: Log in to System	GUI elements done, server logic in place
#9: Log out of System	GUI elements done, server logic in place
#10: Move Game Piece	GUI and client logic done, some server logic in place
#11: Switch Game	Basic logic in place, working on server side information

Demo

***Questions
And
Discussion***