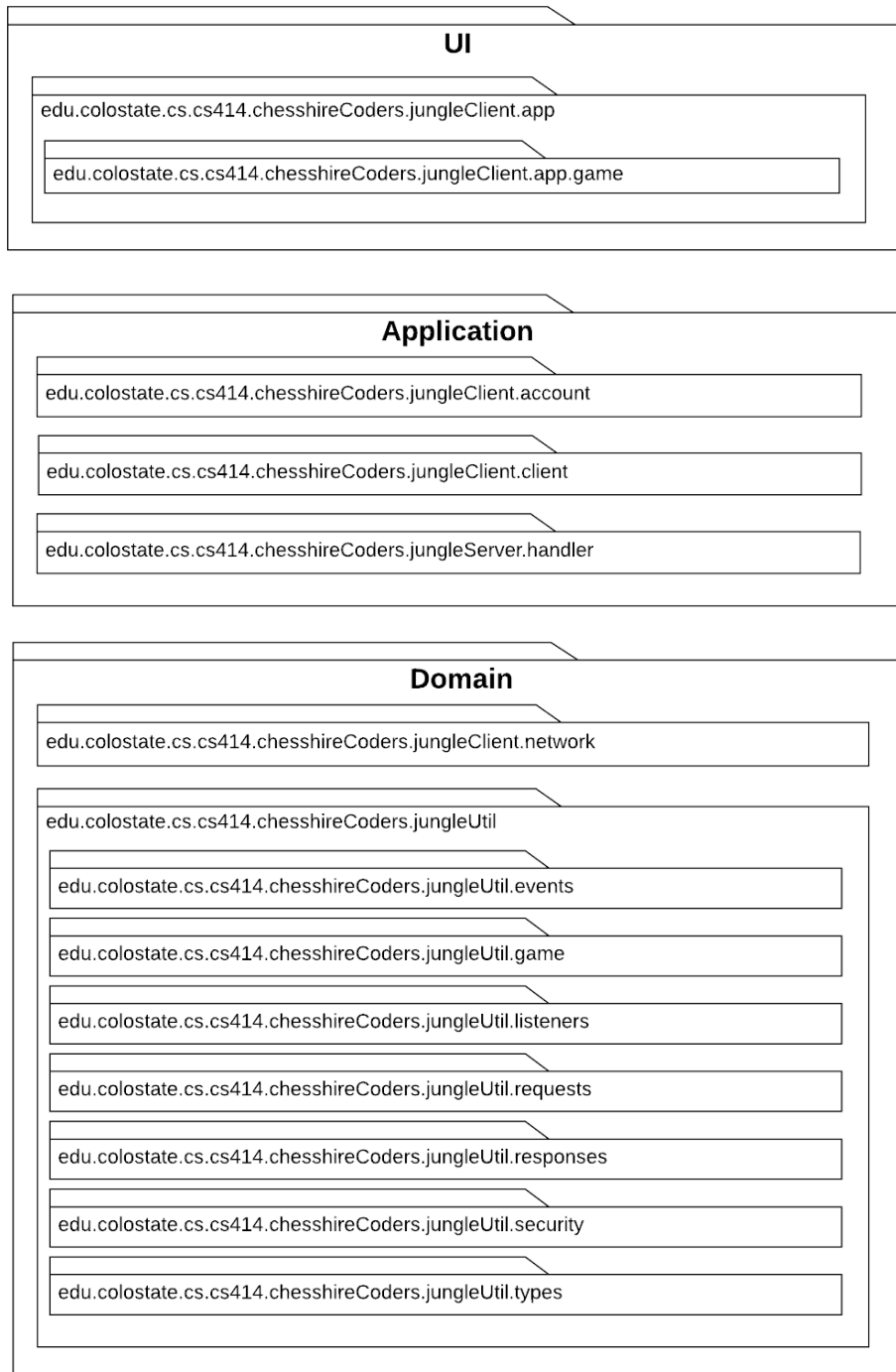# Jungle

Design Document

Package Diagram

UI

edu.colostate.cs.cs414.chesshireCoders.jungleClient.app

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

Application

edu.colostate.cs.cs414.chesshireCoders.jungleClient.account

edu.colostate.cs.cs414.chesshireCoders.jungleClient.client

edu.colostate.cs.cs414.chesshireCoders.jungleServer.handler

Domain

edu.colostate.cs.cs414.chesshireCoders.jungleClient.network

edu.colostate.cs.cs414.chesshireCoders.jungleUtil

edu.colostate.cs.cs414.chesshireCoders.jungleUtil.events

edu.colostate.cs.cs414.chesshireCoders.jungleUtil.game

edu.colostate.cs.cs414.chesshireCoders.jungleUtil.listeners

edu.colostate.cs.cs414.chesshireCoders.jungleUtil.requests

edu.colostate.cs.cs414.chesshireCoders.jungleUtil.responses

edu.colostate.cs.cs414.chesshireCoders.jungleUtil.security

edu.colostate.cs.cs414.chesshireCoders.jungleUtil.types

Technical Services

edu.colostate.cs.cs414.chesshireCoders.jungleServer.service

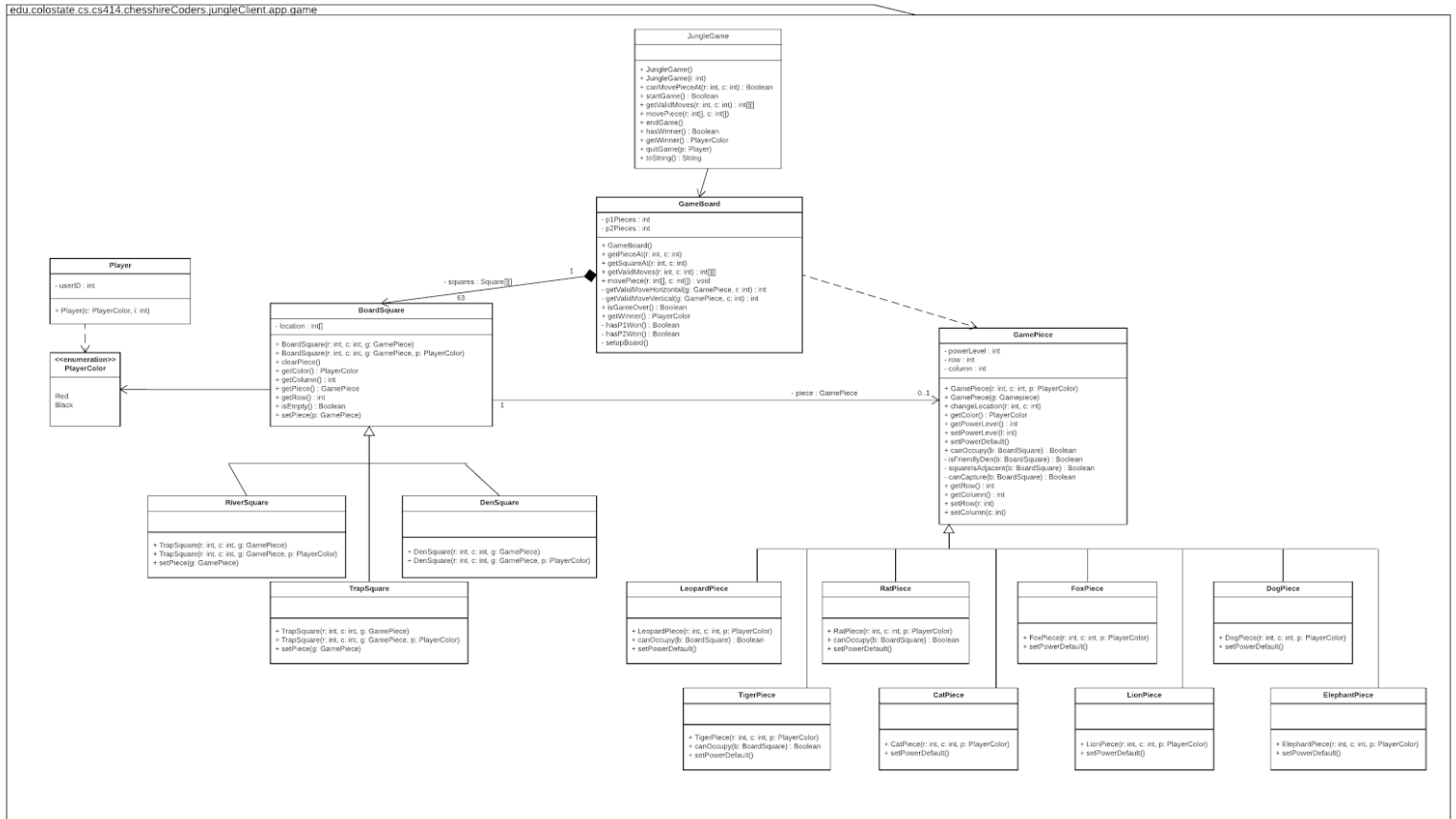edu.colostate.cs.cs414.chesshireCoders.jungleServer.service.impl

Foundation

edu.colostate.cs.cs414.chesshireCoders.jungleServer.persistance

edu.colostate.cs.cs414.chesshireCoders.jungleServer.persistance.dao
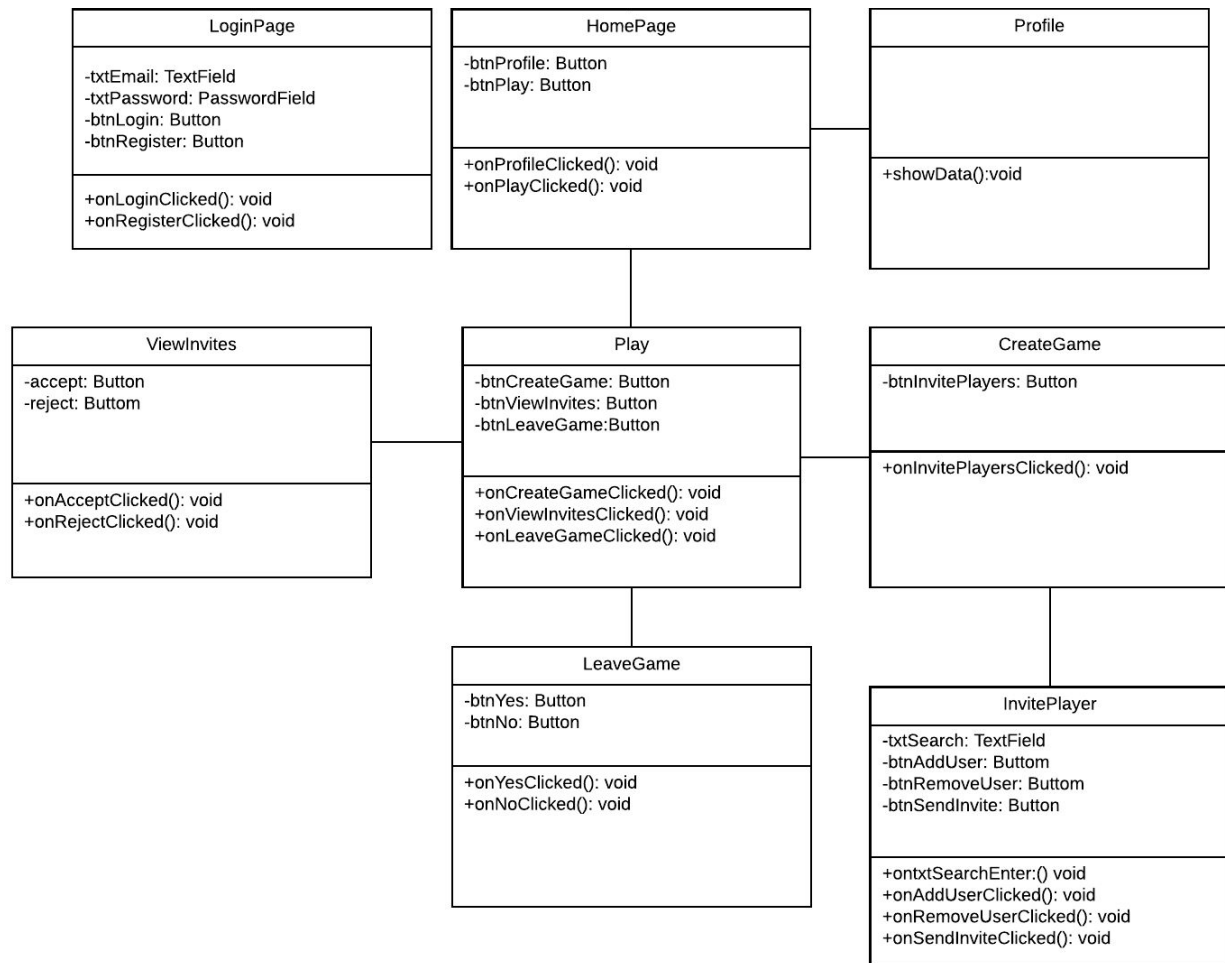
edu.colostate.cs.cs414.chesshireCoders.jungleServer.dao.postgres
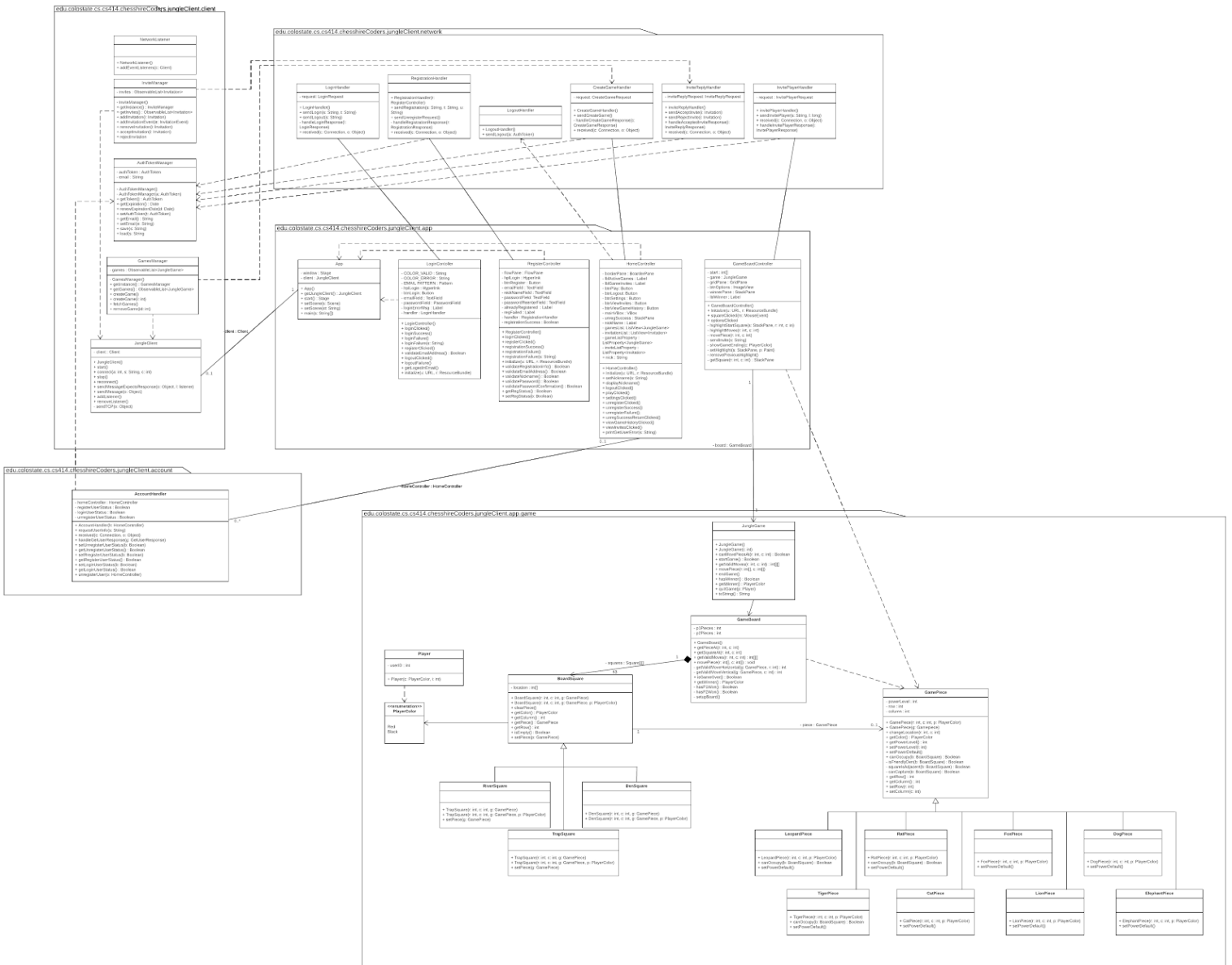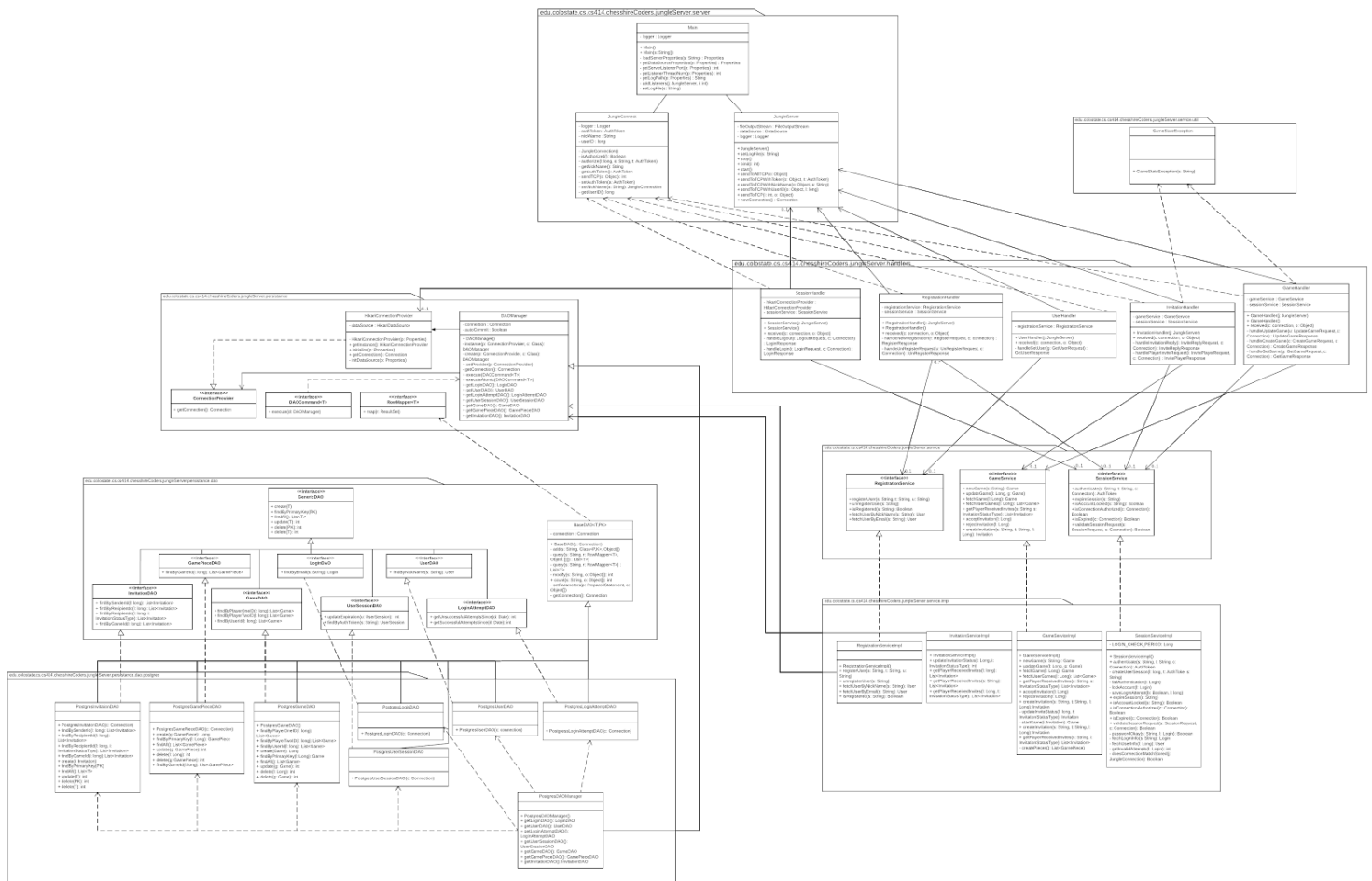
Updated Class Diagram

Game Package Diagram:

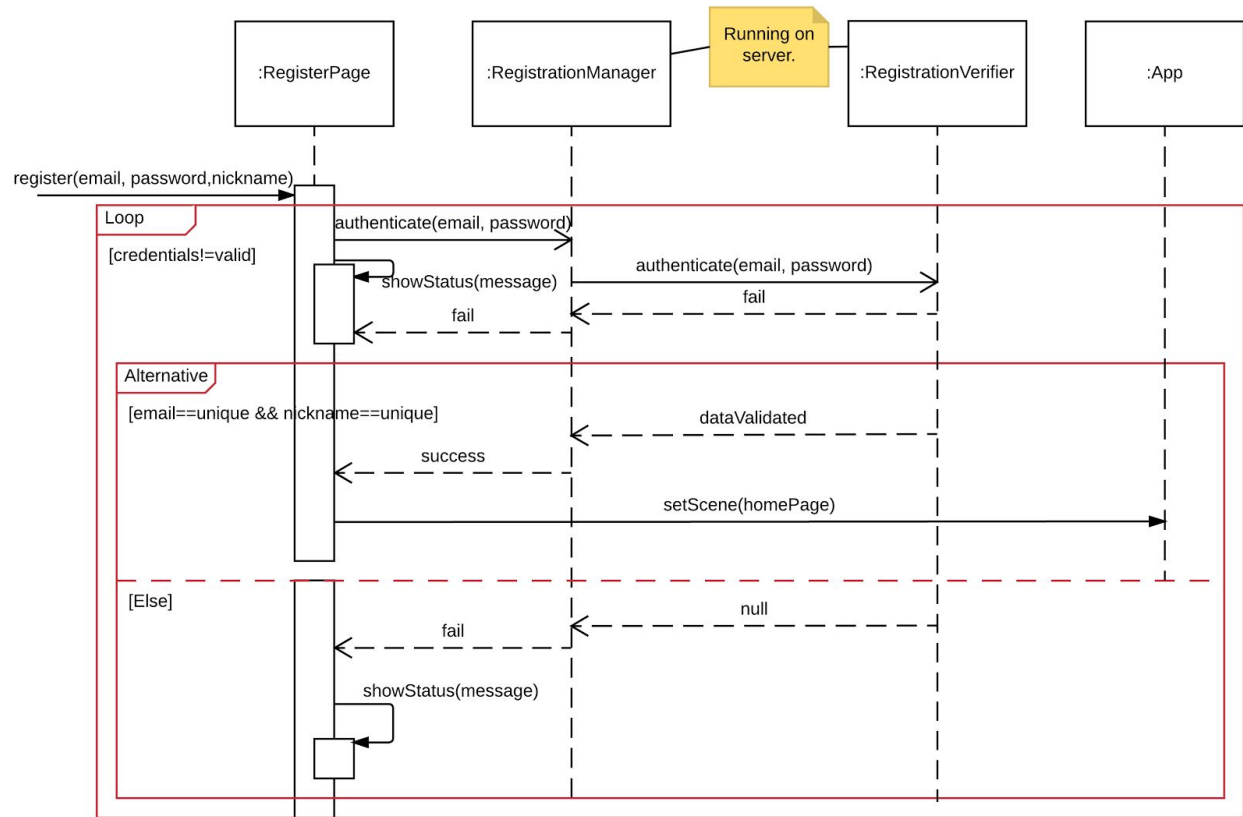edu.colostate.cs.cs414.chesshireCoders.jungleClient.app.game

JungleGame

+ JungleGame()
+ JungleGame(i: int)
+ canMovePieceAt(r: int, c: int) : Boolean
+ startGame() : Boolean
+ getValidMoves(r: int, c: int) : int[][]
+ movePiece(r: int[], c: int[])
+ endGame()
+ hasWinner() : Boolean
+ getWinner() : PlayerColor
+ quitGame(p: Player)
+ toString() : String

GameBoard

- p1Pieces : int
- p2Pieces : int

+ GameBoard()
+ getPieceAt(r: int, c: int)
+ getSquareAt(r: int, c: int)
+ getValidMoves(r: int, c: int) : int[][]
+ movePiece(r: int[], c: int[]) : void
- getValidMoveHorizontal(g: GamePiece, r: int) : int
- getValidMoveVertical(g: GamePiece, c: int) : int
+ isGameOver() : Boolean
+ getWinner() : PlayerColor
- hasP1Won() : Boolean
- hasP2Won() : Boolean
- setupBoard()

Player

- userID : int

+ Player(c: PlayerColor, i: int)

<<enumeration>> PlayerColor

Red
Black

BoardSquare

- location : int[]

+ BoardSquare(r: int, c: int, g: GamePiece)
+ BoardSquare(r: int, c: int, g: GamePiece, p: PlayerColor)
+ clearPiece()
+ getColor() : PlayerColor
+ getColumn() : int
+ getPiece() : GamePiece
+ getRow() : int
+ isEmpty() : Boolean
+ setPiece(p: GamePiece)

- squares : Square[][]   1   63

- piece : GamePiece   0..1   1

GamePiece

- powerLevel : int
- row : int
- column : int

+ GamePiece(r: int, c: int, p: PlayerColor)
+ GamePiece(g: Gamepiece)
+ changeLocation(r: int, c: int)
+ getColor() : PlayerColor
+ getPowerLevel() : int
+ setPowerLevel(l: int)
+ setPowerDefault()
+ canOccupy(b: BoardSquare) : Boolean
- isFriendlyDen(b: BoardSquare) : Boolean
- squareIsAdjacent(b: BoardSquare) : Boolean
- canCapture(b: BoardSquare) : Boolean
+ getRow() : int
+ getColumn() : int
+ setRow(r: int)
+ setColumn(c: int)

RiverSquare

+ TrapSquare(r: int, c: int, g: GamePiece)
+ TrapSquare(r: int, c: int, g: GamePiece, p: PlayerColor)
+ setPiece(g: GamePiece)

DenSquare

+ DenSquare(r: int, c: int, g: GamePiece)
+ DenSquare(r: int, c: int, g: GamePiece, p: PlayerColor)

TrapSquare

+ TrapSquare(r: int, c: int, g: GamePiece)
+ TrapSquare(r: int, c: int, g: GamePiece, p: PlayerColor)
+ setPiece(g: GamePiece)

LeopardPiece

+ LeopardPiece(r: int, c: int, p: PlayerColor)
+ canOccupy(b: BoardSquare) : Boolean
+ setPowerDefault()

RatPiece

+ RatPiece(r: int, c: int, p: PlayerColor)
+ canOccupy(b: BoardSquare) : Boolean
+ setPowerDefault()

FoxPiece

+ FoxPiece(r: int, c: int, p: PlayerColor)
+ setPowerDefault()

DogPiece

+ DogPiece(r: int, c: int, p: PlayerColor)
+ setPowerDefault()

TigerPiece

+ TigerPiece(r: int, c: int, p: PlayerColor)
+ canOccupy(b: BoardSquare) : Boolean
+ setPowerDefault()

CatPiece

+ CatPiece(r: int, c: int, p: PlayerColor)
+ setPowerDefault()

LionPiece

+ LionPiece(r: int, c: int, p: PlayerColor)
+ setPowerDefault()

ElephantPiece

+ ElephantPiece(r: int, c: int, p: PlayerColor)
+ setPowerDefault()

3

Front End:

LoginPage

-txtEmail: TextField
-txtPassword: PasswordField
-btnLogin: Button
-btnRegister: Button

+onLoginClicked(): void
+onRegisterClicked(): void

HomePage

-btnProfile: Button
-btnPlay: Button

+onProfileClicked(): void
+onPlayClicked(): void

Profile

+showData():void

ViewInvites

-accept: Button
-reject: Buttom

+onAcceptClicked(): void
+onRejectClicked(): void

Play

-btnCreateGame: Button
-btnViewInvites: Button
-btnLeaveGame:Button

+onCreateGameClicked(): void
+onViewInvitesClicked(): void
+onLeaveGameClicked(): void

CreateGame

-btnInvitePlayers: Button

+onInvitePlayersClicked(): void

LeaveGame

-btnYes: Button
-btnNo: Button

+onYesClicked(): void
+onNoClicked(): void

InvitePlayer

-txtSearch: TextField
-btnAddUser: Buttom
-btnRemoveUser: Buttom
-btnSendInvite: Button

+ontxtSearchEnter:() void
+onAddUserClicked(): void
+onRemoveUserClicked(): void
+onSendInviteClicked(): void

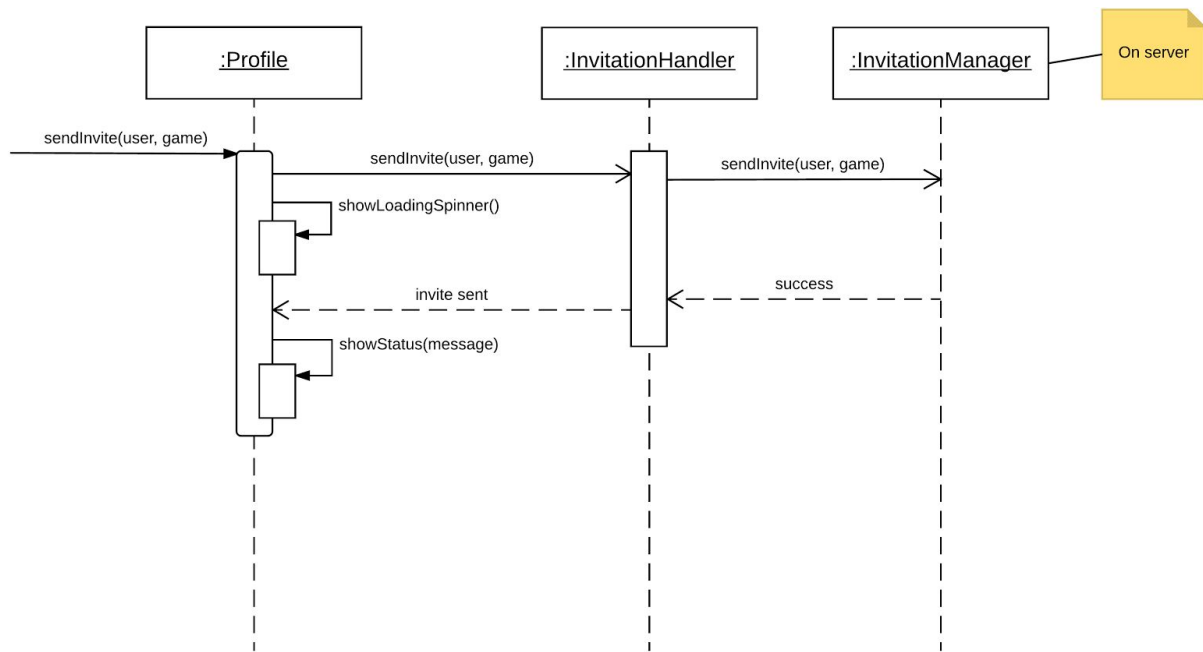# JungleClient:
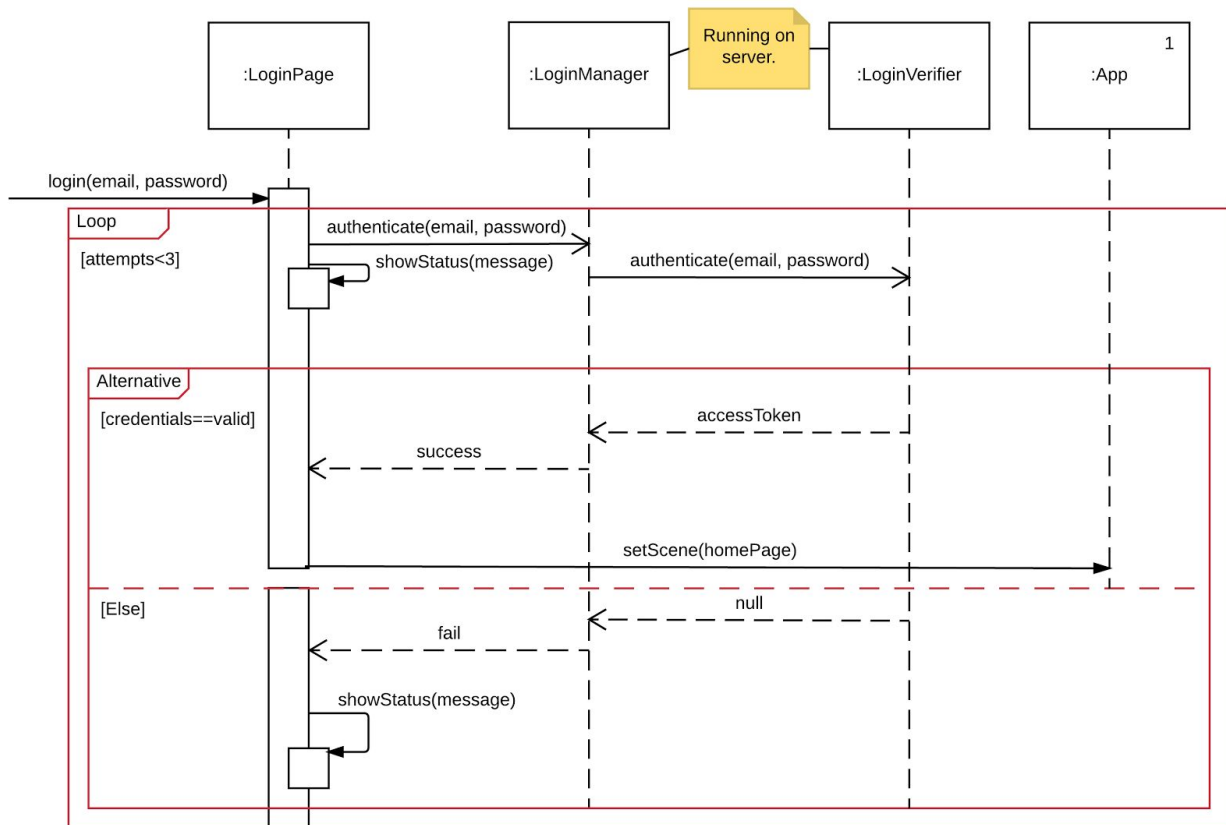
JungleServer:

Updated Sequence Diagram

Register:

Take Turn:

Game Invite:

Login:

Create Game: