# The Jungle Game

#### **Chesshire Coders**

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



## Logical Architecture

#### UI

edu. colostate. cs. cs414. ches shire Coders. jungle Client. ui

#### Application

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

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

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

## Logical Architecture

#### **Technical Services**

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

edu. colostate. cs. cs 414. ches shire Coders. jungle Server. service. impl

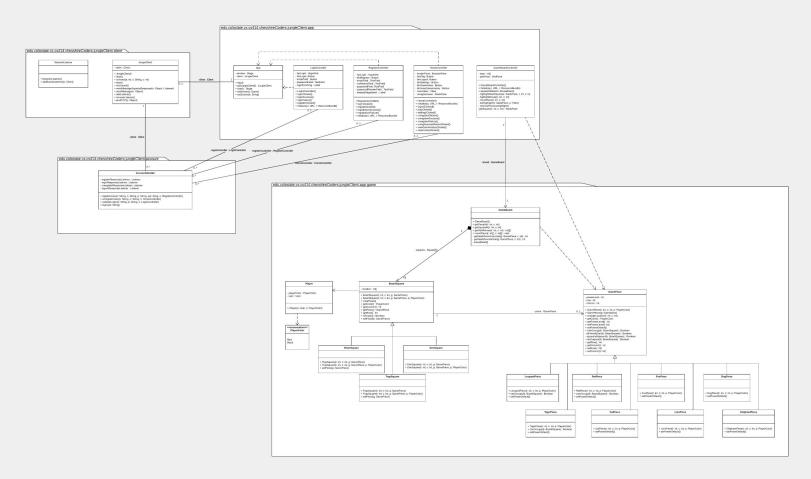
#### **Foundation**

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

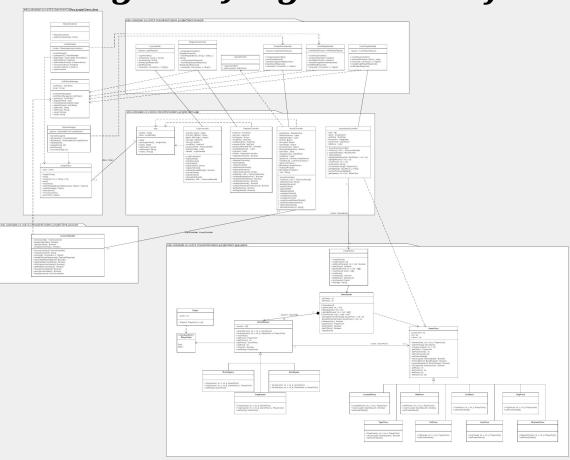
edu. colo state. cs. cs 414. ches shire Coders. jungle Server. per sistance. dao

edu.colostate.cs.cs414.chesshireCoders.jungleServer.dao.postgres

## Design Class Diagram: Jungle Client - Before



## Design Class Diagram: Jungle Client - After



## Design Class Diagram: Jungle Game

#### Addition of Singleton Manager Classes

#### AuthTokenManager

- authToken: AuthToken email: String
- AuthTokenManager()
- AuthTokenManager(a: AuthToken)
- + getToken() : AuthToken
- + getExpiration(): Date
- + renewExpirationDate(d: Date)
- + setAuthToken(t; AuthToken)
- + getEmail() : String
- + setEmail(e: String)
- + save(s: String) + load(s: String

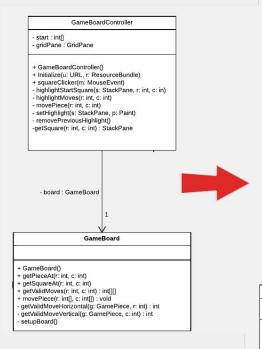
#### InviteManager

- invites : ObservableList<Invitation>
- InviteManager()
- + getInstance() : InviteManager
- + getInvites(): ObservableList<Invitation>
- + addInvitation(i: Invitation)
- + addInvitationEvent(e: InvitationEvent)
- + removeInvitation(i: Invitation)
- + acceptInvitation(i: Invitation)
- + rejectInvitation

#### GamesManager

- games : ObservableList<JungleGame>
- GamesManager()
- + getinstance() : GamesManager
- + getGames(): ObservableList<JungleGame>
- + createGame()
- + createGame(i: int)
- + fetchGames()
- + removeGame(id: int)

#### Addition of JungleGame Class



#### GameBoardController

- start : int∏
- game : JungleGame
- gridPane : GridPane
- btnOptions : ImageView
- winnerPane : StackPane |blWinner:Label
- + GameBoardController()
- + Initialize(u: URL, r: ResourceBundle)
- + squareClicked(m: MouseEvent)
- + optionsClicked highlightStartSquare(s: StackPane, r: int, c: in)
- highlightMoves(r: int, c: int)
- movePiece(r: int, c: int)
- sendInvite(s: String)
- showGameEnding(c: PlayerColor)
- setHighlight(s: StackPane, p: Paint)
- removePreviousHighlight()
- getSquare(r: int, c: int) : StackPane

### JungleGame

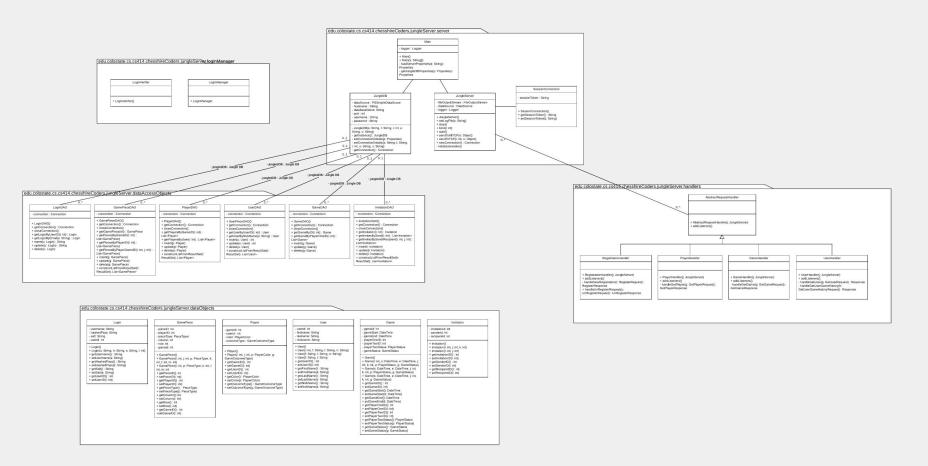
- + JungleGame()
- + JungleGame(i: int)
- + canMovePieceAt(r: int, c: int) : Boolean
- + startGame(): Boolean
- + getValidMoves(r; int, c; int) : int[][]
- + movePiece(r: int[], c: int[])
- + endGame()
- + hasWinner() : Boolean
- + getWinner() : PlayerColor
- + quitGame(p: Player) + toString() : String

#### GameBoard

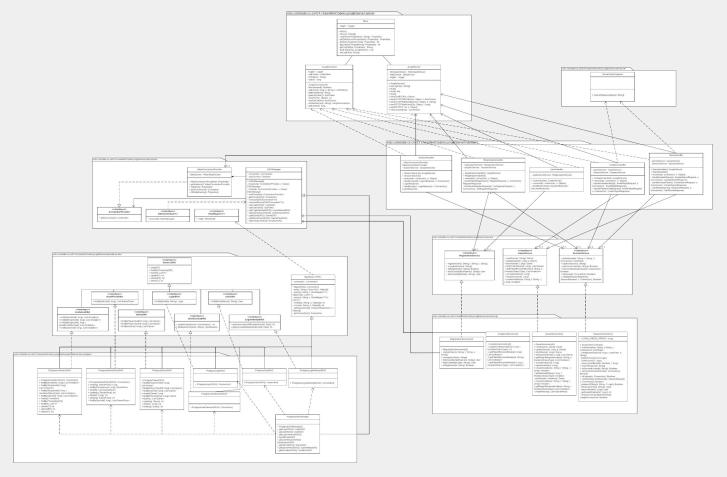
#### p2Pieces: int

- p1Pieces : int + GameBoard()
- + getPieceAt(r: int, c: int)
- + getSquareAt(r: int, c: int) + getValidMoves(r: int, c: int) : int[][]
- + movePiece(r: intfl, c: intfl) : void
- getValidMoveHorizontal(g: GamePiece, r: int) : int
- getValidMoveVertical(g: GamePiece, c: int) : int + isGameOver() : Boolean
- + getWinner(): PlayerColor
- hasP1Won() : Boolean
- hasP2Won() : Boolean
- setupBoard()

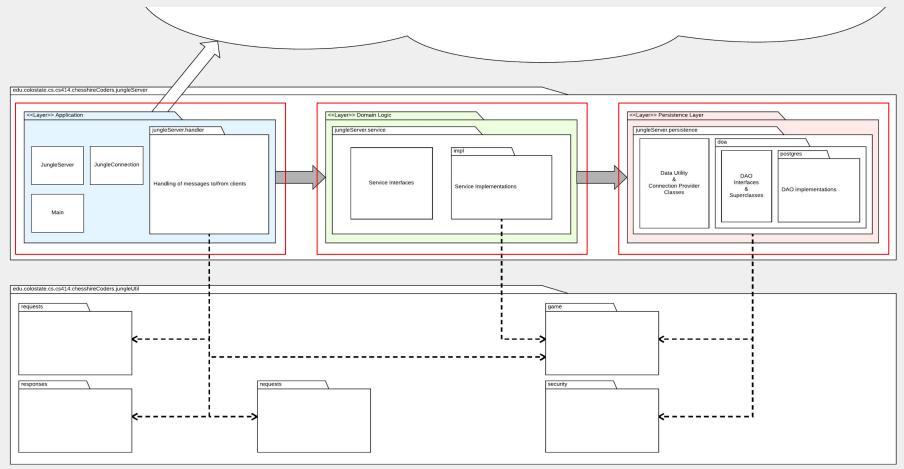
## Design Class Diagram: Jungle Network - Before



## Design Class Diagram: Jungle Server - After

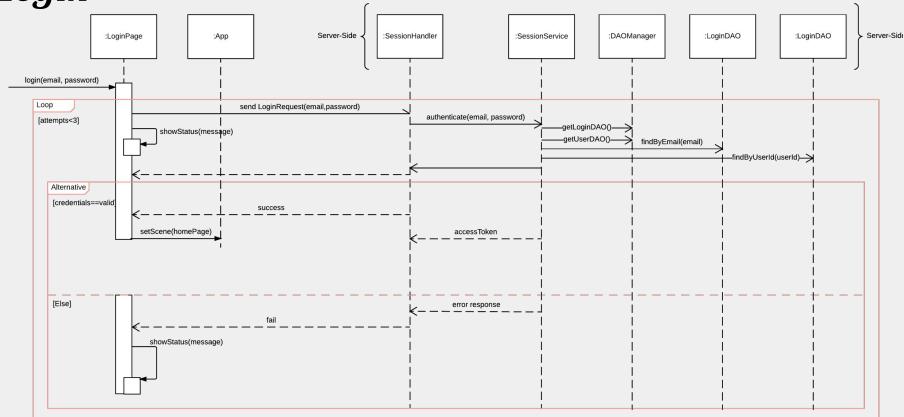


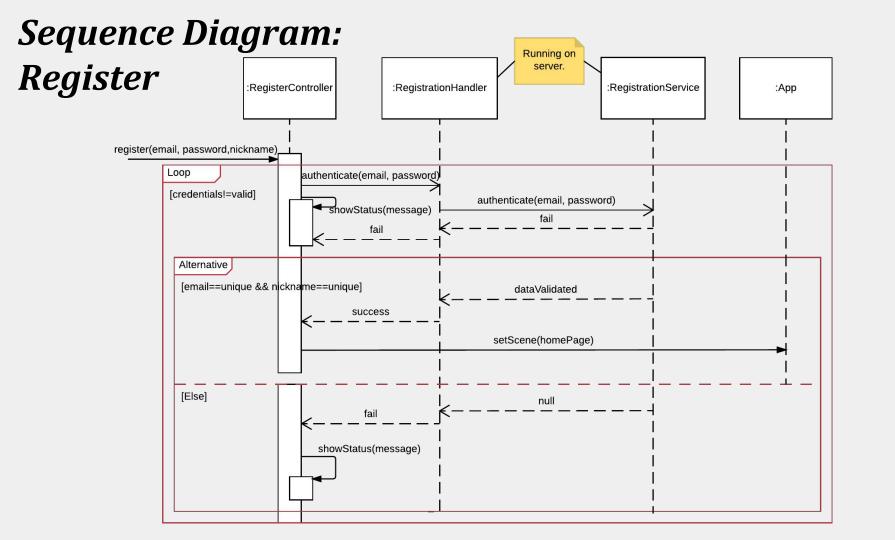
## Server Structure & Packages



Sequence Diagram:

Login





## Refactoring and Design Pattern List

- > Singleton
- Factory Method
- Abstract Factory Method
- > Prototype
- > Facade
- > Decorator

## Traceability Link Matrix

		T		C	D 3			Towns to	Township and
	Login	Jungle Game	Game Board	Game Piece	Board Square	Player	User	Invite Manager	Invitatio n
#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	X			
#11: Switch Game	X	X	X			X			

## Demo

# Questions And Discussion