

A dark, moody photograph of chess pieces. In the foreground, a small, dark chess piece (likely a pawn) is visible. To the right, a much larger, dark chess piece (likely a king or queen) stands tall, casting a long, sharp shadow across the dark surface. The background is a dark, textured wall. The overall lighting is low, creating a dramatic and mysterious atmosphere.

Portal Chess

Strike Team Java

Introduction to XGame



Play Portal
Chess online



Challenge
other users



Create an
account



View your
match history

What is Portal Chess?

- Has the same base rules as ordinary chess
- Each player has two portal pieces they start with, can place them anywhere on the fourth row at the start of the game
- Player can only move their own colored portal, same moves as a King
- If a piece lands on a portal, then it is teleported to the other portal of the same color unless it is blocked. Piece comes out of portal in same direction it moved into the portal.

High Level Project Design



UI will be built with React and will send API requests to the JAVA server to obtain information to be displayed to the user



Restful API that will provide all the information about the application to the frontend. All logic for gameplay will be here.



Database will store information about games, invitation, and users

User Stories

Must Have

- Create an account
 - Register email
 - Create username
- Log in to the account
- Create match
- Invite opponents to the match
- Start game when the first opponent accepts the invitation
- Play Portal Chess

Should Have

- Send invites to multiple users
- Accept/ reject invites from another user
- Close and return to matches at will
- Know when a game is over and who has won

User Stories (cont.)

Could Have

- Be able to log out
- Unregister account
- Switch between multiple games at once
- Abandon a game at any time

Would Have

- View game history- Players, start and end date/times, winner/loser of the match, whether a game was abandoned
- Be able to tell whose turn it is

CRC Cards

Account	
<ul style="list-style-type: none">• Create account• Login to account• Log out of account• Unregister account	<ul style="list-style-type: none">• Player
Player	
<ul style="list-style-type: none">• Accept invites• Reject invites• Notify inviter if invite is rejected	<ul style="list-style-type: none">• Account• MatchFacilitator
Profile	
<ul style="list-style-type: none">• Contains win/loss history from matches• Contains abandoned match history	
MatchFacilitator	
<ul style="list-style-type: none">• Create match• Close match• Abandon match• Resume saved match• Switch between matches	<ul style="list-style-type: none">• Player

CRC Cards

Match		Game
<ul style="list-style-type: none">• Save state of the match in progress• Start when enough players join• Determine if game is over• Reflect victor of game determined by Game (if applicable)• Keep track of active player	<ul style="list-style-type: none">• MatchFacilitator	

History		
<ul style="list-style-type: none">• Stores match history	<ul style="list-style-type: none">• Profile	

Abstract	Game		Match
<ul style="list-style-type: none">• Determine victor (if applicable)	<ul style="list-style-type: none">• Match		

Inbox		
<ul style="list-style-type: none">• Contains notifications		

Invite		Notification
<ul style="list-style-type: none">• Be sent to player's inbox	<ul style="list-style-type: none">• Player	

Abstract	Notification		Invite
<ul style="list-style-type: none">• Be sent to player's inbox	<ul style="list-style-type: none">• Player		

CRC Cards

ChessBoard	
• Show location of each piece	• ChessPiece

Abstract ChessPiece Knight, Pawn, King, Queen, Rook, Bishop, Portal	
• Store color	<ul style="list-style-type: none">• Knight• Pawn• King• Queen• Rook• Bishop• Portal

Knight ChessPiece	
• Determine valid moves	

Pawn ChessPiece	
• Determine valid moves	

Bishop ChessPiece	
• Determine valid moves	

CRC Cards

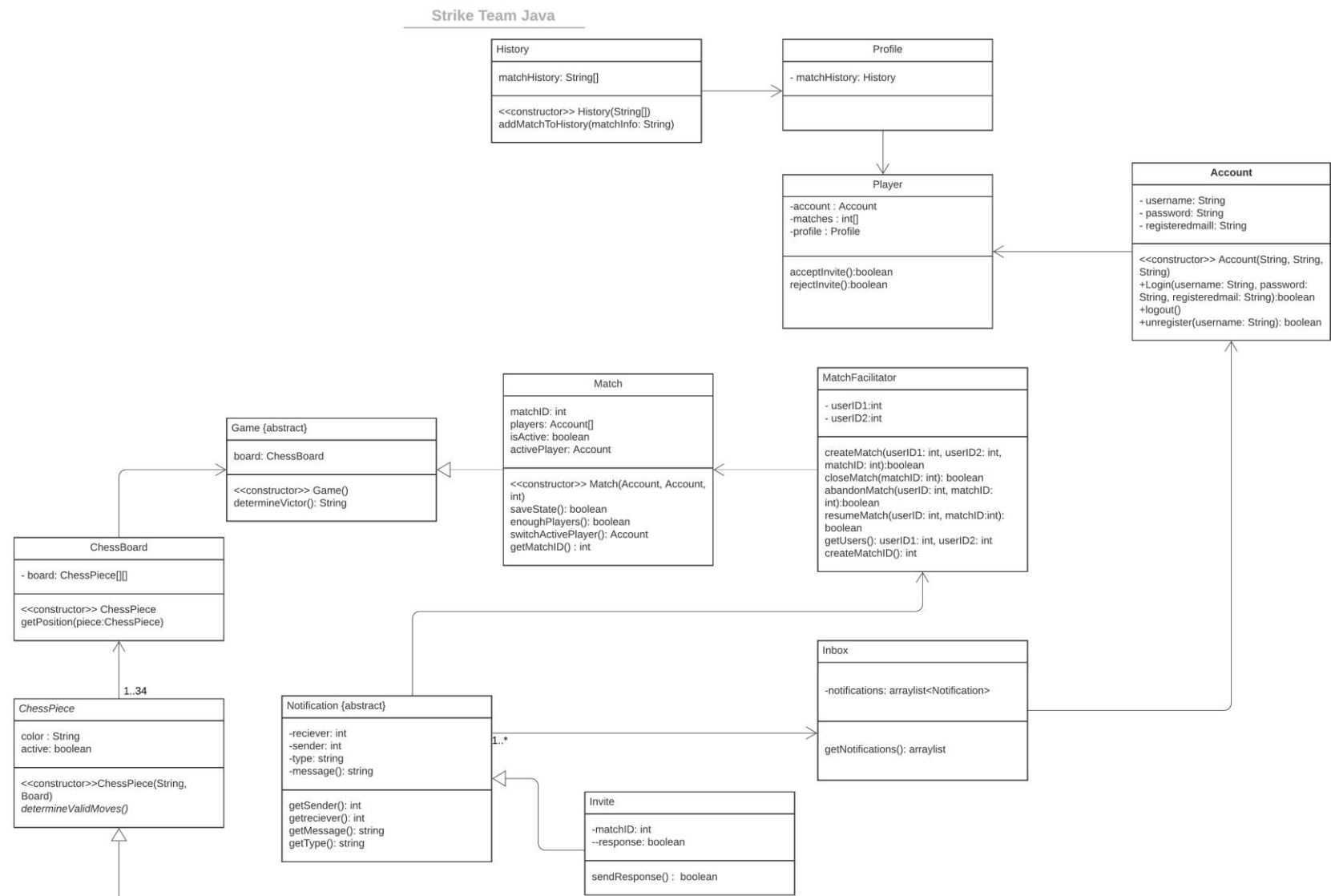
King		ChessPiece
• Determine valid moves		

Queen		ChessPiece
• Determine valid moves		

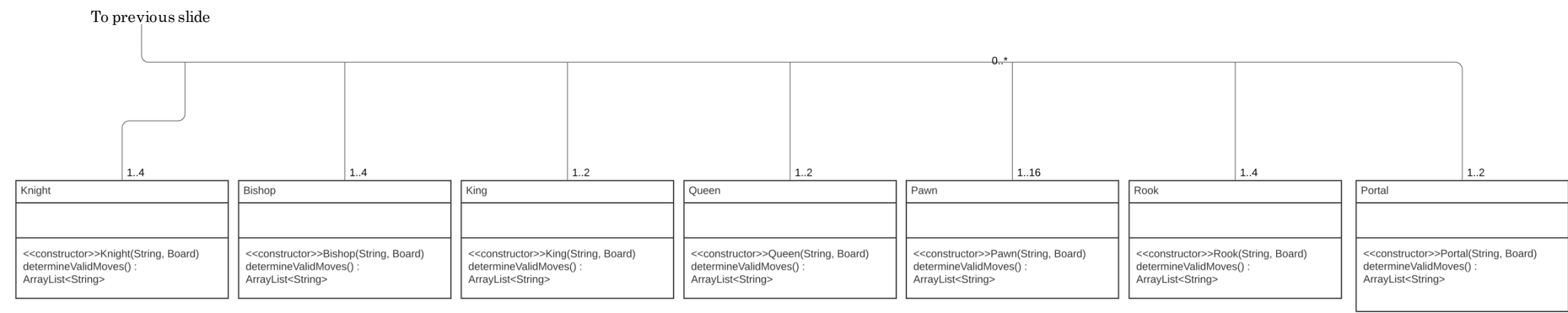
Rook		ChessPiece
• Determine valid moves		

Portal		ChessPiece
• Determine valid moves		

Class Diagram

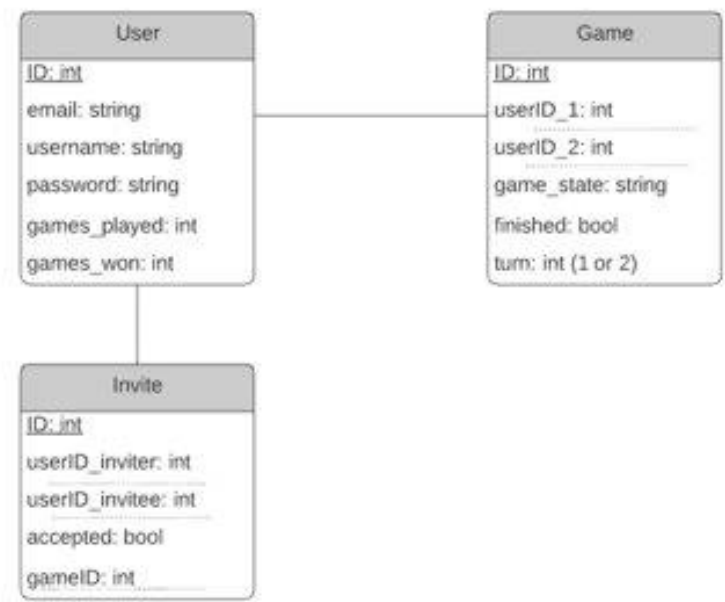


Class Diagram



Strike Team Java

Database Diagram



Kanban Board

Search or jump to...

Pull requestsIssuesMarketplaceExplore

mdcham / cs414-f20-Strike-Team-Java

/ cs414-f20-Strike-Team-Java

Private

Unwatch1Star0Fork0

<> CodeIssues36Pull requestsZenHubActionsProjectsWikiSecurityInsightsSettings

cs414-f20-Strike-Team-Java

Repos(1/1)LabelsMilestonesAssigneesEpicsReleasesEstimatesAuthorsFind Issues(f+i)New Issue+

Customize pipeline name and descriptions to match your team's workflow. Click on the ⓘ next to each pipeline to make changes. Learn more

Icebox0 Issues - 0 Story Points

Backlog33 Issues - 10 Story Points

Current Sprint3 Issues - 2 Story Points

In Progress1 Issue - 0 Story Points

Closed10+ Issues - 4 Story Points

Create...Edit WorkspaceInviteView TutorialsShortcutsOpen in web appSupport and training

Michael Cunninghammdcham

Scrum results

- 09/10/2020 - Setup Github repo, created epics/tasks in Zenhub
- 09/14/2020 - Created questions and scheduled a meeting with PO
- 09/17/2020 - Meeting with PO, reworked user stories gave each of them a priority, got answers to questions
- 09/22/2020 - Created CRC cards for all the planned classes, closed issue #40
- 09/22/2020 - Created wiki page on Github, started uploading documents to Github, started on UML diagram
- 09/23/2020 - Finalized UML diagram, discussed presentation possibilities
- 09/24/2020 - Created PowerPoint presentation, set video recording time for everyone

Going Forward

- Setup rules for following proper Git etiquette
 - Reviewing other users' code
 - Never updating Master directly
- Setting up a Maven build for server
- Create a React app for the client