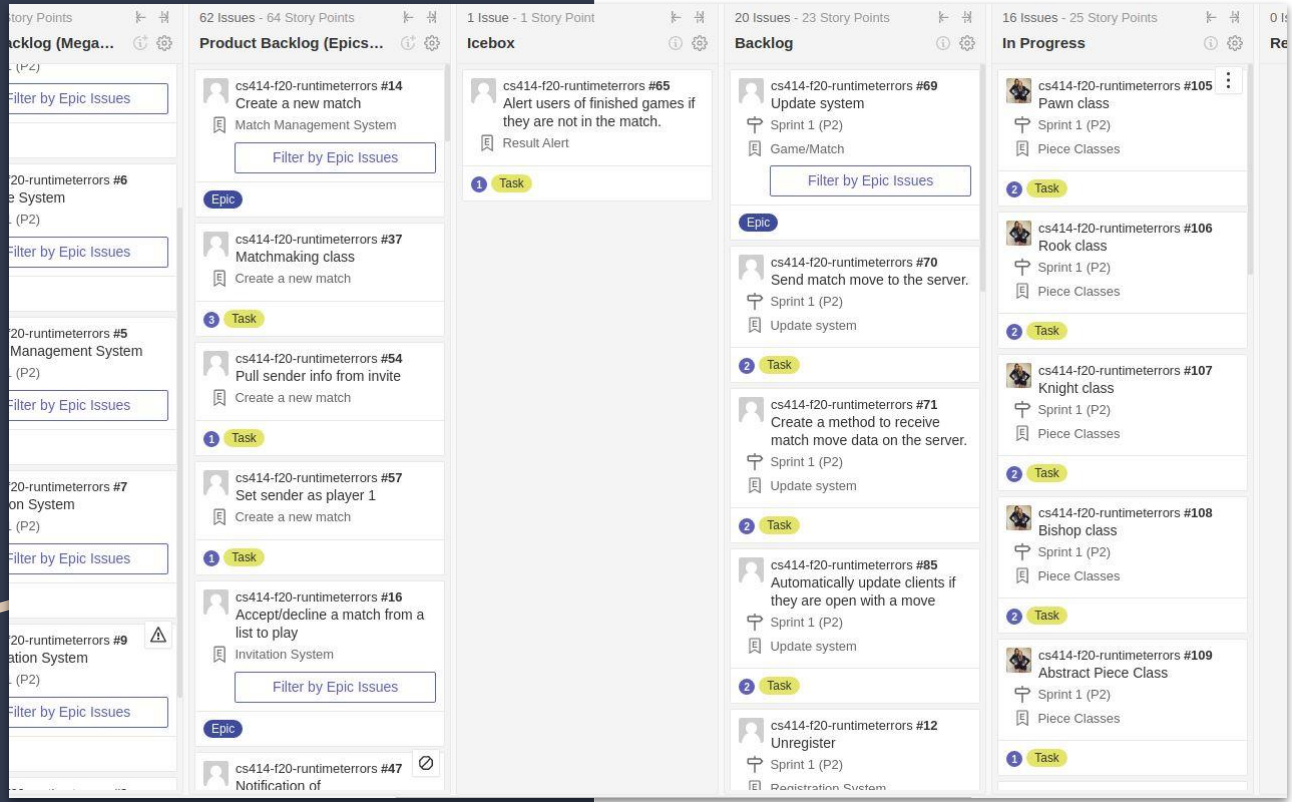
P2 - Runtime Terrors

Darla Drenckhahn
Patrick Lee
Garrett Olsen
Daniel Stephenson
Falkyn Worm

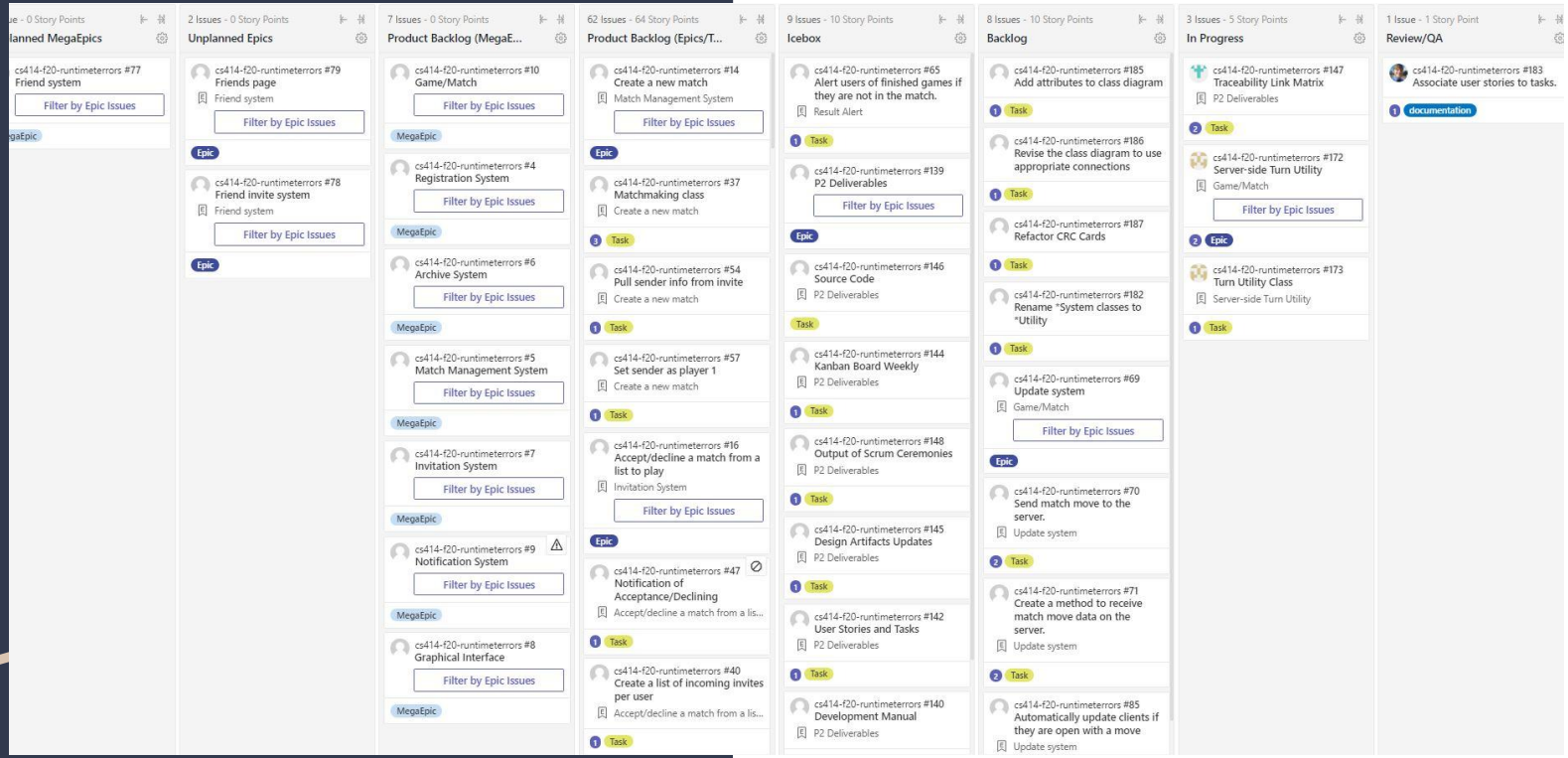
Omega Chess Recap

- Omega Chess is played on a 10x10 board and four “Wizard” squares at each corner.
- It has the same rules as chess except for the behavior of Pawns. The pawn can only move forward and in its initial position it can move one, two or three squares forward.
- There are two additional pieces in Omega Chess, the Champion and the Wizard.
- The Champion is like the Knight. It can move one square orthogonally forward, backward or to either side. It can also jump over these squares and also jump two square diagonally in all four directions.
- The Wizard can move one square diagonally in all four directions, or it can move as a Knight would, but three squares in one direction instead of two.

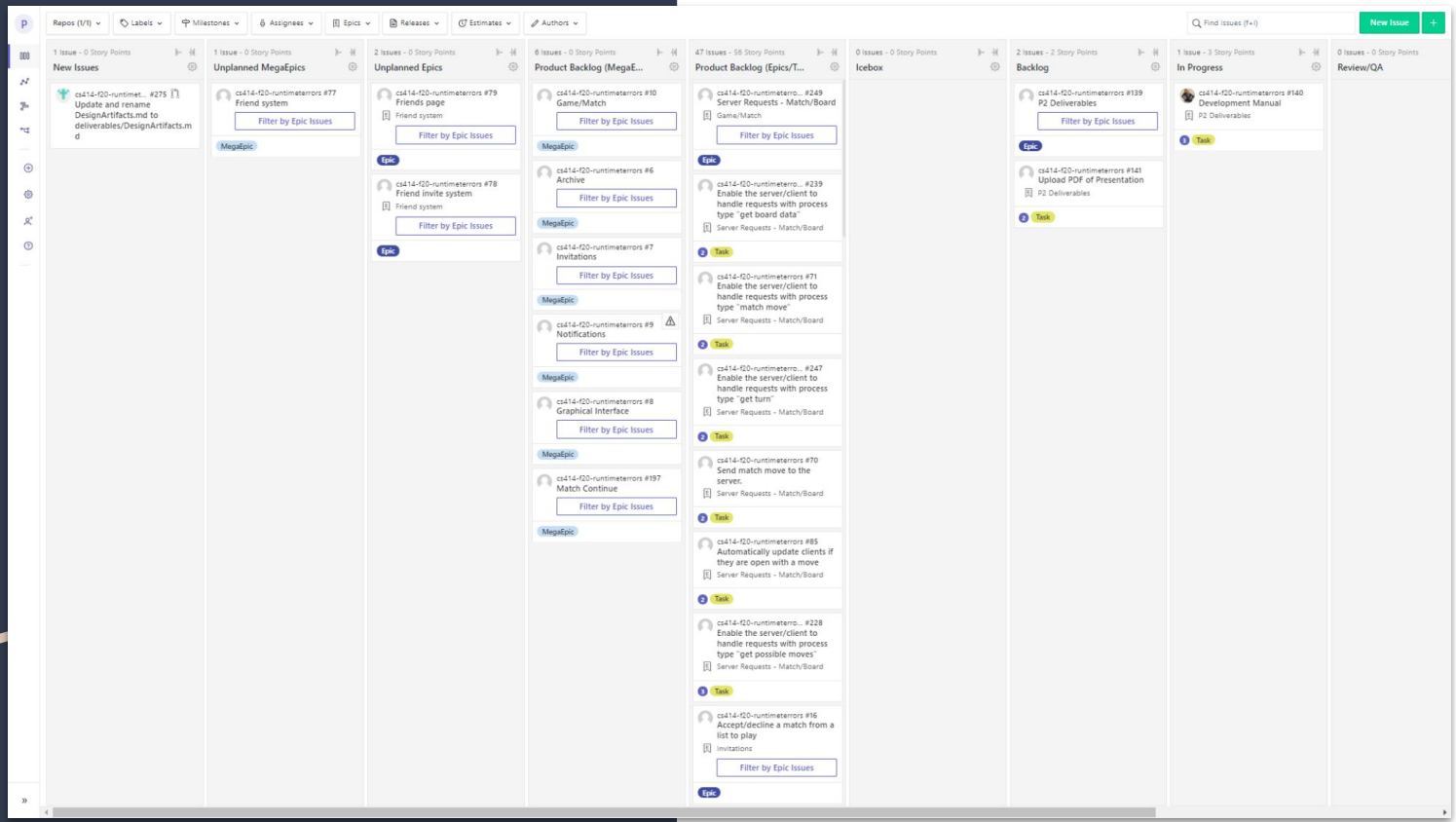
Kanban Board Start of P2



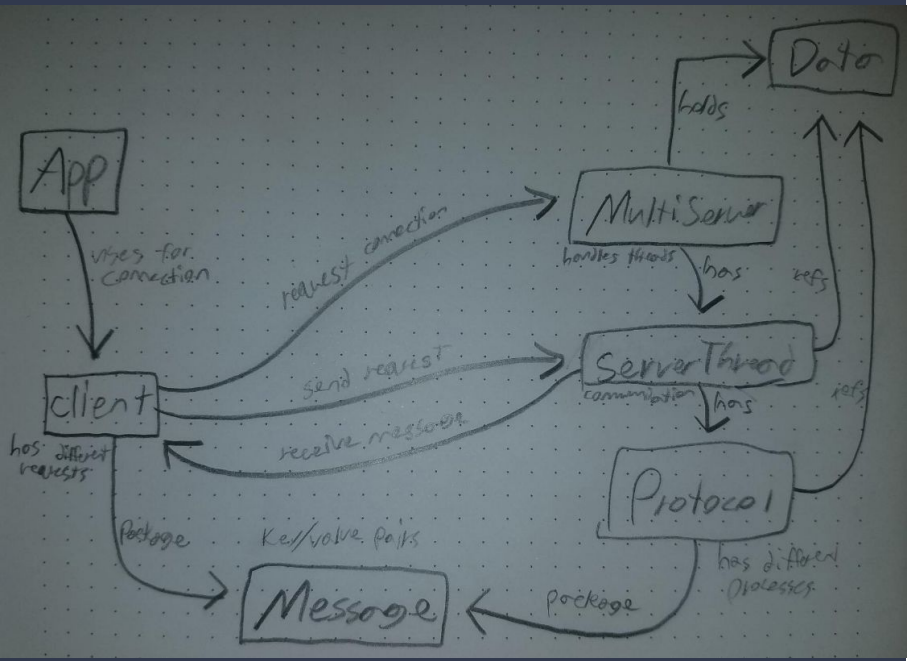
Kanban Board Middle of P2 (Sprint 1)



Kanban Board End of P2 (Sprint 2)



Server Setup



The **MultiServer** was made using java sockets. It waits for new connections to pair with a new **ServerThread**.

The **App** uses a **Client** class as its interface to the **MultiServer** and calls request methods based on user input.

The **Client** establishes a connection with the **MultiServer** and gets assigned a **ServerThread**. It has requests that it can send to the **ServerThread** on behalf of the **App**.

The **ServerThread** reads the content of requests and sends responses back to the **Client** according to the **Protocol**.

The **Protocol** accesses the **Data** in order to fulfill requests.

Both the **Client** and the **Protocol** package messages to be sent using a custom **Message** class, which is essentially a String-to-String HashMap that is capable of being turned into a string and back.

Current Supported Requests

Register

Unregister

Login

Get Profile Data

Send Invite

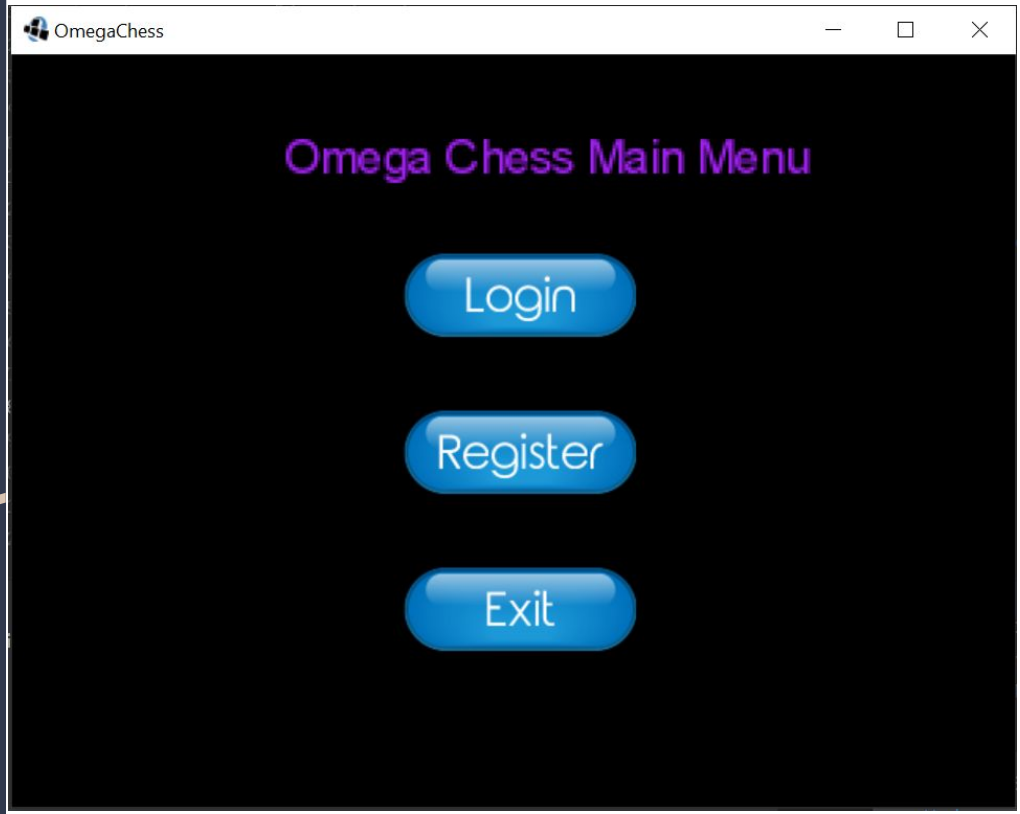
Get Sent Invites

Get Received Invites

Get Notifications

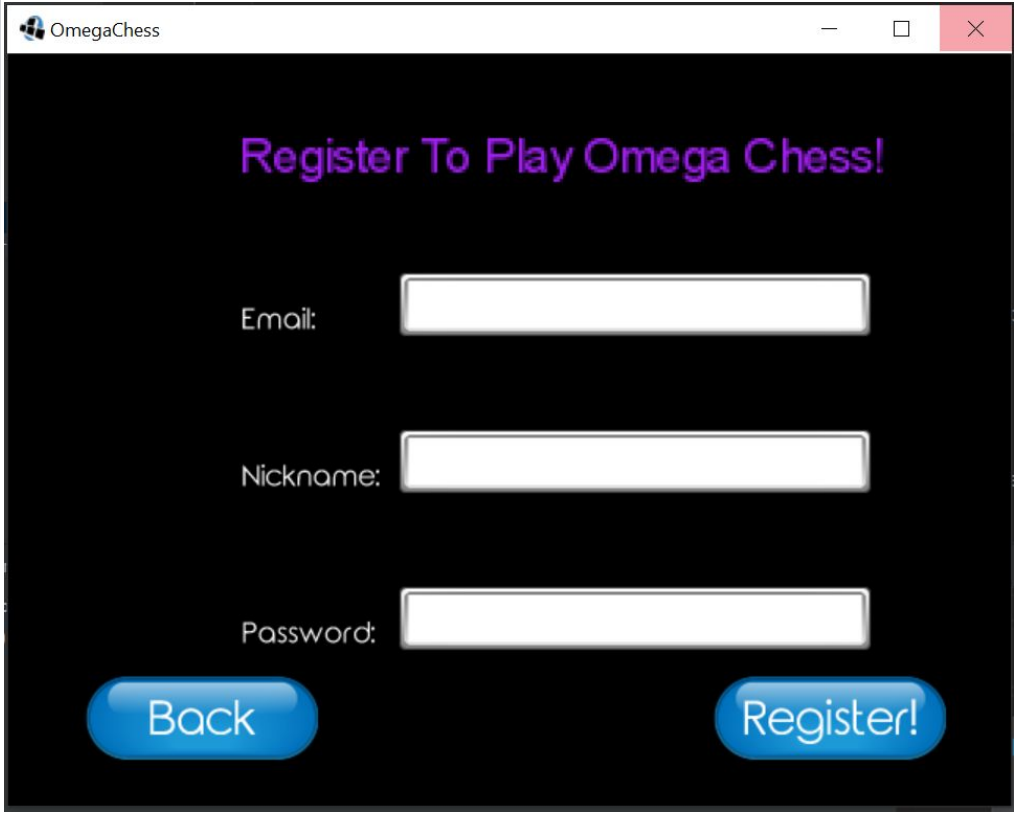
Graphics - Main Menu Screen

Omega Chess Main Menu Screen



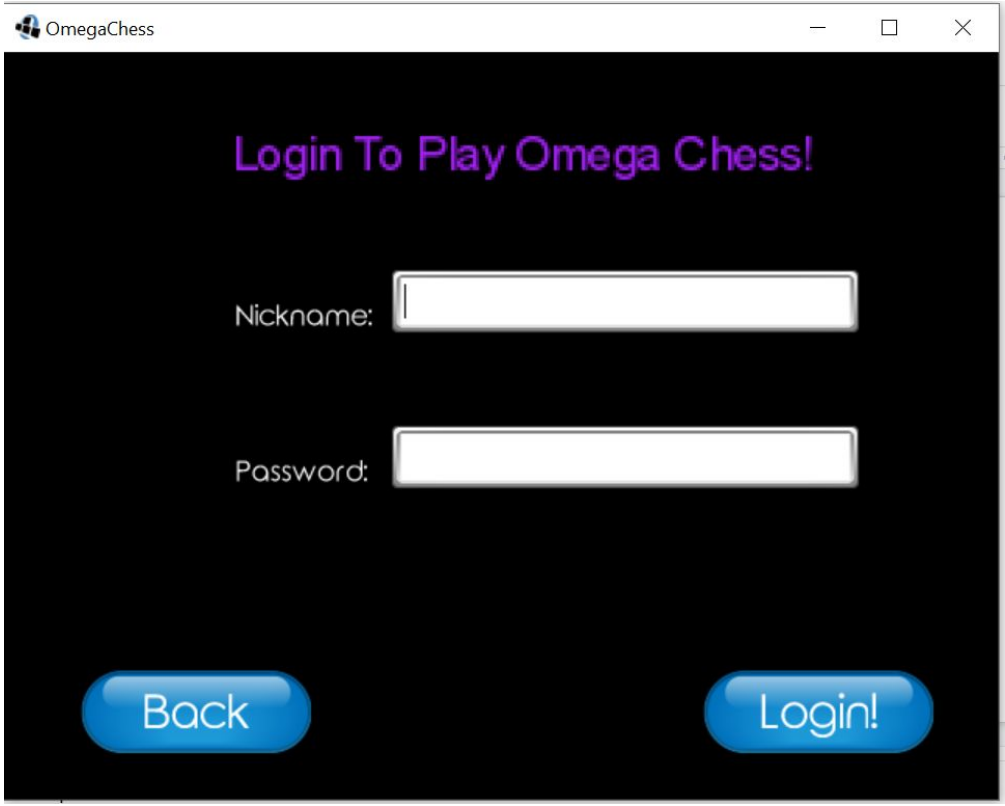
Graphics - Register Screen

Omega Chess Register Screen



Graphics - Login Screen

Omega Chess Login Screen

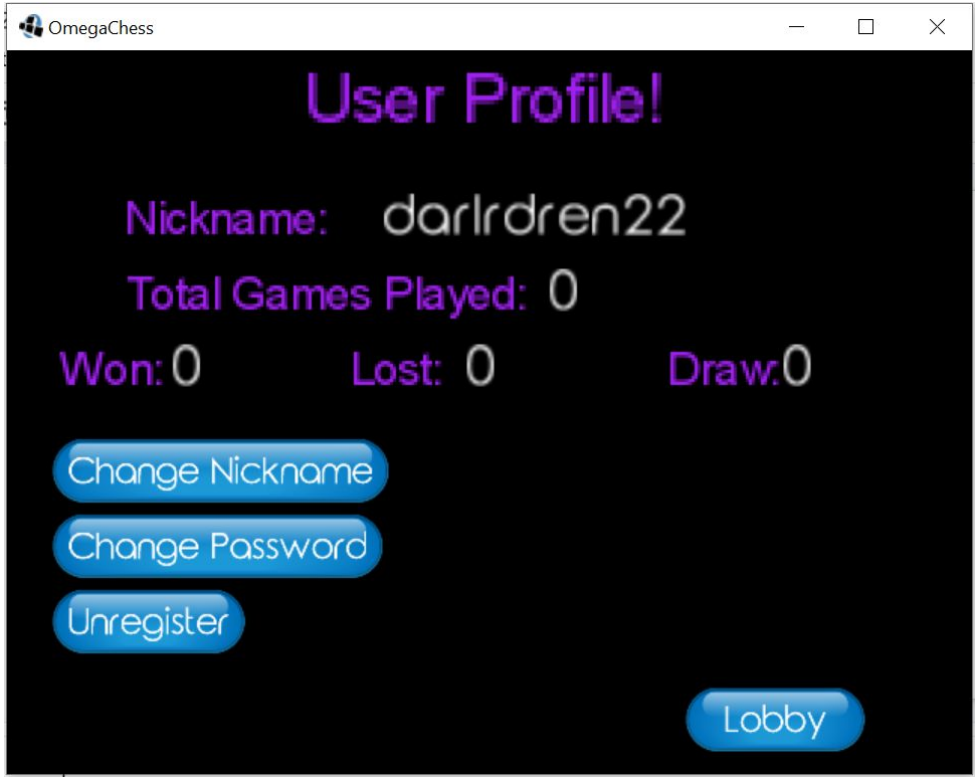


Graphics - Lobby Screen

Omega Chess Lobby Screen

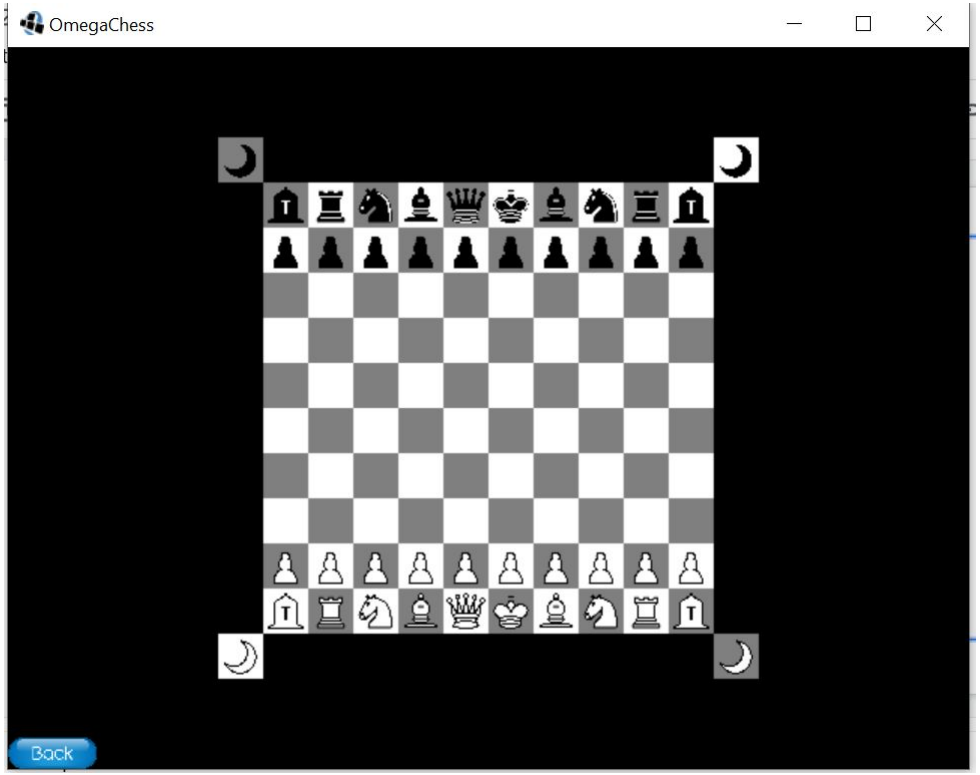
Graphics - Profile Screen

Omega Chess Profile Screen



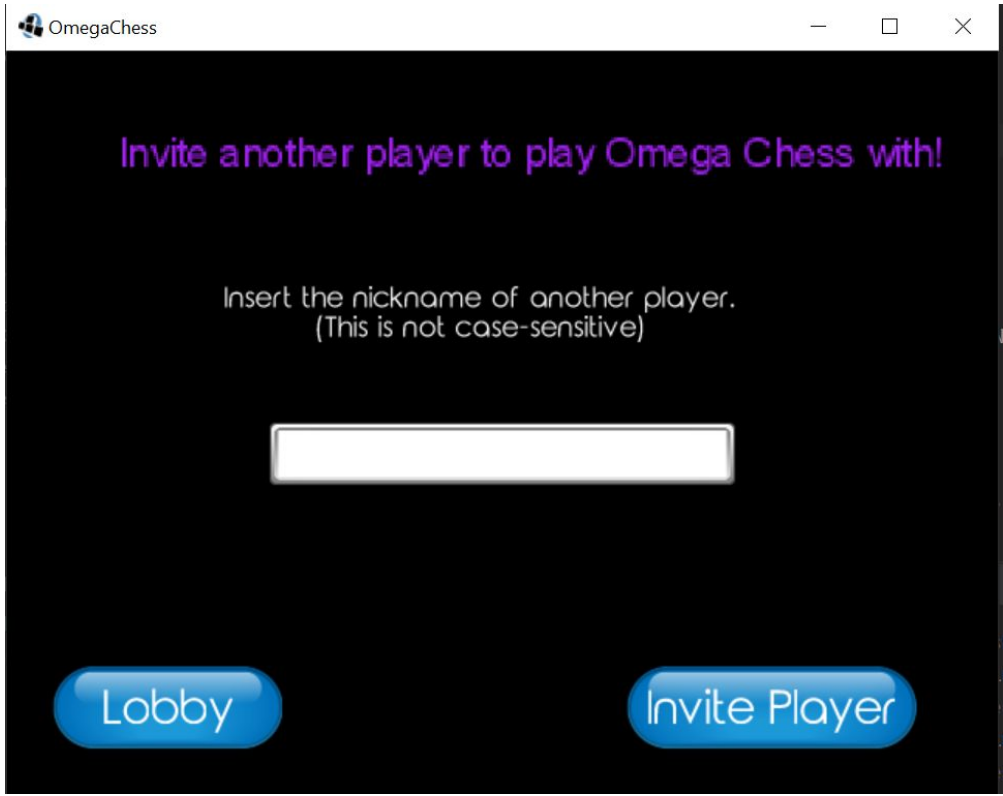
Graphics - Game Board Screen

Omega Chess Game Board Screen



Graphics - Invite Screen

Omega Chess Invite Screen



User Story Improvements

We met with the PO to determine highest priority user stories for each sprint during the P2 timeframe.

We determined acceptance criteria and have been tweaking them as needed.

We have associated each task in our repository with a user story.

Completed User Stories

1) As a user, I can register for the platform by using my email address and set up my profile (nickname, password) for quicker access to the application.

3) As a user, I can invite another user to join the match.

4) As a user, I can send more than one invitation.

8) As a user, I can unregister.

15) As a user, I want to have a lobby which shows me the actions I can perform.

17) As a user, I will be able to view a chess board and pieces when I’m in a match.

User Stories In Progress

2) As a user, I can create a new match.

5) As a user, I can accept or reject an invitation, alerting the inviter.

6) As a user, I can participate in multiple games at the same time.

7) As a user, I can quit from any game at any time.

11) As a user, I can make the first move if I create the match

16) As a user, I will be able to quit the game using a Quit button.

18) As a user, when I choose to move a piece the board will show where I can move the piece.

User Story Acceptance Criteria

(for completed user stories)

1) As a user, I can register for the platform by using my email address and set up my profile (nickname, password) for quicker access to the application.
    a. Test user attempts to login with invalid nickname (fail)
    b. Test user creating new account with email address already registered (fail)
    c. Test user creating new account with email address not already registered (pass)
    d. Test user attempting to login with valid email address (pass)
    e. Test after registering with email address, submitting nickname that already exists (fail under assumption that nickname is unique)
    f. Test after registering with email address, submitting nickname that does not already exist (pass)
    g. Test after registering with email address and nickname, creating password that meets complexity requirements (pass)
    h. Test after registering with email address and nickname, creating password that doesn't meet complexity requirements (fail assuming we have complexity requirements for passwords)
    i. Test logging in with invalid email/password combination (fail)
    j. Test logging in with valid email/password combination (pass)

User Story Acceptance Criteria (continued)

3) As a user, I can invite another user to join the match

   a. Test that a user can invite another to join the match (pass)
   b. Test that it is not possible to create a match with an invalid/blank opponent (pass)

4) As a user, I can send more than one invitation

   a. Test that a user tries to send multiple (>1) invitations without error (pass)

User Story Acceptance Criteria (continued)

8) As a user, I can unregister

a. Test that the user has the option to unregister from the game (pass)
b. Test that when the user hits the unregister button, a warning/confirmation popup comes up asking if the user is sure they want to unregister (pass)
c. Test that hitting cancel on the confirmation popup return the user to their user page and does not do anything to their account (pass)
d. Test that hitting the OK button on the confirmation popup unregisters their account and signs them out of the application (pass)
e. Test that after unregistering, trying to sign in with the previous credentials will result in not being able to find user and prompting to register (pass)

User Story Acceptance Criteria (continued)

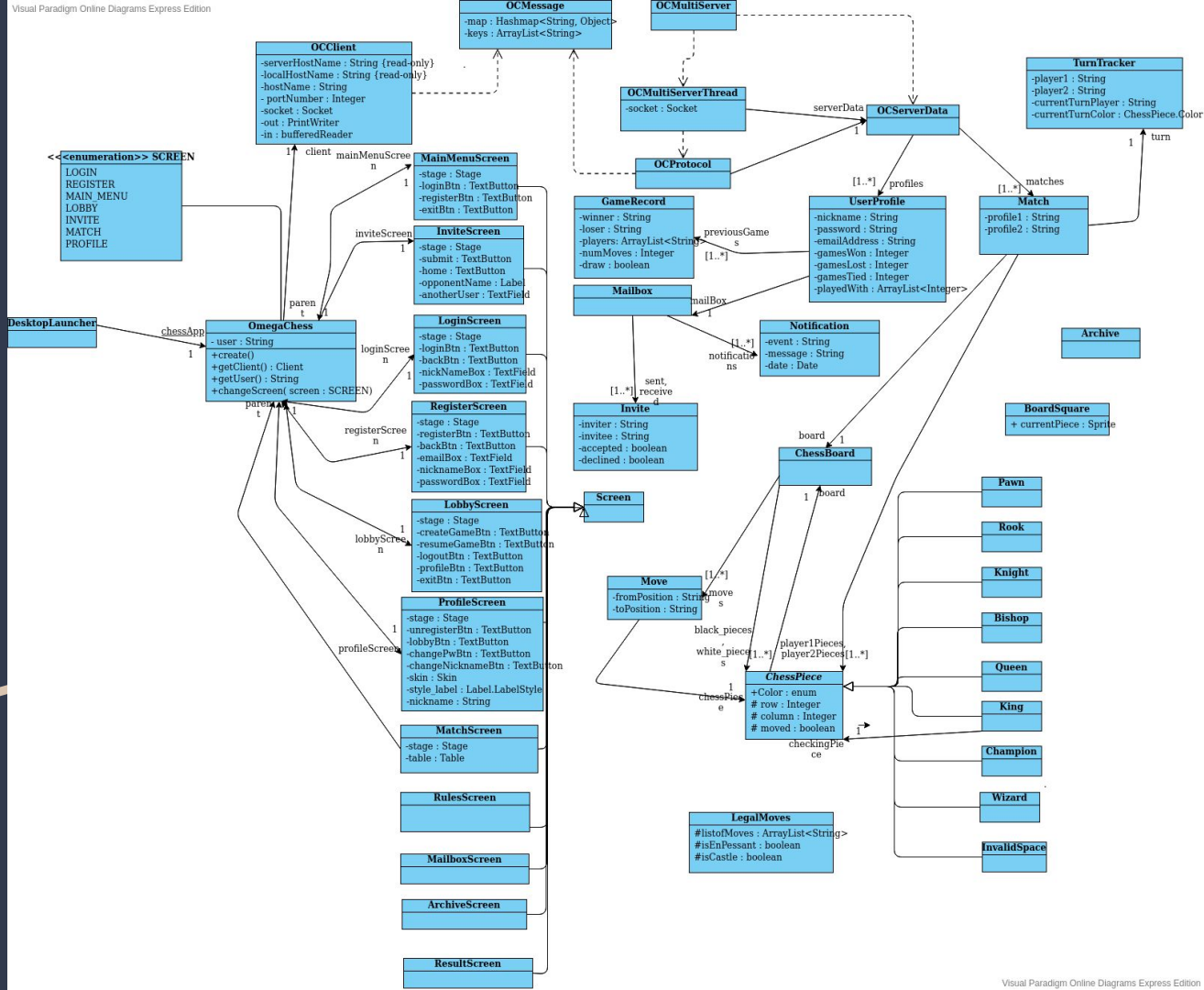
15) As a user, I want to have a lobby which shows me the actions I can perform

a. Test that a lobby screen is present (pass)
b. Test that the profile button takes the user to their profile settings (pass)
c. Test that the continue game button takes the user to show current games

17) As a user, I will be able to view a chess board and pieces when I'm in a match

a. Test that in an active game, a chess board is shown (pass)
b. Test that in an active game, the chess board shows the pieces accurately (pass)

Class Diagram



CRC Cards

OmegaChess	
• Has user String for the current user • Has Enumeration SCREEN • Is the core of our program, it launches the client which processes requests from the different screens and sends them to the server.	

OCClient	
• Has objects serverHostName, localHostName, hostName, portNumber, socket, out, and in	• OmegaChess • OCMessage

OCMessage	
• Has objects map and keys • Converts a request and response to and from a string for Server-Client communication	• OCClient • OCProtocol

OCMultiServer	
• Listens for new connections from clients • If the connection is successful it passes the client to a new OCMultiServerThread	

OCMultiServerThread	
• Uses OCProtocol to process requests from the client	• OCMultiServer

OCProtocol	
• Processes requests sent from the client and returns an OCMessage as a String to the client	• OCMultiServerThread

OCServerData	
• Has objects profiles, matches, and previousGames • Stores variables on the server to allow users to come and go as they please	• OCMultiServer • OCMultiServerThread • OCProtocol

MainMenuScreen	
• Has objects stage, loginBtn, registerBtn, and exitBtn • Choose a screen from login or register	• OmegaChess • OCClient

CRC Cards (continued)

InviteScreen	
• Has objects stage, submit, home, opponentName, anotherUser • Sends invite request to the server from the client	• OmegaChess • OCClient

LoginScreen	
• Has objects stage, loginBtn, backBtn, nickNameBox, passwordBox • Sends login request to the server • Logs the user in and takes them to the lobby screen	• OmegaChess • OCClient

RegisterScreen	
• Has objects stage, registerBtn, backBtn, emailBox, nickNameBox, and passwordBox • Checks password strength • Sends a register request to the server then returns to MainMenuScreen	• OmegaChess • OCClient

RulesScreen	
• Has rules • Shows the rules of Omegachess to the user	• OmegaChess

LobbyScreen	
• Has objects stage, createGameBtnm, resumeGameBtn, logoutBtn, profileBtn, and exitBtn • Gives the player the option to view their profile, send an invite to another user, and resume a match they are in	• OmegaChess

ProfileScreen	
• Has objects, stage, unregisterBtn, lobbyBtn, changePwBtn, changeNicknameBtn, skin, style_label, nickname • Displays the User's information • If you are viewing your own profile option to change nickname and password • View the users win/loss/tie ratio • Sends a request to the server to get the user's data	• OmegaChess • OCClient

MailboxScreen	
• Has objects outBox, inBox, outCount, inCount, • Has a button to show the user their outBox/inBox • When viewing outBox cancel an invite • When viewing inBox accept/decline an invite	• OmegaChess • OCClient

ArchivesScreen	
• Unsure what objects this will have currently • Show previous games that the user has won and lost	• OmegaChess • OCClient

CRC Cards (continued)

ResultScreen	
• Unsure of what objects this will have yet • Show the user the results of a match when it has finished	• OmegaChess • OCClient

BoardSquare	
• Has currentPiece • Contains the sprite of the currentPiece on it • Highlights if it is contained in the legalMoves of a piece	• MatchScreen

GameRecord	
• Has objects winner, loser, players, numMoves, and draw • Creates an object that holds the results of a game • Stores it in the OCServerdata/UserProfile	• OCServerData

UserProfile	
• Has objects nickname, password, emailAddress, gamesWon, gamesLost, gamesTied, playedWith, and mailbox • Stores the data of a user	• OCServerData

Mailbox	
• Has objects sent and received • Holds lists of sent and received invites between other users	• UserProfile

Invite	
• Has objects inviter, invitee, accepted, and declined • Send an invite to a user you deem by their username	• Mailbox • OCMultiServer

Notification	
• Has objects event, message, date • Notifies the user if they have any change in their mailbox • Notify if you have a game request • Notify if someone accepts/declines game invite	• Mailbox • OCMultiServer

CRC Cards (continued)

Match	
• Has objects board, player1Pieces, player2Pieces, and turn • The match is the instance of a game between two players. • I keeps track of the current board state	• OCServerData

TurnTracker	
• Has objects player1, player2, currentTurnPlayer, and currentTurnColor • Keeps track of whose turn it is	• Match

ChessBoard	
• Has objects moves, black_pieces, white_pieces, and board • Allows a piece to move based on its list of legalMoves	• Match • ChessPiece

Abstract **ChessPiece** Pawn, Rook, Knight, Bishop, Queen, King, Champion, Wizard	
• Has objects row, column, and moved • Has Enumeration Color • Returns a list of Moves as a piece's legalMoves	• ChessBoard • Move

LegalMoves	
• Has objects listofMoves, isEnPessant, and isCastle • This is a helper class for holding the legal moves of a piece	• ChessPiece

P2 Sprint 1
9/30 to 10/14

Scrums

Date	Tasks closed	Tasks in progress	Impediments
9/30/20	*N/A*	#19, #105, #106, #107, #108, #109, #130	*N/A*
10/2/20	#19, #130	#20, #21, #48, #84, #106, #107, #108, #109, #112, #113, #129, #131	some problems with getting travis working, but Darla and Falkyn got it working
10/5/20	#20, #21, #48, #84	#33, #105, #106, #107, #109, #112, #113, #128, #129, #131	*none*
10/7/20	#33, #38, #109, #112, #112	#105, #106, #107, #108, #110, #111, #129, #131, #147, #157	some issues with running unit tests with intellij
10/9/20	#105, #106, #107, #108, #110, #111, #129, #131	#70, #71, #147, #157, #160, #162, #163, #166	*none*
10/12/20	#157, #160, #162, #163, #166, #171	#147, #173, #178, #179, #183, #188	*none*
10/14/20	#178, #179, #183, #188	*see sprint2.md*	*none*

Metrics

Statistic	# Planned	# Completed
Epics	*8*	*6*
Tasks	*35*	*29*
Story Points	*53*	*44*

P2 Sprint 2

10/14 to 10/30

Date	Tasks closed	Tasks in progress	Impediments
10/16/20	*#173*	*#37, #95, #98, #99, #147, #187, #200, #209*	*none*
10/19/20	*#98, #182, #200, #205, #206, #207, #208, #209*	*#37, #95, #99, #147, #187, #216*	*Figuring out libGDX and JSON*
10/21/20	*#99*	*#37, #95, #147, #187, #216, #220, #221, #222*	*Still figuring out libGDX*
10/23/20	*#37, #216, #221*	*#95, #101, #147, #187, #222, #226, #227*	*JSON issues, resulting in no JSON*
10/26/20	*#222, #101, #226, #227, #220*	*#46, #50, #66, #86, #95, #147, #187, #229, #238, #241, #242, #243, #244, #245*	*none*
10/28/20	*#238, #146, #144, #148, #95, #94, #93, #92, #241, #242, #229, #210, #66, #49, #46, #244, #245, #50, #243, #86, #60, #54, #40, #57*	*#97, #140, #142, #145, #147, #185, #186, #187*	*none*
10/30/20	*#186, #185, #147, #187, #145, #142, #97*	*#140*	*none*

Metrics

Statistic	# Planned	# Completed
Epics	6	6
Tasks	44	44
Story Points	69	69