

# BoilerChess

## Team 18 - Product Backlog

Brendan Raftery, Eric Lee, Kyle Copenhaver, Nick Zetzel, Sam Kwarteng

### Problem Statement

With this project, we aim to create a web-based version of Chess that will challenge people to develop and harness critical thinking and strategic planning. By matching players against those of similar skill, users can find competitive play easily, or we provide the option for players to face off against their friends. This project will also provide unique programming challenges as we explore the innumerable possible game states and interactions. Not only this, but a logic based game provides many different aspects to practice our software testing skills.

### Background Information

#### Audience

There is always a need for people to expand their critical thinking and improve their problem solving abilities, and chess is an excellent method of doing so.

#### Similar Applications

There does exist many similar versions of this, but our goal is not only to help fulfill this societal need, but also to improve our own skills in testing software. A game of chess provides many clear test cases and since it is a logic game with nearly  $7.7 * 10^{45}$  states, there are more than a few interactions which would need to be tested.

#### Limitations

As for limitations, the most obvious concern is players of higher skill not being challenged enough, and less skilled players being against opponents they can't possibly beat. To solve this, we will implement a skill based Matchmaking Rating (MMR), to track how many wins/losses players get against each other.

### Environment

Our project will primarily be implemented in Javascript, using NodeJS for the back end and a different front-end framework to handle the user's client. As for testing, there are a few automatic test frameworks such as Karma or Protractor which we'll use to automatically test our software. We also plan to use a MySQL database to store information about the users, games, etc.

## Functional Requirements:

Backlog ID	Functional Requirement	Hours	Status
1	As a user, I would like to be able to create a profile which contains my email, username, password, matchmaking rating(MMR), match history, and statistics.	40	Planned for Sprint 1
2	As a user, I would like to be able to recover and/or change my account password by being sent a link by email when I select forgot/change password	14	Planned for Sprint 2
3	As a user, I would like to see my profile which will show the email, username, MMR, match history and statistics.	10	Planned for Sprint 1
4	As a user, I would like to be able to see a global leaderboard listing the top players' usernames, rank, and MMR as well as my rank on the leaderboard.	6	Planned for Sprint 2
5	As a user, I would like to be matched to the opponent that's MMR is closest to my own and within the range of MMR specified by me when playing ranked.	14	Planned for Sprint 1
6	As a user, I would like to have a list of friends that displays whether they are currently online or not and allows me to invite them to play a game.	24	Planned for Sprint 2
7	As a user, I would like to invite someone to play an unranked game by sending an email with a link that invites them to accept or decline the invitation. Neither player is required to log into an account.	40	Planned for Sprint 1
8	As a user, I would like to be able to chat with my opponent in a chat box that can be hidden and reopened.	16	Planned for Sprint 1
9	As a user, I would like to be able to resign or offer a draw.	6	Planned for Sprint 1

10	As a user, I would like the ability to report my opponent for harassment.	26	Planned for Sprint 2
11	As a user, I would like a display showing all the past moves of the current game.	6	Planned for Sprint 1
12	As a user, I would like review all moves of past games by selecting them in the match history.	14	Planned for Sprint 2
13	As a user, I would like squares to be highlighted that are valid moves for the currently selected piece when it is my turn	6	Planned for Sprint 1
14	As an administrator, I would like to be able to ban, remove and edit user accounts.	16	Planned for Sprint 1
15	As an administrator, I would like to be able to view chat history around the time of the reported incident	10	Planned for Sprint 2
16	As an administrator, I would like to be able to see a list of current games with their details	20	Planned for Sprint 1
17	As an administrator, I would like to be able to join and view an ongoing game.	28	Planned for Sprint 2
18	As a user, I would like to see the current population of the site.	6	Planned for Sprint 1

## Non-Functional Requirements:

1. As a user, I would like a simplistic user interface that still gives a good amount of functionality
2. As an administrator, I would like for the website to be easy to maintain
3. As a user, I would like the game to run without lag or framerate issues
4. As a user, I would like the game to not use any of my computer's personal storage
5. As a user, I would like for any personal information to remain secure

## Use Cases:

- Create a profile (1)

1. Site prompts user to create profile	
2. User puts in email, username, password, etc	3. Server check for valid info
4. If necessary, user puts in new information	5. Once valid info is sent by user, server places user in database
	6. User is redirected to home screen

- Log In (1)

1. User navigates to log in page	
2. User puts their information in	3. Server runs check against database for correct user/password
	4. Server prompts user for correct password if necessary
5. User re-enters information	6. Once valid information has been entered, user is navigated to home

- Profile Editing (2)

1. User navigates to profile page	
2. User selects “Edit Profile” button	
3. User enters information they desire to change	4. Server checks for valid information, then prompts user again or change user’s profile in the database

- Recovering an Account (2)

1. User navigates to login page	
2. User selects “Forgot Password” button	3. Server prompts user for username or email address
	4. Server sends temporary password to user and changes their profile in the database to reflect this

- Player Profile Page (3)

1. User navigates to login page and logs in	2. Server validates information and redirects them if it is correct
---	---

3. User navigates to their profile page	4. Server pulls up profile information from their entry in database, including match history, MMR, and friends
---	--

- Global Leaderboards (4, 18)

1. User navigates to “Global Leaderboard” page	2. Server pulls up list of user metrics, such as User Population and Top Ranked Players
--	---

- Finding a game [User logs in, random matchmaking] (5)

1. User selects “Find a game” from home	2. Server checks if user is logged in, then prompts them to do so
3. If not logged in, user enters info or selects	
4. Player selects ranked or unranked	5. Server validates information
	6. Server checks list of waiting players for similar MMR
	7. If none are found, places player in waiting list until one is
	8. As wait time increases, MMR lookup range increases
	9. Once match is found, creates a room for the two, redirects the users, and starts the game

- Inviting User to game (6, 7)

1. User selects “Create a game” from home	2. Server sets up room for user’s game
	3. Server prompts user for opponent’s information
4. User selects user from friendlist or types in a username	5. Server checks if user selected is online, then prompts user if choice is invalid
	6. Once valid choice is made, sends invite to opponent, and waits for response
	7. Once response is accepted, starts the game

- Chatting with an opponent (8)

1. User enters game	2. Server initializes chat client and assigns IDs to all players/spectators
3. User types chat message and presses send	4. Server sanitizes string and then displays it to all users in room

- Hiding a chatbox (8)

1. While in game, user or spectator presses the “-” button in corner of chat box	2. Screen is updated and chat box is changed to a small box with a “+” button to re-engage chat box
3. User presses “+” button	4. Screen is update and chat box is re-opened. All previous messages are available

- Resigning a match (9)

1. While in game, user presses “Resign” button	2. Screen which asks user if they are sure pops up
3. User hits “Yes. I resign”	4. Server updates state of game, and logs the game as a loss for the user

- Offering a draw (9)

1. While in game, user presses “Offer Draw” button	2. Server sends offer to other player, and waits until accepted or rejected
	3. If accepted, game is ended and marked as Draw. Else, game is continued
4. User must wait an additional 10 moves before they may offer a draw again	

- Reporting a user for harassment (10, 15)

1. While in game, user clicks on the offending player’s name and selects “Report Player”	2. Dialog box is presented to the user
3. User selects the reason for reporting the offending player	4. Server sends information (players, chat history, etc) to database to be reviewed by an administrator
	5. Server temporarily mutes the offending player if necessary

- Checking match move history in current game (11)

1. While in game, user selects “View Move History”	2. Server brings up list of moves in the current match by each player on a sidebar
--	--

- Checking Match Move History in Previous Game (12)

1. At the player’s profile, they select the desired match from their match history	2. Server brings up the match summary (time, move history, players, outcome), and displays the information in a pop-up
--	--

- Displaying legal moves for selected piece (13)

1. While in game and during the user’s turn, the user selects a piece	2. Server checks the piece the user has selected and calculates the possible moves for the piece
	3. Server updates the screen to show legal moves

- Modifying User Accounts (14, 15)

1. Administrator logs in and navigates to the “Reports” page	2. Server brings up list of unresolved reports and displays them for the admin
3. Admin clicks on any given report	4. Server shows relevant information to the case
5. Admin decides whether or not to ban player and selects from 2 options: “Dismiss” or “Ban”	6. Server changes status of account to “Banned” if necessary, then throws out the report

- Viewing Current Games (16, 17)

1. User navigates to the “Ongoing Matches” tab	2. Server brings up list of current matches
3. User selects any match	4. Server navigates user to the chosen game and adds them as a “Spectator”