

# Application Logic

Checkers 667

Team D

*Milestone 4*

## **Team Members**

1. Niftiyev Hasan
2. Salgado Rodolfo
3. Singh Sukhjit
4. Liszewski Ryan

## **GitHub Repository Link**

<https://github.com/sfsu-csc-667-fall-2017/term-project-fall-2017-teamD-checkers>

## **Live Application**

<https://checkers667.herokuapp.com>

# Table of Contents

- 1. Resources**
- 2. Table of Contents**
- 3. User Actions**
- 4. System Events**
- 5. API Documentation**

# User Actions

## **Player creates a game room**

**Inputs:** player

**Outcome:** New game room created, with the player assigned to that game room. The game is in a pending state. **Note:** *that this event will only occur from the application lobby.*

## **Player joins a game room**

**Inputs:** player, game room

**Outcome:** Player added to the game room. **Note:** this event will occur from within the application lobby.

## **Player sends a message in game room**

**Inputs:** player, game room, message

**Outcome:** Message broadcast to all players in the room

## **Player makes a move (rotates hourglass)**

**Conditions:** the player triggers timer by making a move

**Outcome:** the time player spends on a move will be added to opponents leftover time, simulating the hourglass of complementing to 3 minutes.

## **Player sends a message in lobby**

**Inputs:** player, message, lobby

**Outcome:** Message broadcast to all players in the lobby

## **Player exits the game room**

**Inputs:** lobby

**Outcome:** Player resigns and his rating gets updated.

# System Events

## **Game room receives all required players**

**Conditions:** Game room required player count is met after a player joins the room

**Outcome:** Random color of piece will be set to each player to initiate first move.

## **Player is given a turn**

**Conditions:** The game state indicates that the player is the current player

**Outcome:** Player is given a relative amount of time define by the opponents time, if no moves are made, player will lose the game by default.

## **Player draws option**

**Conditions:** If a player has made more than 50 moves

**Outcome:** It will allow players to offer each other a draw (option only available after 50 moves)

## **Player runs out of pieces**

**Conditions:** the player has no piece left

**Outcome:** An event has occurred that changes the state of game, and the player will declared as a loser. A pop up window will appear to give user an option to “rematch” or “Return to Lobby”

## **Player runs out of time**

**Conditions:** either white side player reaches to 3 minutes, or black side player reaches to 0.

**Outcome:** player that satisfies the condition above will declare as a loser.

## **Players rating gets updated**

**Conditions:** player either win or lose

**Outcome:** ratings will be updated accordingly.

# Endpoint Overview (API Documentation)

This API is for an online multiplayer checkers game. There are three main pages that users will interact with. Each page has API endpoints that support the functionality of that page.

*Basic description of what each page will do:*

## Login Page

This is the initial page the user will see, the root page. It's purpose is to allow a user to log into the game, or create an account so that they can log into to game. It's primary purpose is to allow the user persistence, so that if they accidentally close the game window they can logon and return to a game in progress. After logging into the system the user will be directed to the lobby page.

## Lobby Page

This is the location where an authenticated user can either select a game to begin, or create a new game. It will list the games that are awaiting an opponent, so that a user can select one and click the join button. If the user clicks on the create new button a new game will be generated. If the user clicks on the "settings" option, it will be redirected to the profile page. If the user clicks on the "create new game" or selects an "existing game", the next page the user will be directed to is the game page and will become a player in the selected game.

## Game Page

This is the location where the user will interact and play the checkers game. There are two interactions that the user will perform. Initially a user will be selected at random to start the game with the first move. Once the user makes its move, it will await for a maximum of 3 minutes for its opponent to move otherwise it will win by default. The user will also have the option to message its opponent on the messaging sidebar. If more than 50 moves are made an option to withdraw will become available to both users.

## Signup Page

HTTP Method	Endpoint	Function
POST	/	<b>Authenticate User (Login)</b> <ul style="list-style-type: none"><li>• get db.hashPassword</li><li>• compare hash.password from user Input</li><li>• db.password if match redirect to /lobby page</li><li>• If no match display error and redirect to / page</li></ul>

POST	/register	<b>Create New User</b> <ul style="list-style-type: none"> <li>• check db for uniqueness username</li> <li>• check db for uniqueness email</li> <li>• hash.password</li> <li>• Add db.addUser</li> </ul>
------	-----------	---

## Lobby Page

HTTP Method	Endpoint	Function
POST	/game/:id/:playerid	<b>Create New Game</b> <ul style="list-style-type: none"> <li>• create db.createGame</li> <li>• get players ID</li> <li>• get players Rank</li> <li>• isActive</li> <li>• random assigned player first move</li> </ul>
POST	/dashboard/:userid	<b>User Information / Edit Setting</b> <ul style="list-style-type: none"> <li>• get player db.Players</li> <li>• select p_rank, p_password, p_username</li> <li>• get player image</li> </ul>

## Game Page

HTTP Method	Endpoint	Function
POST	/game/:id/draw:playerid	Player clicks the DRAW button which will be clickable after move 50 when game is in a dead draw situation that nobody could have an upper hand on each other. Once the opponent accepts the draw game state will be isGameOver.
POST	/game/:id/rematch/:playerid	Player clicks the REMATCH button when the game state reaches isGameOver state, a new game is initiated for both players to play with opposite colors.
POST	/game/:id/lobby/:playerid	Player clicks the LOBBY button, both players return to the lobby page.