

CS451 Group 8 Checkers Web

Application

Test Cases Document

Team: Alex Sladek, Brian Stump, Eric Szabo, Philip Stephenson

Team Number: 8

Instructor: Dr. Filippos Vokolos, Ph. D.

Date Submitted: 3.3.2019

Table of Contents

Table of Contents

1. Introduction

Purpose & Scope of Document

Document Contents

2. Testing Environments

3. Testing Setup Information

4. Test Cases

Test Case Table Legend

Priority Descriptions

Key terms

4.1 Checkers Web Application Main Page

Description

Pre-conditions

Tests

4.2 Finding a Checkers Match

Description

Pre-conditions

Tests

4.3 Checkers Match Initialization

Description

Pre-conditions

Tests

4.4 Checkers Match Gameplay

Description

Pre-conditions

Tests

4.5 Checkers Match Options & Conclusion

Description

Pre-conditions

Tests

5. References

1. Introduction

Purpose & Scope of Document

The purpose of this document is to define testing environments and outline a comprehensive set of test cases for testing the functionalities of the Checkers Web Application. These test cases will reflect the functionalities described in the “CS451 Group 8 Checkers Web Application Software Requirements Specification” document and provide descriptions of how a tester can test that these requirements are met. The scope of this document is to allow anyone with no prior knowledge of the application to test it using the test cases specified in this document. Thus, the tests outlined in this document do not test implementation details of the application but rather the functionalities these implementations facilitate from an end user perspective. This document will thus serve as a guideline for the testing process and will allow for testers to record their testing results.

Document Contents

This document consists of 3 major sections. The “Testing Environment” outlines information of tester systems, the “Testing Setup Information” outlines basic steps that must be undertaken to perform testing and finally the “Test cases” section outlines the different testing scenarios that must be performed by testers. This section in itself is subdivided into the major functional sections of the Checkers Web Application so that testers may choose a particular component and test its elements.

2. Testing Environments

Tester Name	System	Environment	Test Date	Comments	Pass/Fail Status
Alex Sladek	Windows 10	Python 3.7 Chrome v.72	3.10.19	Version 1.0 tested. Gameplay critical tests passing. Some functionalities not implemented and could thus not be tested.	PASS
Eric Szabo	N/A	N/A	N/A	N/A	N/A
Philip Stephenson	N/A	N/A	N/A	N/A	N/A
Brian Stump	Ubuntu 18 / Windows 10	Python 3.7 NodeJS 8.11.1 Firefox 65.0.2 (64-bit) Chrome v.72	N/A	N/A	N/A

3. Testing Setup Information

The following conditions must be met for testing to take place:

1. The Checkers Web Application server must be running and capable of accepting connections over the internet.
2. The tester must have a computer capable of running a web browser
3. The tester must have an internet connection and a modern up to date web browser such as Chrome or Firefox.
4. The tester must have a functioning keyboard and mouse (or other cursor manipulation tool) connected to their computer
5. The tester must connect to the Checkers Web Application server using the appropriate URL for connecting to the server.
6. The tester must be successfully connected to the Checkers Web Application before running any tests

4. Test Cases

Test Case Table Legend

- **TCID** - Test Case Identification Number
- **Req.** - Requirement ID being tested
- **Pr.** - Priority of test case status
- **Description** - Description of test case
- **Execution Steps** - Outlines steps tester must take to test the test case described
- **Expected Result** - Describes what a passing test result should look like
- **Actual Result** - Describes results of executing test case on application

Priority Descriptions

- **High (H)** - Test cases with high priority must pass before any release can be made
- **Medium (M)** - Test cases with medium priority must pass before a final release can be made
- **Low (L)** - Test cases with low priority can pass or fail without affecting release status

Key terms

- **LMB** - Left Mouse Button
- **King & Regular piece** - Types of checkers pieces
- **Move** - The basic movement type of a checkers piece in a checkers match
- **Jump** - A move a checkers piece can make, in which it jumps over an opponent's piece on the board and eliminates it.

4.1 Checkers Web Application Main Page

Description

The test cases in this section will address functionalities of the main page of the Checkers Web Application.

Pre-conditions

The tester has connected to the main page of the Checkers Web Application.

Tests

TC ID	Req.	Pr.	Description	Execution Steps	Expected Result	Actual Result
4.1.1	UI 1.1.1	M	Entering a valid username	<ol style="list-style-type: none">1. Click the textbox titled "Please choose a username"2. Enter a username consisting of alphanumeric characters3. Press the "Play" button	After clicking the textbox, text can be entered into it. When pressing "Play", the user is redirected to the match search screen	PASS: User redirected to Match Search screen after entering a username and pressing Play
4.1.2	UI 1.1.1	M	Entering an invalid username: Invalid symbols in name	<ol style="list-style-type: none">1. Click the textbox titled "Please choose a username"2. Enter an invalid username, such as a string with symbols3. Press the "Play" button	User is notified with a text below the textbox that the username they've entered is invalid. User is not directed to match search page.	PASS: Entering invalid username yields error message. User prompted to enter new name.
4.1.3	UI 1.1.1	M	Not entering a username	<ol style="list-style-type: none">1. Leave the textbox titled "Please choose a username" blank2. Press the "Play" button	User is notified with a text below the textbox that they must enter a username. User is not redirected to match search page.	PASS: Entering invalid username yields error message. User prompted to enter new name.
4.1.4	UI 1.1.2	H	Redirection to match search page	<ol style="list-style-type: none">1. Enter a valid username and press the "Play" button	User is redirected to the match search page	PASS: User redirected to match search page

4.1.5	UI 1.1.1	M	Registration of username	1. Enter a valid username and press the “Play” button	Match search page displays the username entered by the user in the upper left corner	PASS: After entering username “Jack” and being redirected to match search page, “Jack” displayed in upper left corner
-------	-------------	---	--------------------------	---	--	--

4.2 Finding a Checkers Match

Description

This section contains test cases for testing the match finding and match initialization functionalities of the Checkers Web Application

Pre-conditions

The tester has connected to the main page of the Checkers Web Application and has entered a valid username. Tester is on “Match search” page.

Tests

TC ID	Req.	Pr.	Description	Execution Steps	Expected Result	Actual Result
4.2.1	FR 1.1.1 1.1.3 UI 1.3.1	H	Search Match Button operation	1. Click the “Search for Match” button	User is shown popup with text “Searching...”	PASS: Pressing button shows popup “Searching...”
4.2.2	FR 1.1.4 UI 1.3.2	L	Search Match Timer functionality	1. Click the “Search for Match” button 2. Keep the “Searching...” popup open for 60 seconds	“Searching...” popup displays a timer showing the number of seconds that have elapsed since the “Search for Match button was” pressed.	N/A: Timer not implemented
4.2.3	FR 1.1.5 UI 1.3.3	M	Match found options popup	1. Click the “Search for Match button” 2. Have another user also click the “Search for Match button” in another instance of	Both users are displayed a popup with text “Match found” and two buttons, “Accept” and “Decline”	N/A: Match found popup not implemented

				the application		
4.2.4	FR 1.1.6	M	Accepting a match	<ol style="list-style-type: none"> 1. Click the “Search for Match button” 2. Have another user also click the “Search for Match button” in another instance of the application 3. Click the “Accept” button in the “Match found” popup 4. Have the other user also click the “Accept” button in their instance 	After clicking the “Accept” button, both users are directed to the Checkers Match page.	N/A: Match found popup not implemented
4.2.5	FR 1.1.6	M	Declining a match	<ol style="list-style-type: none"> 1. Click the “Search for Match button” 2. Have another user also click the “Search for Match button” in another instance of the application 3. Click the “Decline” button in the “Match found” popup 	The “Match found” popup closes and user is left on the Match Search page. User who pressed “Decline” button is no longer in match search queue.	N/A: Match found popup not implemented
4.2.6	FR 1.1.6	M	Opponent declines match	<ol style="list-style-type: none"> 1. Click the “Search for Match button” 2. Have another user also click the “Search for Match button” in another instance of the application 3. Click the “Accept” button in the “Match found” popup 4. Have the other user also click the “Decline” button in their instance 	The “Match found” popup closes, “Searching...” popup reopens indicating user is still in queue. User who pressed “Decline” is left at Match Search page.	N/A: Match found popup not implemented
4.2.7	UI 1.3.2	M	Closing the “Searching...” popup.	<ol style="list-style-type: none"> 1. Click the “Search for Match button” 2. Click the “X” symbol in the “Searching...” popup 	The “Searching...” popup is closed and the user is no longer in the match search queue. (No “Match found” popups display)	PASS: Clicking the “Cancel Search” button closes the popup and removes user from

						search queue.
4.2.8	FR -	M	Connection interruption during “Match found” popup	<ol style="list-style-type: none"> 1. Click the “Search for Match button” 2. Have another user also click the “Search for Match button” in another instance of the application 3. Have one of the users close their application instance 	User who did not close their application instance is returned to the match search queue and displayed the “Searching...”	N/A: Match found popup not implemented

4.3 Checkers Match Initialization

Description

The test cases in this section are meant to test the initialization of a checkers match instance by the Checkers Web Application.

Pre-conditions

These test cases shall assume that two players have been successfully matched in the Match Search page and have both been redirected to the Match page of the application.

Tests

TC ID	Req.	Pr.	Description	Execution Steps	Expected Result	Actual Result
4.3.1	FR 2.1.1 2.1.2	H	Checkers board initialization	<ol style="list-style-type: none"> 1. Observe the checkers board rendered when redirected to the Match page 	A valid checkers board is rendered	PASS: An accurate representation of a checkers board is rendered once a match is found
4.3.2	FR 6.1.3	H	Correct checkers piece allocation	<ol style="list-style-type: none"> 1. Count the checkers pieces on the board immediately after redirected to Match page 	The board has a total of 24 checkers pieces of two colors, with 12 belonging to one player and the other 12 to the other player.	PASS: Each player has 12 pieces, one player has white pieces and the other red pieces
4.3.3	FR	H	Correct checkers	<ol style="list-style-type: none"> 1. Observe the 	The checkers	PASS: The

	6.1.6		piece placement initialization	locations of checkers pieces on the Match page at match beginning	pieces are in valid starting locations (See checkers rules)	pieces are rendered in the correct locations
4.3.4	FR 6.1.4	H	Correct checkers piece allocation: Piece types	1. Observe types of checkers pieces on the Match page at beginning of match	All the checkers pieces on the board are regular checkers pieces, there are no king checkers pieces on the board	PASS: Every piece on the board is a regular checkers piece. There are no King pieces on the board.
4.3.5	FR 2.1.3 3.1.2	L	Resizing checkers board	1. Enlarge the browser window displaying the checkers board 2. Restrict the size of the window displaying the checkers board	The checkers board and checkers pieces remain proportionally the same size with respect to each other throughout resizing	PASS: The checkers board representation remains proportional as the window is resized.
4.3.6	FR 6.2.1 6.2.2 6.2.3		First turn allocation	1. Observe the turn indicator at the beginning of the checkers match for both players	Turn indicator should display “Your turn” for one player and “Opponent’s turn” for other player	PASS: The turn indicator displays the turns correctly to each player

4.4 Checkers Match Gameplay

Description

The test cases in this section are meant to test the gameplay elements of the Checkers Web Application. These cases will cover gameplay events like moving pieces, jumping over enemy pieces and invalid movement handling.

Pre-conditions

These test cases shall assume that a checkers match has been successfully initiated with two players and is currently running in the tester's browser. The tester is in the match screen of the application.

Tests

TC ID	Req.	Pr.	Description	Execution Steps	Expected Result	Actual Result
4.4.1	FR 4.2.2 4.3.2 4.2.1	H	Valid move: forward non-capturing movement of regular checkers piece	<ol style="list-style-type: none">1. Move cursor to on top of own regular checkers piece2. Hold LMB and drag piece to empty board square 1 square forward and to the right/left of piece's current position.3. Release LMB	The regular checkers piece that was moved moves to the specified position	PASS: Pieces may be moved with the method specified. Regular pieces can be moved forward.
4.4.2	FR 4.2.2 4.3.3 4.2.1	H	Valid move: forward non-capturing movement of king checkers piece	<ol style="list-style-type: none">1. Move cursor on top of own king checkers piece.2. Hold LMB and drag piece to empty board square 1 square forward and to the right/left of piece's current position.3. Release LMB	The king checkers piece that was moved moves to the specified position	PASS: Pieces may be moved with the method specified. King pieces can be moved forward.
4.4.3	FR 4.2.2 4.3.3	H	Valid move: backward non-capturing movement of king checkers	<ol style="list-style-type: none">1. Move cursor on top of own king checkers piece.2. Hold LMB and drag piece to empty	The king checkers piece that was moved moves to the specified position	PASS: Pieces may be moved with the method specified. King pieces can be

			piece	board square 1 square backward and to the left/right of piece's current position. 3. Release LMB.		moved backwards.
4.4.4	FR 4.3.4 4.2.5	H	Invalid move: Moving checkers piece into an occupied board square	1. Move cursor on top of own checkers piece. 2. Hold LMB and drag piece to a board square occupied by another checkers piece 3. Release LMB	The checkers piece does not move to the specified board square and goes back to its original board square	PASS: Popup "invalid move" appears and piece moved back to its original location. User allowed to continue turn.
4.4.5	FR 4.3.2 4.2.5	H	Invalid move: Moving regular checkers piece diagonally forward more than two squares at a time	1. Move cursor on top of own regular checkers piece. 2. Hold LMB and drag piece to empty board square diagonally forward to the left/right of piece's current position and more than 2 squares away. 3. Release LMB	The checkers piece does not move to the specified board square and goes back to its original board square	PASS: Popup "That jump is invalid" appears and piece moved back to its original location. User allowed to continue turn.
4.4.5	FR 4.3.3 4.2.5	H	Invalid move: Moving king checkers piece diagonally forward/backward more than two squares at a time	1. Move cursor on top of own king checkers piece. 2. Hold LMB and drag piece to empty board square diagonally forward or backward to the left/right of piece's current position and more than 2 squares away. 3. Release LMB	The checkers piece does not move to the specified board square and goes back to its original board square	PASS: Popup displays saying "That jump is invalid" and piece moved back to its original location.
4.4.6	FR 4.3.5 4.2.5	H	Invalid move: Moving checkers piece off of the checkers board	1. Move cursor on top of own king checkers piece. 2. Hold LMB and drag piece off of the checkers board	The checkers piece does not move off of the checkers board and goes back to its original board	PASS: Piece does not move from its original location.

				3. Release LMB	square	
4.4.7	FR 6.3.2	H	Invalid action: Moving an opponent's checkers piece	<ol style="list-style-type: none"> 1. Move cursor on top of opponent's checkers piece. 2. Hold LMB and drag piece to any board square 3. Release LMB 	The opponent's checkers piece does not move to the specified board square and returns to its original location	PASS: Application does not allow user to move opponent's pieces, they remain in their original location.
4.4.8	FR 4.3.7 4.3.6 5.1.2	H	Valid jump: Regular checkers piece capturing movement	<ol style="list-style-type: none"> 1. Move cursor on top of own regular checkers piece. 2. Hold LMB and drag piece diagonally forward (left or right) 2 squares from piece's original location. Piece must also pass over an enemy's checkers piece during the move. 3. Release LMB 	Checkers piece moves to the specified location and the opposing players checkers piece is eliminated from the board	PASS: Player's checkers piece jumps to position beyond opponent's piece. Opponent's piece disappears.
4.4.9	FR 4.3.6 4.3.7	H	Valid jump: King checkers piece capturing movement	<ol style="list-style-type: none"> 1. Move cursor on top of own king checkers piece. 2. Hold LMB and drag piece diagonally 2 squares from piece's original location in any direction. Piece must also pass over an enemy's checkers piece during the move. 3. Release LMB 	Checkers piece moves to the specified location and the opposing players checkers piece is eliminated from the board	PASS: Player's checkers piece jumps to position beyond opponent's piece. Opponent's piece disappears. Forwards and backwards jump work as expected.
4.4.10	FR 5.1.1	H	Invalid jump: Jumping over own piece	<ol style="list-style-type: none"> 1. Move cursor on top of own checkers piece. 2. Hold LMB and drag piece diagonally 2 squares from piece's original location in any direction. Piece must also pass over own checkers piece 	Checkers piece does not move to specified location and returns to its original position. Checkers piece that was jumped over is not eliminated from board.	PASS: Popup "That jump is invalid" appears and piece returned back to its original location.

				during the move. 3. Release LMB		
4.4.11	FR 4.2.5 4.3.2	H	Invalid move: Moving checkers piece non-diagonally	1. Move cursor on top of own checkers piece. 2. Hold LMB and drag piece to board square vertically or horizontally 1 or more board squares away from piece's original location 3. Release LMB	Checkers piece does not move to specified location and returns to its original position.	PASS: Popup "That move is invalid displays and piece moved back to its original location. User allowed to continue turn.
4.4.12	FR 6.3.4	H	Invalid turn end condition: Moving checkers piece back to its original location	1. Move cursor on top of own checkers piece. 2. Hold LMB and drag piece to the location where it was originally. 3. Release LMB	Checkers piece remains in the location it was and player's turn does not end	PASS: Popup "That move is invalid" displays and piece does not move. User's turn does not end.
4.4.13	FR 6.3.4	H	Valid turn end condition: Completing a valid movement	1. Move a checkers piece on the board with a valid movement	Player's turn ends and opponent's turn begins	PASS: When a valid move made by user, user's turn ends.
4.4.14	FR 4.3.8	H	Invalid movement: Making a normal move with a checkers piece when a jump is available	1. Have a own checkers piece on the board that can make a jump move 2. Move another checkers piece with a regular move	Checkers piece moved does not move to specified location and returns back to its original location.	PASS: Popup "You have a jump available, you must take that jump" displays and piece returned to its original location. User allowed to continue turn until making the valid jump.
4.4.15	FR 4.3.9	H	Valid movement: Multiple jumps	1. Select a checkers piece that can make a jump 2. Jump with the checkers piece into a board square where another valid jump is possible	The player may and must move the checkers piece for as long as there are valid jumps available. The turn ends once all valid	PASS: User's turn continues as long as the piece lands into a position where a next valid jump is immediately

				3. Repeat for as many jumps as possible	jumps made	available. Once no jumps available, user's turn ends.
4.4.16	FR 4.2.6	H	Invalid movement: Continuing turn after an invalid move	<ol style="list-style-type: none"> 1. Select a checkers piece on the board 2. Make an invalid move with the checkers piece 	The checkers piece does not move and player allowed to continue their turn by making another movement attempt	PASS: Whenever an invalid move made, user allowed to continue turn until making a valid move.
4.4.17	FR 4.3.11	H	Valid movement: Allowing player to choose and then move a checkers piece when there are multiple pieces that can jump	<ol style="list-style-type: none"> 1. Setup a board configuration where the player has multiple pieces that may perform a jump 2. Choose one of the pieces and move it with a valid jump 	Player is allowed to move the selected checkers piece and the players turn ends once they move the piece	PASS: Setting up two identical board conditions where a user has a choice between two jumps shows that user can choose between either one.
4.4.18	FR 4.2.7	H	Valid movement: Reaching last row of board and promoting regular piece to king piece	<ol style="list-style-type: none"> 1. Setup board with a regular checkers piece in a position where it can make a valid move to the last row on the board 2. Move the regular checkers piece using the move 	The regular checkers piece that was moved to the last row is transformed into a king checkers piece	PASS: When either player reaches their opponents first row, the piece is promoted to a king. Signified by a crown on the checker piece. Added movements also functional.
4.4.19	FR 6.3.3	H	Invalid movement: Moving pieces during an opponent's turn	<ol style="list-style-type: none"> 1. Wait for opponent's turn 2. Attempt to move a checkers piece to a board square 	The piece that was moved should return to its original location	PASS: Popup "You cannot make moves during the opponent's turn displays. No pieces can be successfully moved during opponent's turn.
4.4.20	FR	L	Notification:	1. Setup a board	Player is notified	N/A: Not

	5.1.7		Checkers piece elimination	<p>condition where opponent can capture player's piece</p> <p>2. Opponent captures player's piece</p>	that their piece was eliminated with a popup	implemented
4.4.21	FR 4.2.4	L	Notification: Invalid move	1. Select a checkers piece and move it with an invalid move	Player is notified that they have made an invalid move with a popup	PASS: When making invalid move or jump, popup saying invalid move or invalid jump displayed.
4.4.22	FR 4.2.8	L	Notification: Checkers piece has been crowned	1. Move a checkers piece to the last row of the checkers board with a valid movement	Player is notified with a popup that one of their pieces has become a king	N/A: Not implemented
4.4.23	FR 5.1.8 UI 1.4.3	L	Notification: Regular Checkers Piece Counters	1. Simulate game condition where a regular checkers piece is eliminated through kinging or capturing	Player who loses regular checkers piece has their regular piece counter decremented by one	N/A: Not implemented
4.4.24	FR 5.1.9 UI 1.4.3	L	Notification: King Checkers Piece Counters	1. Simulate game condition where a king checkers piece is eliminated through capturing	Player who loses king checkers piece has their king piece counter decremented by one	N/A: Not implemented
4.4.25	UI 1.4.4	L	Notification: Turn counter	1. Complete a valid checkers match turn	Turn count indicator is incremented by one	PASS: Counter increments each time a valid turn has completed. Counter accurately represents turn count.
4.4.26	FR 6.2.1 UI 1.4.5	L	Turn indicator	<p>1. Complete a valid checkers match turn</p> <p>2. Observer turn indicator</p>	When player's turn ends, turn indicator changes from "Your turn" to "Opponent's	PASS: UI displays "It's X's turn" where X is the player's name who turn

					turn” and vice versa for opponent	it is.
--	--	--	--	--	-----------------------------------	--------

4.5 Checkers Match Options & Conclusion

Description

The test cases in this section concern determining whether appropriate match ending conditions occur through gameplay and game options given to players.

Pre-conditions

These test cases shall assume that a checkers match has been successfully initiated with two players and is currently running in the tester’s browser. The tester is in the match screen of the application.

Tests

TC ID	Req.	Pr.	Description	Execution Steps	Expected Result	Actual Result
4.5.1	FR 6.4.1 6.4.2 6.4.3	H	Match ending: Player loses all pieces	<ol style="list-style-type: none"> Create a checkers match where opponent has 1 piece left Capture opponent’s last piece 	After capture, match ends after notifying player that captured last piece that they’ve won and notifying opponent that they’ve lost.	PASS: Game displays “You win” to winner and “You lose” to loser. Match ends successfully.
4.5.2	FR 6.4.4 6.4.3	H	Match ending: Opponent disconnect	<ol style="list-style-type: none"> Have opponent leave the current checkers match by closing their application instance 	After opponent has disconnected, player left notified that they have won the game with “You win!” popup.	PASS: When either player disconnects, player left shown popup saying opponent disconnected and that they win.
4.5.3	FR 6.4.8 UI	H	Match ending: Redirection to main page	<ol style="list-style-type: none"> Create a checkers match where opponent has 1 piece left 	Once the match has ended, player(s) prompted to leave match. Accepting	PASS: When a match ends, user’s are

	1.6.3			2. Capture opponent's last piece	prompt redirects to main page	redirected to match search page.
4.5.4	FR 6.4.5 6.4.6 6.4.7 UI 1.5.2	M	Match draw: Offering a draw	1. Open the match options menu 2. Click "Offer Draw" 3. Opponent accepts/declines draw	If the opponent accepts the draw, both players displayed "Draw" popup". If opponent declines draw, game continues	N/A: Not implemented
4.5.5	UI 1.5.2	M	Match ending: Quitting a match	1. Open the options menu 2. Press the "Quit game" button	Player redirected to main page, opponent displayed "Victory" message	PASS: Redirected to main page, opponent displayed message as specified.
4.5.6	UI 1.5.1	H	Opening the options menu	1. Click the "Options" button in the match screen	Pull down options menu appears to player's screen with "Quit game" and "Request Draw" option buttons	N/A: Not implemented
4.5.7	UI 1.5.2	H	Closing the options menu	1. Open the options menu in the match screen 2. Click the up arrow button at the bottom of the options menu	Options menu is minimized	N/A: Not implemented
4.5.8	FR 6.4.9	L	Match ending condition: Repetition of board positions	1. Play a match such that the same board layout occurs 3 times	Checkers match ends with both players notified that a draw has occurred. The match ends.	PASS: Match ends when repetition of board layouts occurs
4.5.9	FR 6.4.10	L	Match ending condition: In 40 moves no new kings have been created	1. Play a match where in 40 turns, neither player moves a piece to the last row	Checkers match ends with both players notified that a draw has occurred. The match ends.	N/A: Not implemented

4.5.10	FR 6.4.6	M	Draw notification: Accepting a draw	<ol style="list-style-type: none"> 1. Have opposing player send a draw request during a match 2. Click the “Accept” button in the “Accept Draw?” popup 	Checkers match ends with both players notified that a draw has occurred. The match ends.	N/A: Not implemented
4.5.11	FR 6.4.7	M	Draw notification: Declining a draw	<ol style="list-style-type: none"> 1. Have opposing player send a draw request during a match 2. Click the “Decline” button in the “Accept Draw?” popup 	The draw notification window disappears and the match continues as normal	N/A: Not implemented

5. References

Application Functional Requirements information gathered from:

“CS451 Group 8 Checkers Web Application Software Requirements Specification” document written by Alex Sladek, Eric Szabo, Philip Stephenson and Brian Stump, 2.8.2019

Document format adapted from:

“Software Test Case Document - Chess of Champions” written by Akshay Sharma, Jeremy Bennett, Austin Herring and Josh Weinstein. 2.26.2015.