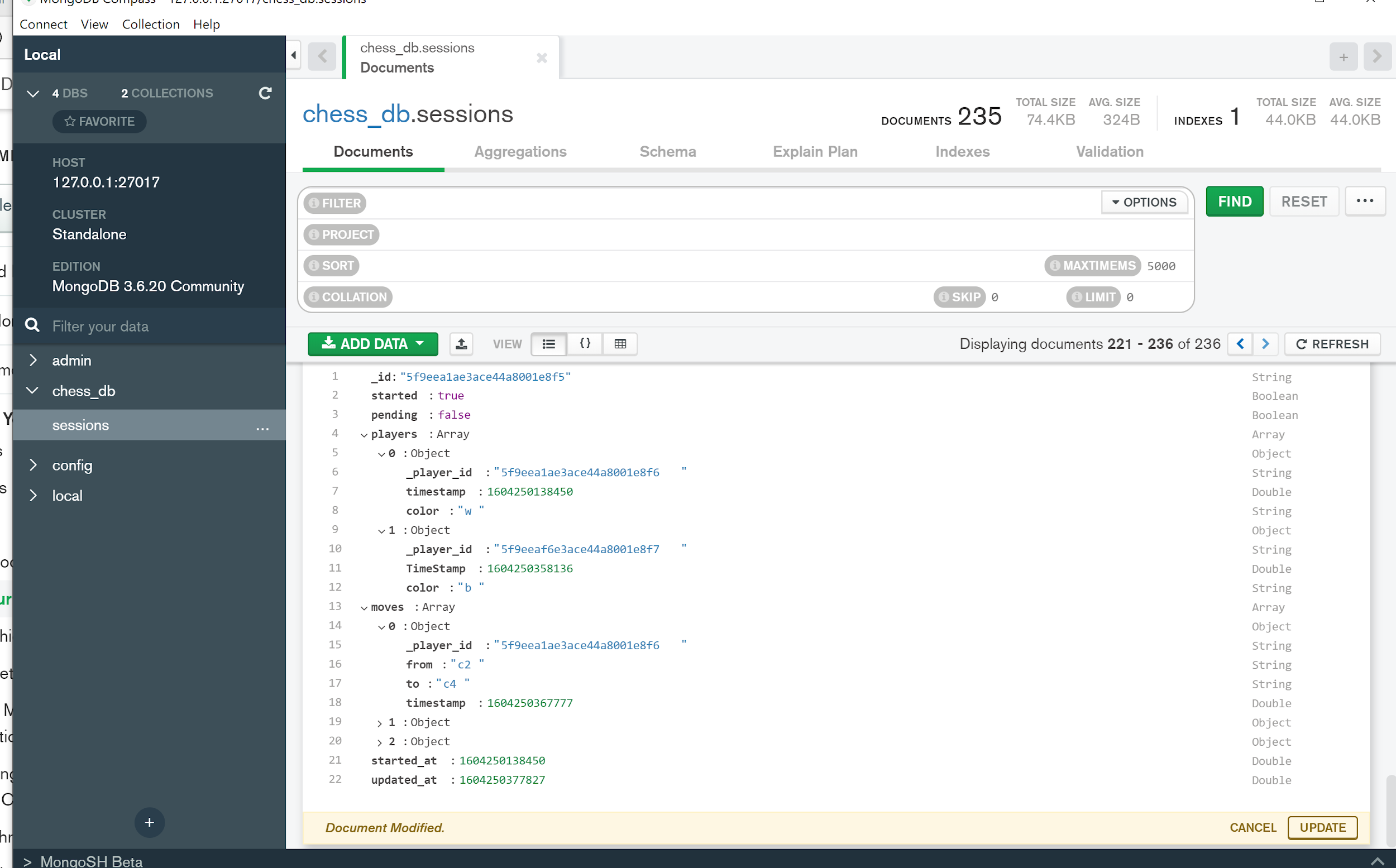
**Tools used during testing:**

* Windows 10 - OS
* Intellij IDE + Vscode
* Mongo DB Compass
* Selenium + Typescript
* Protractor
* Pupeeteer
* Testcafe
* Google Docs for documenting the case observation
* Notepad
* MongoDB Compass

**For my understanding:**

* First of all, I read this document provided in ReadMe.md and understand it carefully.
* Then I set up the environment with command prompt
* I set up env in two different machines

|  |
| --- |
| 1.Machine 1: has port 8080 occupied for another application APPLICATION started on port 8081 :  there should be message for user on which port application should be running 2. Machine 2: APPLICATION started on port 8080 |

* I went through the code base provided and try to understand it
* I set up the environment
* Then I did exploratory testing and functional testing with front end and check with three different tabs and underand the chess board functionality
* I also see the details with database connection
* 

**My Observation:**

1. I tried with selenium - protractor - puppeteer initially
2. **I observe that drag and drop is not supporting very well with all of them for this application**
3. **I also tried Move mouse and hover event - but no success**
4. **But i can also verify later with JAVA or python luggage with selenium**
5. Then I checked with test cafe and it was working with it
6. It was the first time I have worked with test cafe so , please test code structure is very simple.
7. ALso there is no much requirement i see for now to create different component and pages ans it is single page application and done not have many complex functionalities
8. I use the helper class for adding common utilities

**Steps for Happy Path:**

1. Set up the env as mentioned in the Readme.md
2. Start the frontend service and backend connection
3. On browser CHROME Tab 1: open <http://localhost:8080>
4. On browser CHROME Tab 2: open <http://localhost:8080>
5. On browser EDGE Tab 1: open <http://localhost:8080>
6. On step3: Start new Game ( User 1)
7. On step4: Join Game (User 2)
8. User 1 : performs valid move
9. User 2 : Chess board is updated performs valid move

**Observation 1:**

As mentioned in document visitor user can quit the game but it should not affect the session

If visitor user clicks on the Quit Game is closes the live session as well.

**Observation 2:**

I could not write any unit tests for this application.

Because I do not understand a few processes and code base at the moment.

Also it took so much time to verify with selenium and other tools.

**Observation 3:**

Also on page refresh the live session is disconnected . This should not happen

**Observation 4:**

On user turn , User can move pieces to

* Opponent piece location
* EMpty box location
* Just they are not allowed to put piece on own piece location
* There is no chess rule working for this application

**For TEST Plans keep following steps**

1. Analyze the product

* Web application - supported to browsers and mobile browsers
* Contains frontend and backend tests

1. Design the Test Strategy

* UAT tests
* Functional tests
* Browser specific tests
* Performance load tests
* Unit Tests

|  |
| --- |
| I did not get a chance as I could not understand for create unit tests in this project . I will require guidance in first hand then I can create on my own  **Also Some of the functionality cases will be covered during unit tests** |

* System tests

**Functional tests**

1. E2E tests
2. Regression
3. Smoke
4. Sanity
5. Functionality based tests
6. Define the Test Objectives
7. Define Test Criteria
8. Resource Planning
9. Plan Test Environment
10. Schedule & Estimation
11. Determine Test Deliverables

**I have added some of the functionality based scenarios or we can say different modules based on that we can create such flows**

**Funcatinalty bases tests**

Scenario 1: Connection time out behaviour

|  |
| --- |
| Internet connection time out when there is live session   - User 1 turn - User 1 - connection lost for 5 Second and reconnect  - User 1 turn - User 2 - connection lost  - User 2 turn - User 1 - connection lost  - User 2 turn - User 2 - connection lost   - User 2 turn - Server is restarted or server connection is lost  - User 1 turn - Server is restarted or server connection is lost |

Scenario 2: Connection time out behaviour

|  |
| --- |
| Internet connection time out when User has clicked start new Game   - User 1 turn - User 1 - connection lost for 5 second and re connect to system  - A: User 2 clicks on Join the game before 5 seconds  - B: User 2 clicks on Join the game After 5 seconds |

Scenario 3: Session behaviour

|  |
| --- |
| User is already playing in active session  - Verify chess board behaviour on different Browser  - Verify session after closing the tab and re opening the tab on browser  - Verify session after closing the browser and re opening the browser  - Verify session after Navigating to different Tabs on same browser  - Verify session after Refreshing the browser tab  - Verify session on clearing browser cookies in live session  - connecting in multiple devices or browser |

Scenario 4: Browser specific

|  |
| --- |
| - Verify chess board behaviour on different Browser  - Verify chess board behaviour on different OS Device browser  - Verify chess board behaviour on different Browser Resolution   - connecting in multiple devices or browser  Also:   * Verify the functionality and chess board after clicking on Button / moving pieces |

Scenario 5: User Interface

|  |
| --- |
| - Verify system shows correct messages and buttons as per the user activity performed   1. Main User 2. Opponent 3. Visitor   ON JOINING the game :  - Check whether all the pieces of the both sides are available or not and also check whether they are properly organised or not.  ON QUITTING the game:  - Check whether all the pieces of the both sides are rearranged in an old manner or not and also check whether they are properly organised or not.  ON Moving the pieces:  - Check whether all the pieces of the both sides are getting updated and respective message is displayed  ON SYSTEM connection notification :  - Check proper message is displayed on browser to all users   1. 1. System connection issue notifications 2. 2. User turn 3. 3. User specific messages 4. 4. Additional information Instruction message |

Scenario 6: User permission

|  |
| --- |
| - Verify user can access pieces of his own only : When his turn is active  - Verify user can access pieces of his own only : When his turn is NOT active  - Verify visitor user has only view permission and can not edit the board of any user  - Verify visitor user can quit the view  **- Verify on clicking JOIN GAME - button at same time by two different users check the system behavior** |

Scenario 7: Performance / Load test

|  |
| --- |
| - It is a web application  - When it will be live with different user session  → Check the performance test JMETER / New Relic agent or OTHER tool |

Scenario 8: GeoLocation test

|  |
| --- |
| - Verify when the users are interacting from two different geographic location and observe the system behavior  - Also observe the database entry |

Scenario 9: Database

|  |
| --- |
| 1. Verify in database stores and updates all details related to   - user session history  - player move details  - make sure all details are correct  There can be many cases as per the flow |

Scenario 10: Functional Test

|  |
| --- |
| - Visiter user clicks on Quit game - Main user clicks on Quit game  - Opponent user clicks on Quit game  Main user Refresh the Live session  Opponent user : Refresh the live session  Verify when the MainUser user quits the session |

**CASTING**

|  |
| --- |
| Verify casting: E1 to A1 Verify casting: E1 to H1 Verify casting: E8 to H8 Verify casting: E8 to H8 Verify casting: After A1 is already moved once Verify casting: After H1 is already moved once Verify casting: After E1 is already moved once Verify casting: After A8 is already moved once  Verify casting: After H8 is already moved once  Verify casting: After E8 is already moved once  Verify casting: After E8 is has been checked once  Verify casting: After E1 is has been checked once  1. The castling must be kingside or queenside.  2. Neither the king nor the chosen rook has previously moved.  3. There are no pieces between the king and the chosen rook.  4. The king is not currently in check.  5. The king does not pass through a square that is attacked by an enemy piece.  6. The king does not end up in check. (True of any legal move.) |

**Player moves**

|  |
| --- |
| **Verify player can not move pieces in wrong place Verify player can not move opponent pieces** |

Pieces Direction

|  |
| --- |
| Verify valid move direction : Knight Verify valid move direction : Rook  Verify valid move direction : Bishop  Verify valid move direction : King  Verify valid move direction : Queen  Verify valid move direction : Pawn |

|  |
| --- |
| Verify valid move direction : Pawn  - Single move forward  - double move forward  - move forward is Stopped as there is piece in front of it  - move cross  - Also at the end of black line - PAWN can create the valid (QUEEN-ROOK-Bishop-Knight) |

Check - Check Mate cases

|  |
| --- |
| White to black: Verify move : check Black to White: Verify move : check Black to White: Verify move : check mate White to black: Verify move : check mate |

Quit Game

|  |
| --- |
| White/Black player quits game: on his turn White/Black player quits game: Not on his turn |

**Website DESIGN**

|  |
| --- |
| APART from this there will be other cases related   * LOGO * Website design * Tool tip * Page headers * Messages * User look and feel * Spelling mistakes |
|  |

Scenario 11: System test

|  |
| --- |
| Once all the system is working fine .   * Check in the beta version and test it * Also make pair testing session * in HOUSE Testing * Friends and Family session |