

CSE3001 – Software Engineering
Slot – L11 + L12



VIT[®]

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

CHECKMATE

PROJECT REPORT

16BCE0640
16BCE0785
16BCE2148

NIRUT GUPTA
LAKSHAY ARORA
PARTH JAWA

COURSE INSTRUCTOR
PROF: SHASHANK MOULI SATAPATHY

Software Requirements Specification

for

CHECKMATE

Version 1.0

TABLE OF CONTENTS:

1. Introduction
2. Purpose
3. Product Scope
4. Intended Audience and users
5. Functional Requirements
 - 5.1 Pawn
 - 5.2 Rook
 - 5.3 Knight
 - 5.4 Bishop
 - 5.5 Queen
 - 5.6 King
 - 5.7 Castling
 - 5.8 General capture
 - 5.9 Pawn capture
 - 5.10 En passant
 - 5.11 Promotion
 - 5.12 Legality
6. Non-functional requirements
 - 6.1 Ability
 - 6.1.1 Establishing
 - 6.1.2 Protocol
 - 6.2 Scalability
7. Process Model
 - 7.1 Characteristics
 - 7.2 Advantages
 - 7.3 Tasks Involved
8. Work Breakdown Structure
9. Activity Network
 - 9.1 User interface
 - 9.1.1 Expected input
 - 9.1.2 Expected output
10. Critical path identification
11. Gantt Chart
12. ER Diagram
 - 12.1 Individual classes of the system
 - 12.1.1 User
 - 12.1.2 Admin
 - 12.1.3 Outsider
13. Data flow diagram
 - 13.1 Level 0
 - 13.2 Level 1
14. DFD data dictionary

1. INTRODUCTION

- This chess application is web based, which enables users to play chess against any other users who are logged on to the site.
- The only requirement is to have a Java enabled browser and access to the Internet.
- Since it is fully Internet based, no special chess software is required to be able to play chess.
- This application enables players to play chess against other users who are logged in.
- It features a user-friendly interface and GUI design. Hence, even beginners can learn how to connect and play at Chess.

below are various modules of the web application:

A. Homepage

- The home page contains an option to login into the application with brief description about chess.
- Here the user chooses the color he wants to play.
- The user clicks on one of them to login.

B. Login

- After choosing to login the user enters a page with an image of chess board
- asking for his username and password and click on the password button
- you will be successfully logged in after clicking the button provided you gave the correct details

C. Chess Board Design

- The basic layout for the chess board where the player can place their pieces based on the specific rules.
- a chessboard was represented as an ASCII 8x8 grid

D. Chess Board Interface

- after a move is defined
- This show how each player can move following the predefined rules

E. Dashboard

- This allows the user to get details about previous matches and names of all players who are online.

F. Connecting the Player

- We establish the connection between two players after giving the port address

G. Working Behind the Scenes

- We can see the moves being played between the player on the terminal

2. PURPOSE

The purpose of this application is to allow users to play chess easily without installing any heavy application in their system. The algorithms used in this application are also optimized to ensure smooth flow of the game. Existing chess application have a drawback that it takes them a few second to make a move because all of them are on http connection but instead here we make use of sockets so time taken for a move is almost real time.

3. PRODUCT SCOPE

The scope of this project is to make this website a base for chess players. This project can also be extended in various ways. One of them is conduct tournaments in the site. Other features which can be added are creating a discussion forum within the site. Also, ratings can be given to players based on their wins and other factors. Tutorials can be provided to beginners by professionals. It can be one of the most interactive site where many people, keen on playing chess with others in the world, visit.

4. INTENDED AUDIENCE AND USERS

The main audience of this application is chess players. They will get a very good platform to showcase their skills. The second type of audience which can be attracted is people who want to learn how to play chess from a very basic level to a professional level.

5. FUNCTIONAL REQUIREMENTS

- 5.1 Pawn** Pawns shall move one space forward, optionally two spaces forward on their opening move.
- 5.2 Rook** Rooks shall move vertically or horizontally any number of spaces unless impeded by another piece.
- 5.3 Knight** Knights shall move two spaces either vertically or horizontally followed by one space perpendicularly.
- 5.4 Bishop** Bishops shall move diagonally any number of spaces unless impeded by another piece.
- 5.5 Queen** Queens shall move vertically, horizontally, or diagonally any number of spaces unless impeded by another piece.
- 5.6 King** Kings shall move one space in any direction.
- 5.7 Castling** When requirements are met for castling (see definition), kings may move two spaces towards a rook, with the rook moving onto the space crossed over by the king.
- 5.8 General Capture** If a piece other than a pawn, moving in its normal fashion, may move into a square occupied by an opposing piece, the friendly piece may capture the opposing piece.
- 5.9 Pawn** Pawns shall capture by moving forward one space diagonally into an opposing piece.
- 5.10 En Passant** When requirements are met for en passant capture, a pawn may capture as above into a space crossed, but no longer occupied by an opposing piece.
- 5.11 Promotion** A pawn, having entered the rank opposite where it started, shall be promoted to a piece of its controller's choosing.
- 5.12 Legality** A move shall be deemed illegal if it does not follow the above rules or would cause the moving player's king to become in check.

Other functional requirements:

A player shall be able to save his game to a .sav file.

The positions of each player's pieces Whose turn it is

The most recent move made

6. NON-FUNCTIONAL REQUIREMENTS

6.1 Ability

6.1.1 Establishing Connection shall be between two computers, each with the Chess program.

6.1.2 Protocol Messages shall be passed using algebraic chess notation.

6.2 Scalability

Scalability is the capacity to be changed in size or scale. The project is scalable to large extent and can accommodate a large number of users and multiple games.

7. PROCESS MODEL

We decided to use **Incremental build model** for our project.

The incremental build model is a method of software development where the product is designed, implemented and tested incrementally until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements. This model combines the elements of the waterfall model with the iterative philosophy of prototyping.

7.1 Characteristics:

1. System is broken down into many mini development projects.
2. Partial systems are built to produce the final system.
3. First tackled highest priority requirements.
4. The requirement of a portion is frozen once the incremented portion is developed.

7.2 Advantages:

1. It is generally easier to test and debug than other methods of software development because relatively smaller changes are made during each iteration. This allows for more targeted and rigorous testing of each element within the overall product.

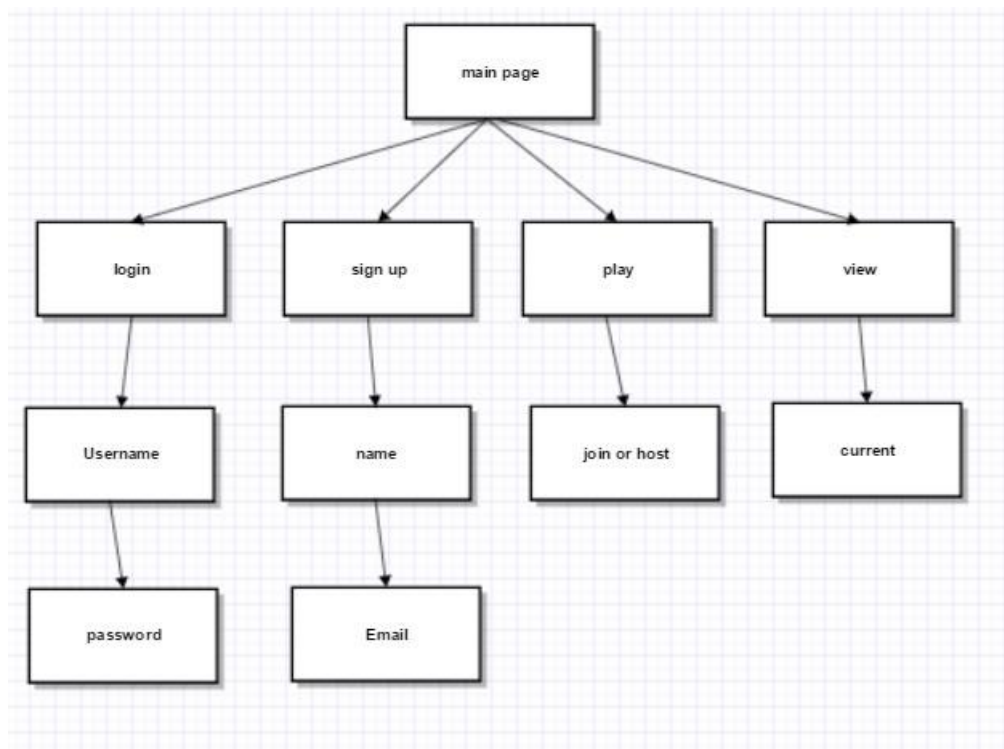
2. Customer can respond to features and review the product for any needed or useful changes.
3. Initial product delivery is faster and costs less.

After considering all the advantages of Incremental build model, we thought of using this process model in our project.

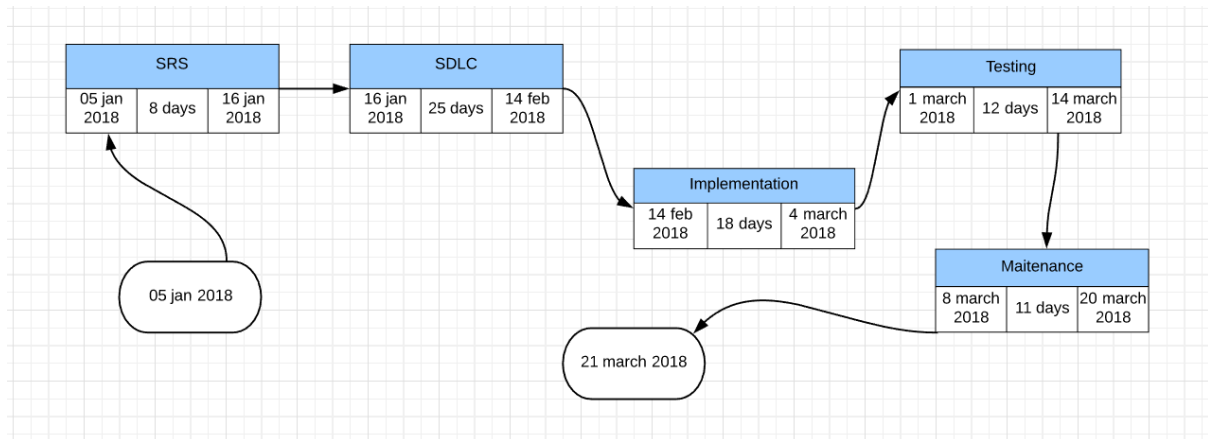
7.3 Tasks Involved

1. Communication: helps to understand the objective.
2. Planning: required as many people (software teams) work on the same project but different function at same time.
3. Modeling: involves business modeling, data modeling, and process modeling.
4. Construction: this involves the reuse software components and automatic code.
5. Deployment: integration of all the increments.

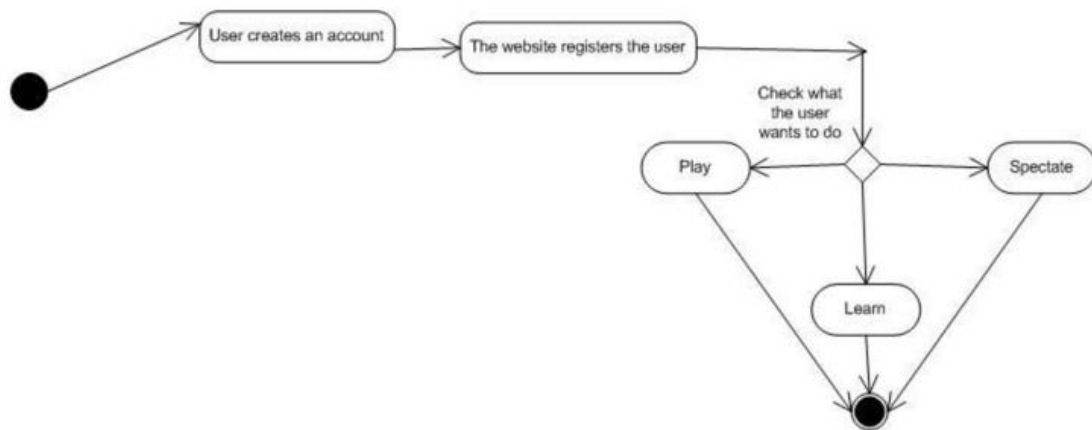
8. WORK BREAKDOWN STRUCTURE



9. ACTIVITY NETWORK



Slack time is zero since the critical path is same as the activity network.



9.1 User interface:

The users that interact with the system via database for data updation or upload purposes shall interact with the help of an input and output screen where the user will be asked several fields.

9.1.1 Expected Input

For user Registration

- Name
- Email ID
- Password
- Date of birth

- Phone number
- Username

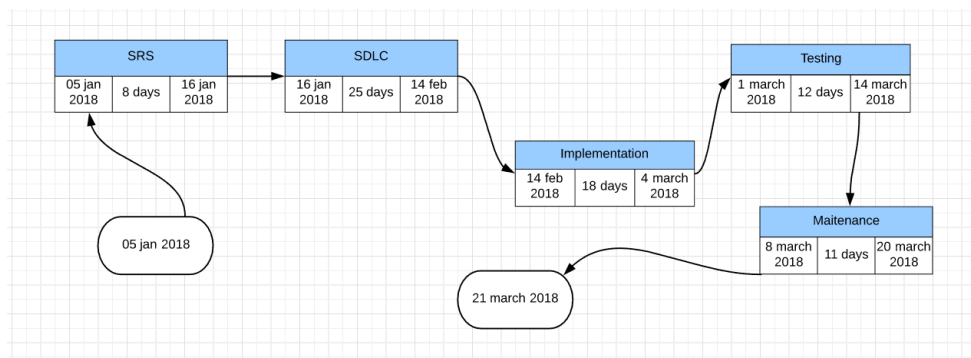
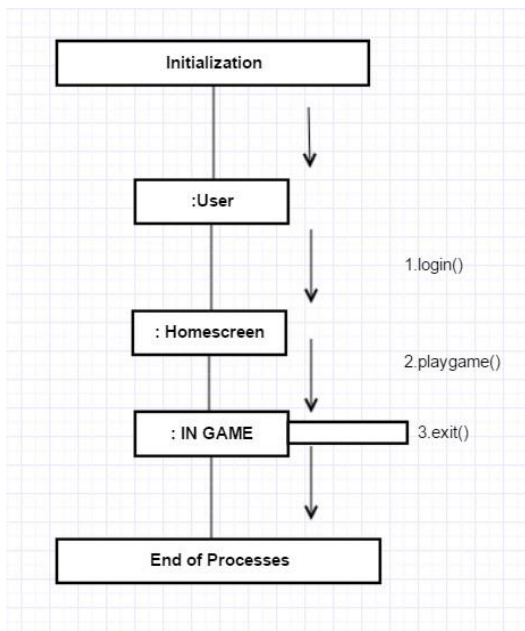
For Login

- Username
- Password

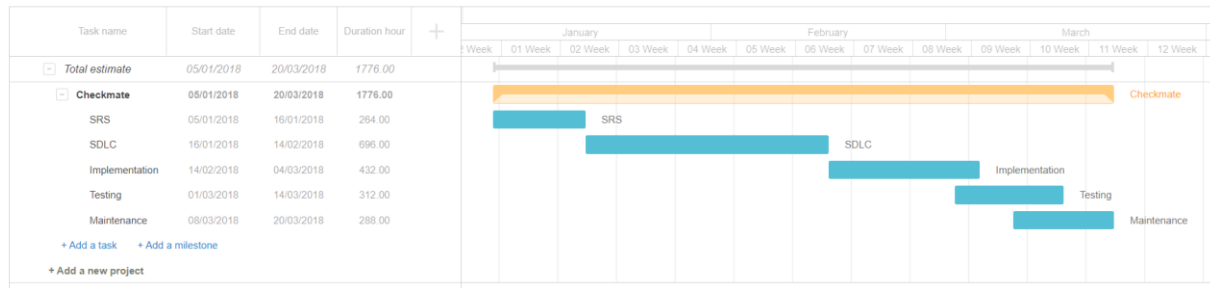
9.1.2 Expected Output

The user will be able to see the next screen if the initials entered matches with the database, if not the user will get a pop up message to retry and get a help screen in case of registration and using the application or forgot password option in case of login error.

10. CRITICAL PATH IDENTIFICATION



11. GANTT CHART



TASK	OPTIMISTIC TIME	MOST LIKELY TIME	PESSIMISTIC TIME
SRS	7	8	9
SDLC	24	25	27
IMPLEMENTATION	16	18	20
TESTING	11	12	14
MAINTENANCE	10	11	12

Expected time of completion

$$= (\text{optimistic} + (4 * \text{most likely}) + \text{pessimistic})/6$$

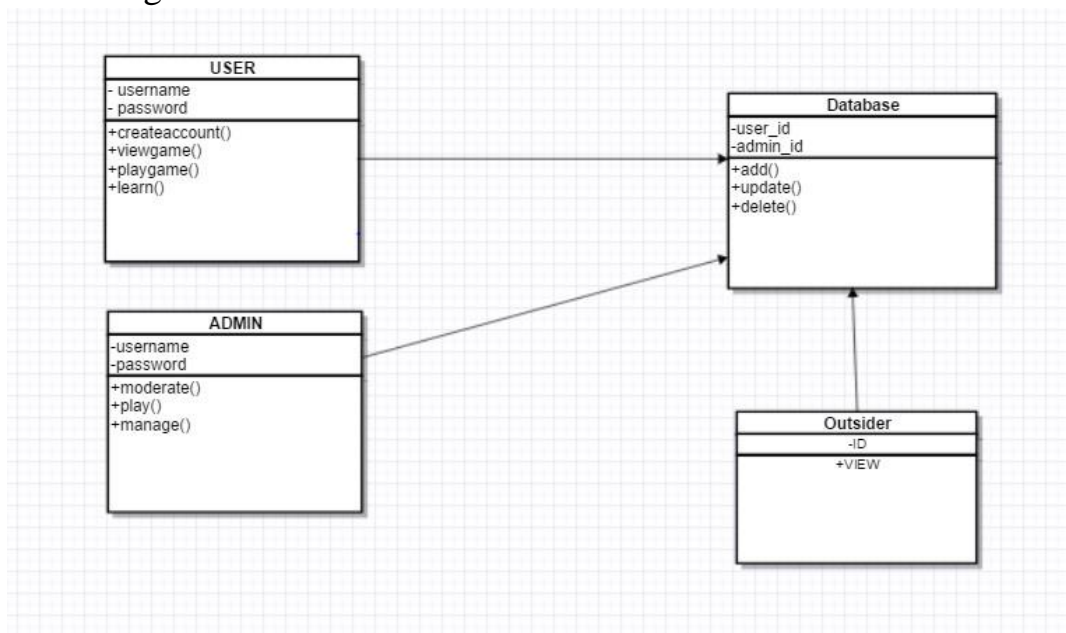
$$= (68 + (4*74) + 82)/6$$

$$= 74 \text{ days approximately}$$

12.ER DIAGRAM



Class diagram



12.1 Individual Classes of the System

12.1.1 USER

- User can create an account on the website and fill up the necessary details required
- Users can see the on-going games and can challenge any online player free to play a game of chess
- User can learn how to play a game of chess by playing an already set up game

12.1.2 ADMIN

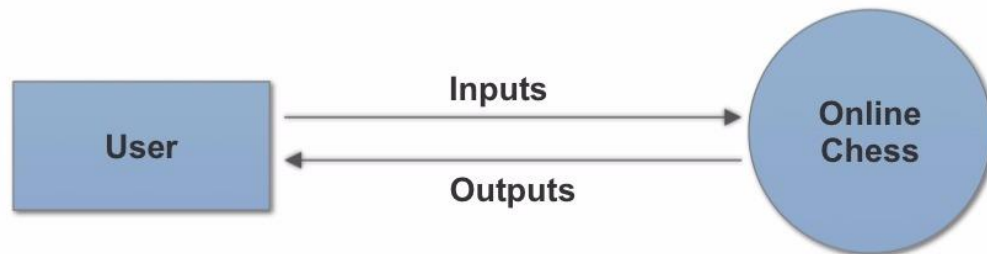
- Admin can moderate a game and see whether the website is functioning properly
- Admin can resolve issues posted by the user in the forum section of the site.
- Admin should be available at all times for the user, in cases where a unbiased opinion and verdict is required

12.1.3 OUTSIDER

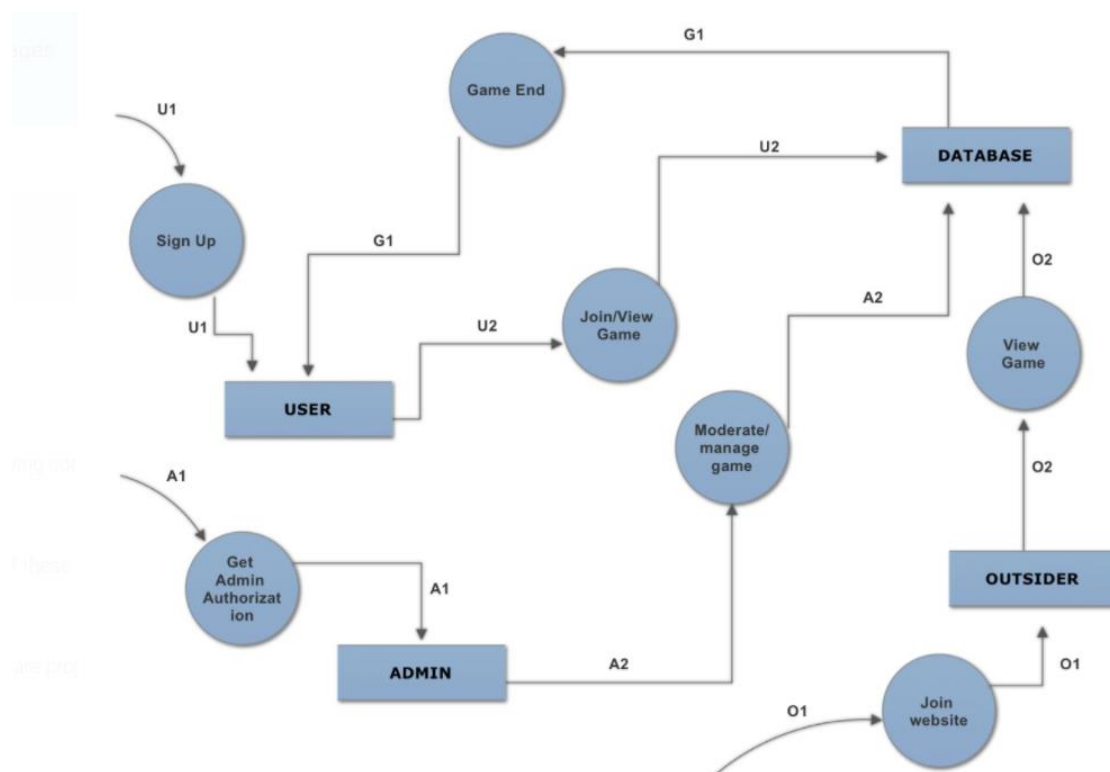
- An outsider can view the website as a whole but can't play a game of chess until he makes an account
- Outsider can see the ongoing matches and can apply to the admin to spectate an ongoing match

13. DATA FLOW DIAGRAM

Level 0



Level 1



14. DATA DICTIONARY

A1 – Information of Admin

(Username, Password, Name, Date of Birth, Phone Number, Email)

A2 – Information of Admin

(Username)

U1 – Information of User

(Username, Password, Name, Date of Birth, Phone Number, Email)

U2 – Information of User

(Username)

O1 – Information of Outsider

(ID, Name)

O2 – Information of Outsider

(ID)

G1 – Information about Game

(Win/Loss)

Software Design Specification

for

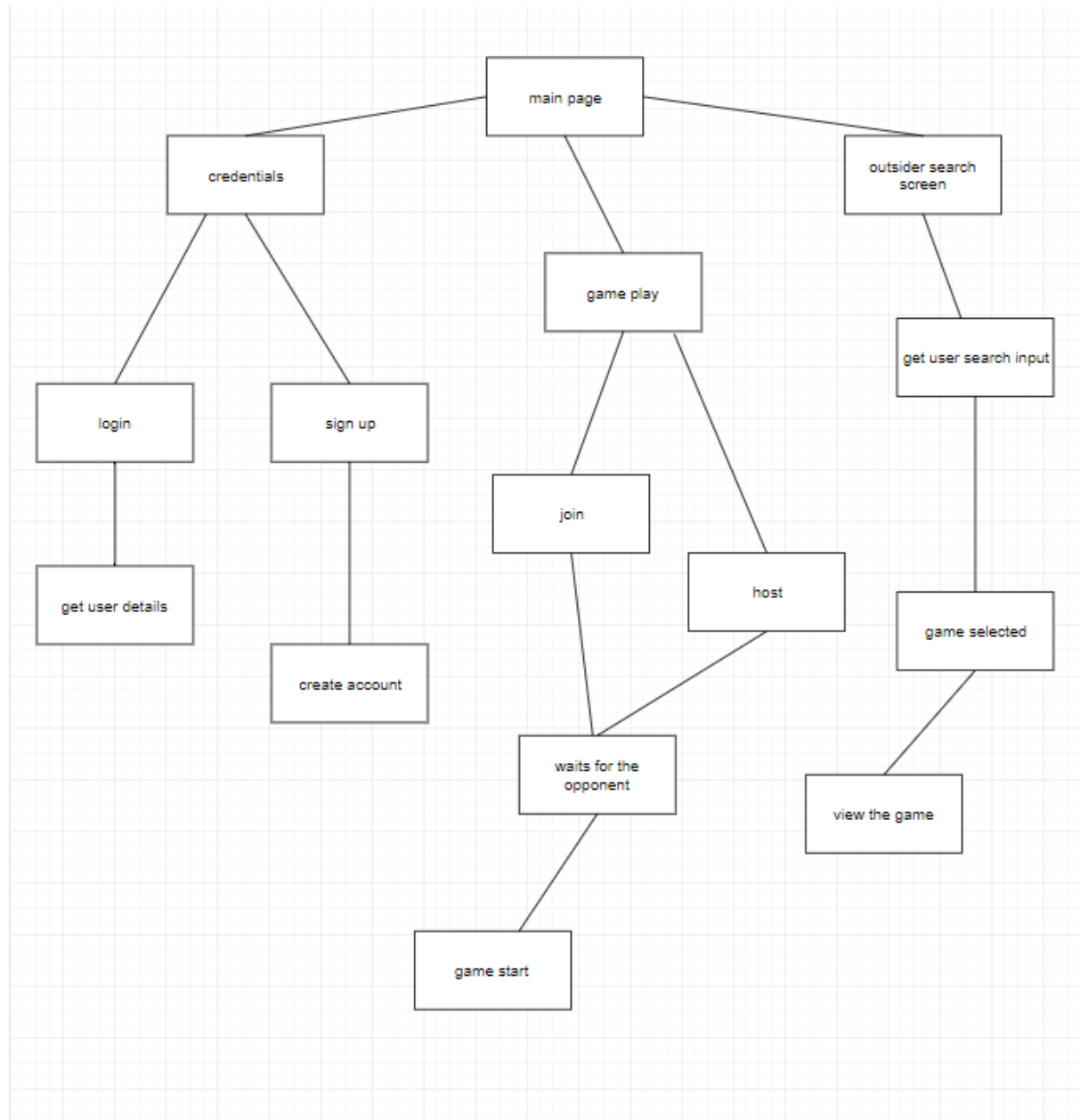
CHECKMATE

Version 1.0

TABLE OF CONTENTS:

1. Structure chart
2. Figures
 - 2.1 Use Case Diagram
 - 2.1.1 Use Case 1
 - 2.1.1.1 Create Account
 - 2.1.1.2 Spectate Game
 - 2.1.2 Use Case 2
 - 2.1.2.1 Join Game
 - 2.2 Activity Diagram
 - 2.3 Class Diagram
 - 2.3.1 CRC Card
 - 2.3.1.1 User
 - 2.3.1.2 Admin
 - 2.3.1.3 Outsider
 - 2.4 State Chart Diagram
 - 2.5 Sequence Diagram
 - 2.6 Collaboration Diagram
 - 2.7 Component Diagram
 - 2.8 Deployment Diagram

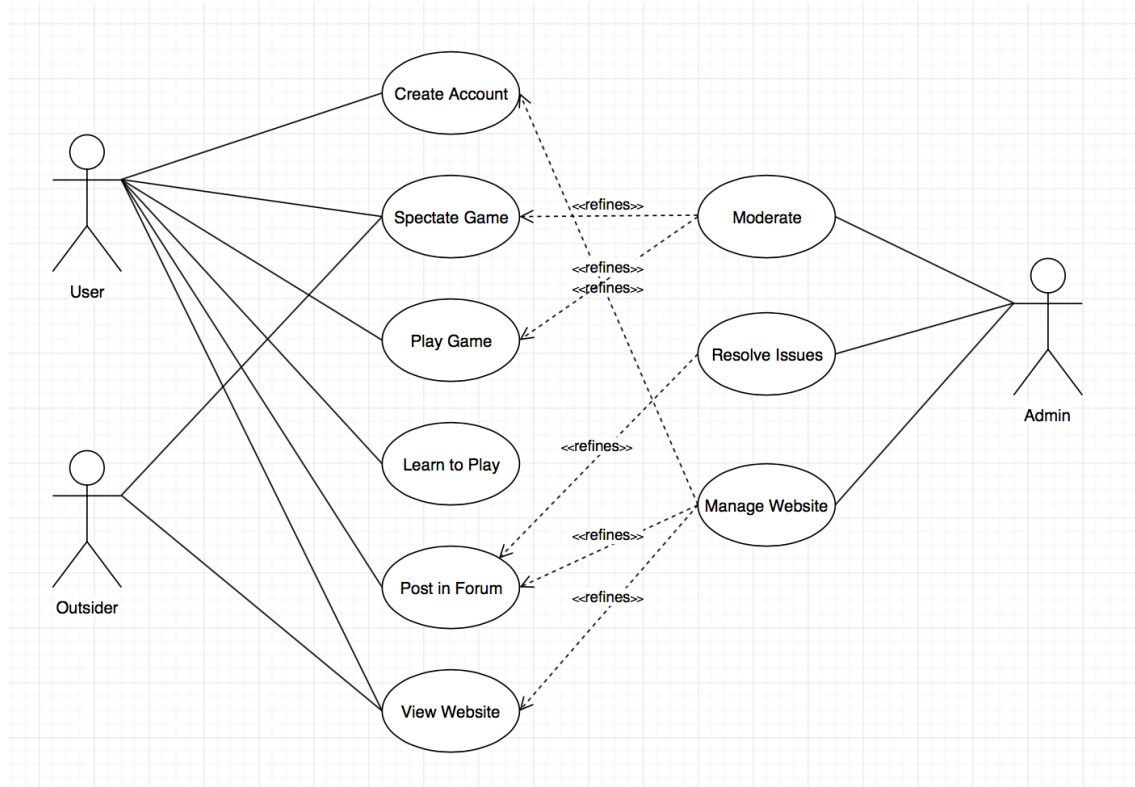
1. STRUCTURE CHART



2. FIGURES

2.1 USE CASE DIAGRAM

2.1.1 USE CASE 1



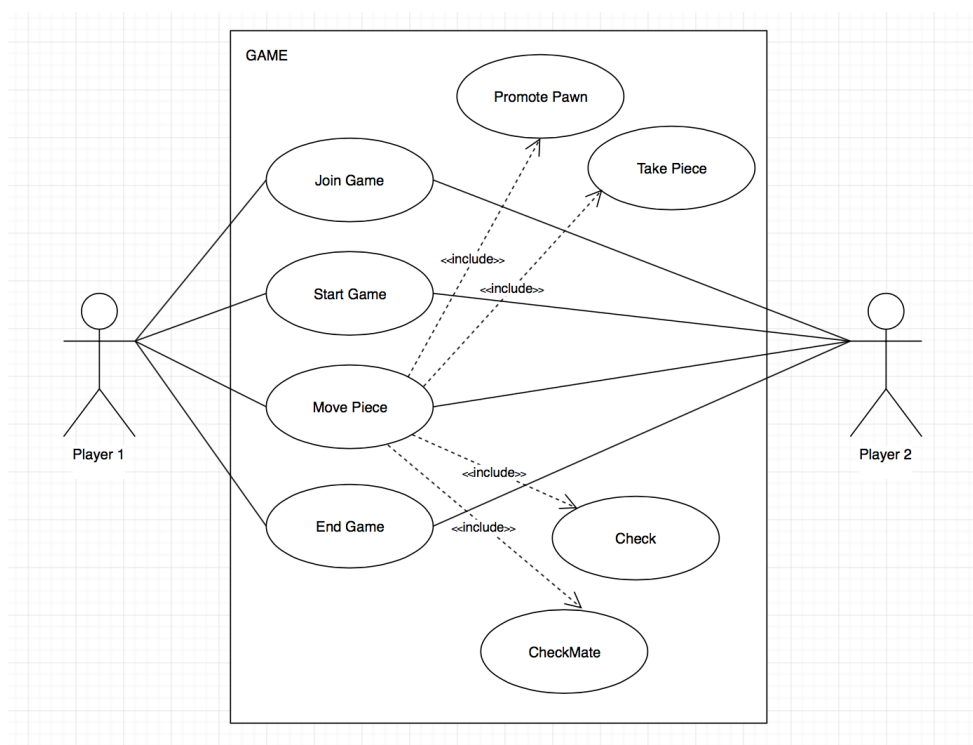
2.1.1.1 Create Account:

Description: User can create account by entering the details that displayed on the screen.
Actors: User
Preconditions: User needs internet access.
Flow of events: 1) User goes to website 2) Clicks on create account to register to the website. 3) Enters personal details 4) Clicks on sign up
Extension: 1) User unable to access website. 2) System fails to register user.
Post Condition: Account Created.
Special Requirements: NULL

2.1.1.2 Spectate Game:

Description: User can create account by entering the details that displayed on the screen.
Actors: User, Outsider
Preconditions: User as well as outsider needs internet access.
Flow of events: For user 1) User goes to website 2) User logs in 3) Clicks any of the available games that are going on For outsider 1) Goes to website. 2) Search for the game to spectate.
Extension: 1) User unable to access website. 2) System fails to show any available games.
Post Condition: Spectates the game.
Special Requirements: NULL

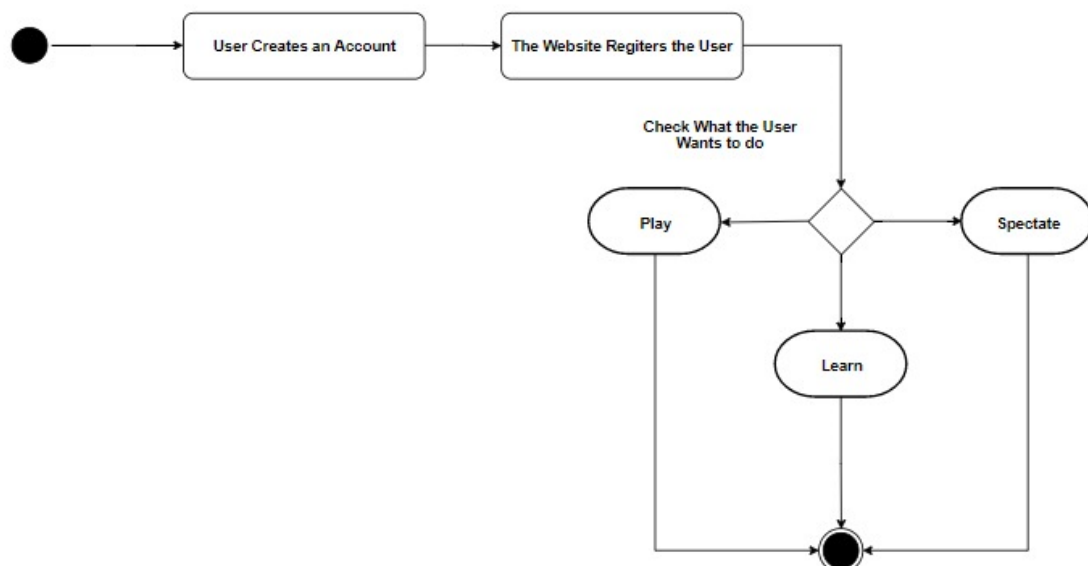
2.1.2 USE CASE 2

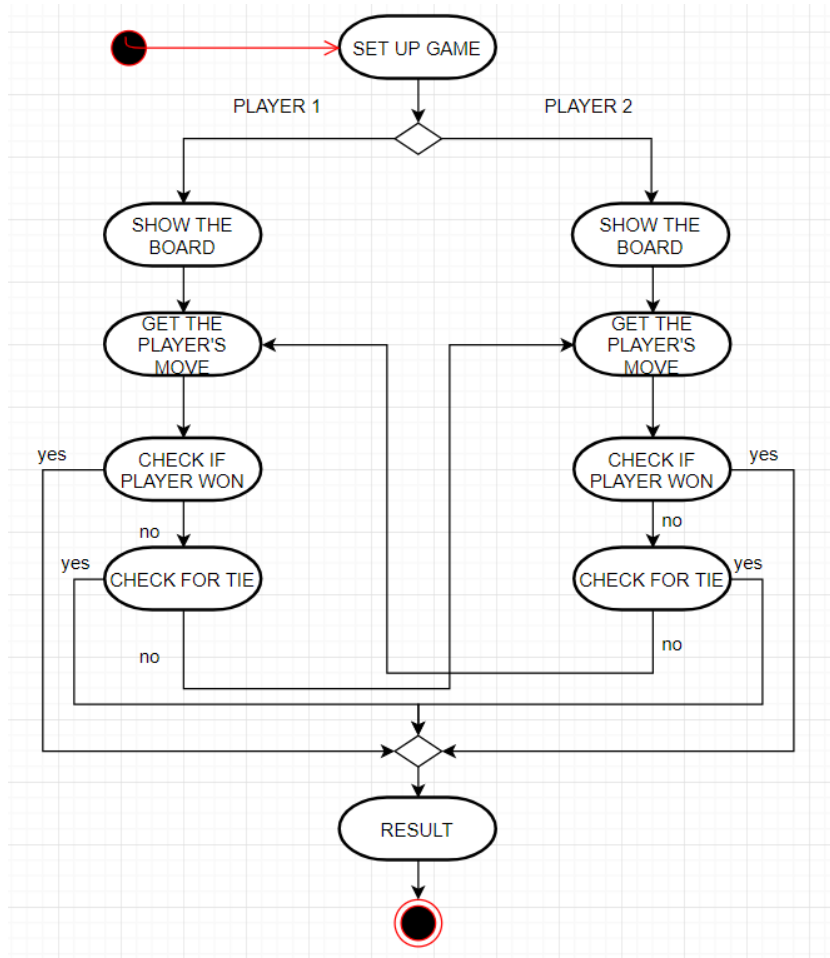


2.1.2.1 Join Game:

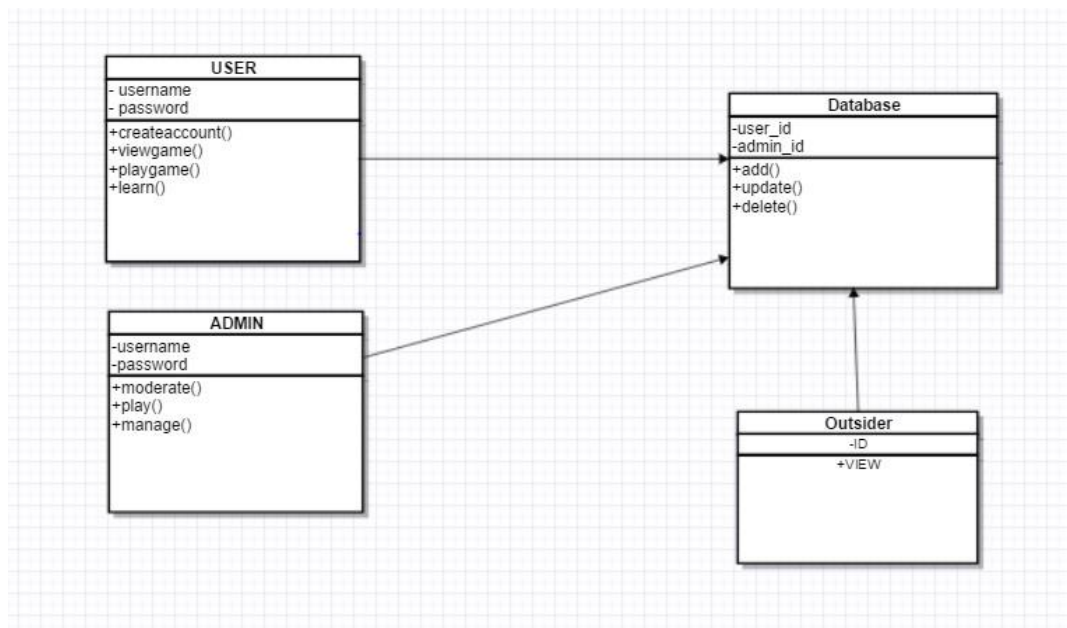
Description: User can join a game.
Actors: Player 1, Player 2
Preconditions: 1) Player needs to be a registered user. 2) Another player should be active.
Flow of events: 1) User goes to website 2) User logs in. 3) Chooses opponent 4) Starts a game.
Extension: 1) User unable to access website. 2) User unable to login 3) User unable to connect to another player.
Post Condition: User joins game
Special Requirements: User needs internet access.

2.2 ACTIVITY DIAGRAM





2.3 CLASS DIAGRAM



2.3.1 CRC CARD

2.3.1.1 USER

USER:	Collaborators
Responsibilities <ul style="list-style-type: none">• User can create an account on the website and fill up the necessary details required• Users can see the on-going games and can challenge any online player free to play a game of chess• User can learn how to play a game of chess by playing an already set up game	<ul style="list-style-type: none">• ADMIN

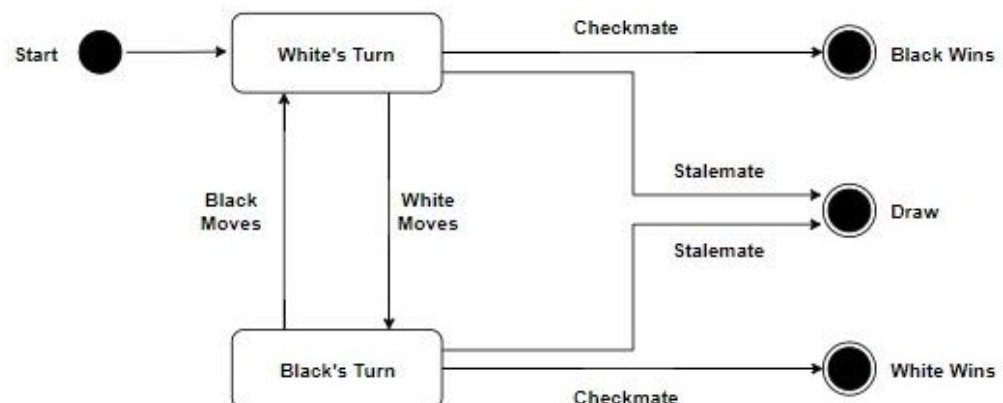
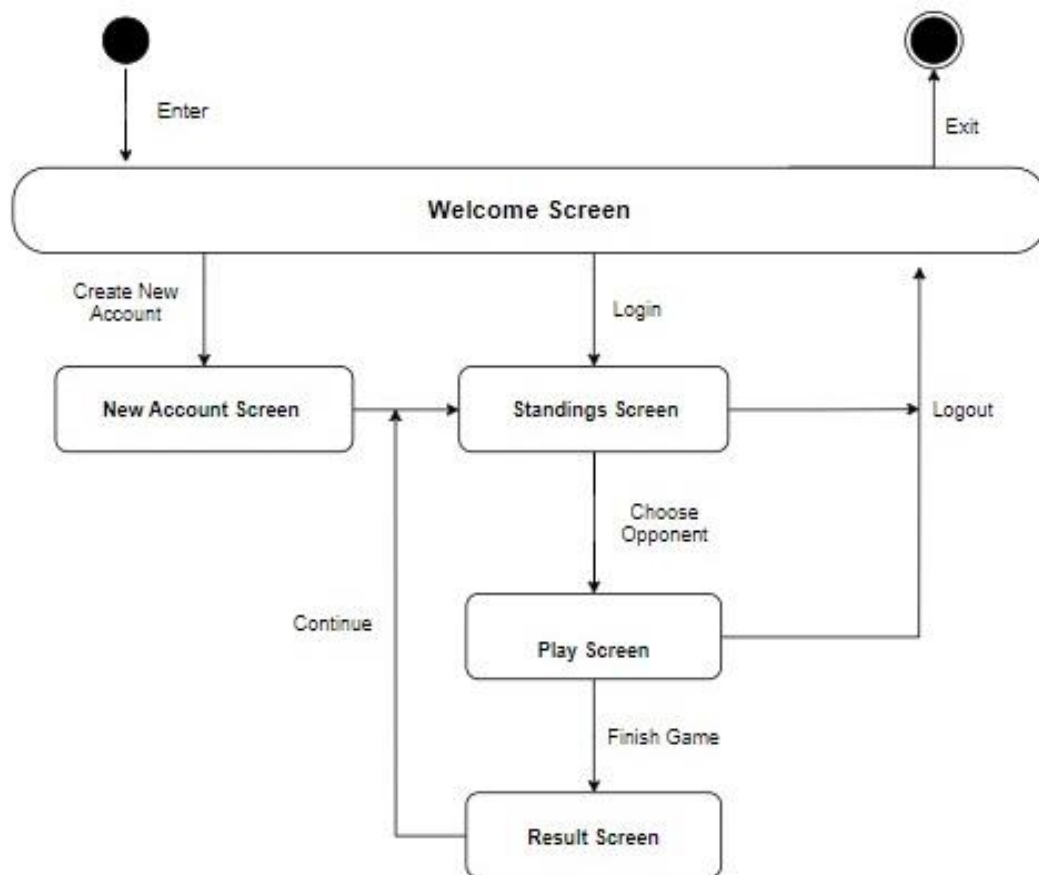
2.3.1.2 ADMIN

ADMIN:	Collaborators
Responsibilities <ul style="list-style-type: none">• Admin can moderate a game and see whether the website is functioning properly• Admin can resolve issues posted by the user in the forum section of the site.• Admin should be available at all times for the user, in cases where a unbiased opinion and verdict is required	<ul style="list-style-type: none">• none

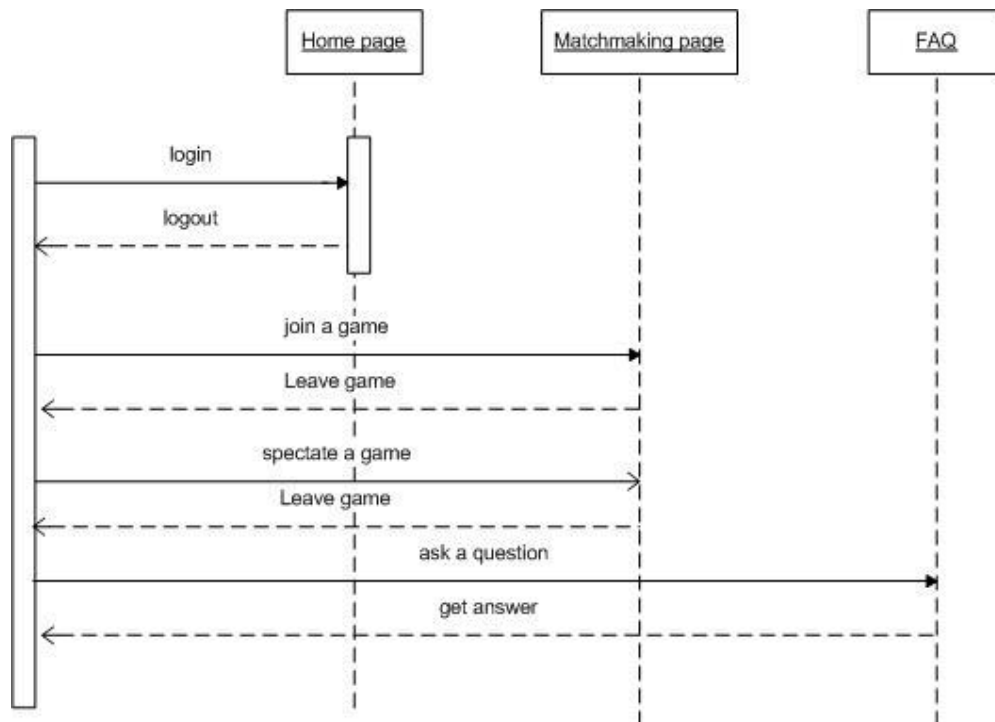
2.3.1.3 OUTSIDER

OUTSIDER:	Collaborators
Responsibilities <ul style="list-style-type: none">• An outsider can view the website as a whole but can't play a game of chess until he makes an account• Outsider can see the ongoing matches and can apply to the admin to spectate an ongoing match.	<ul style="list-style-type: none">• ADMIN

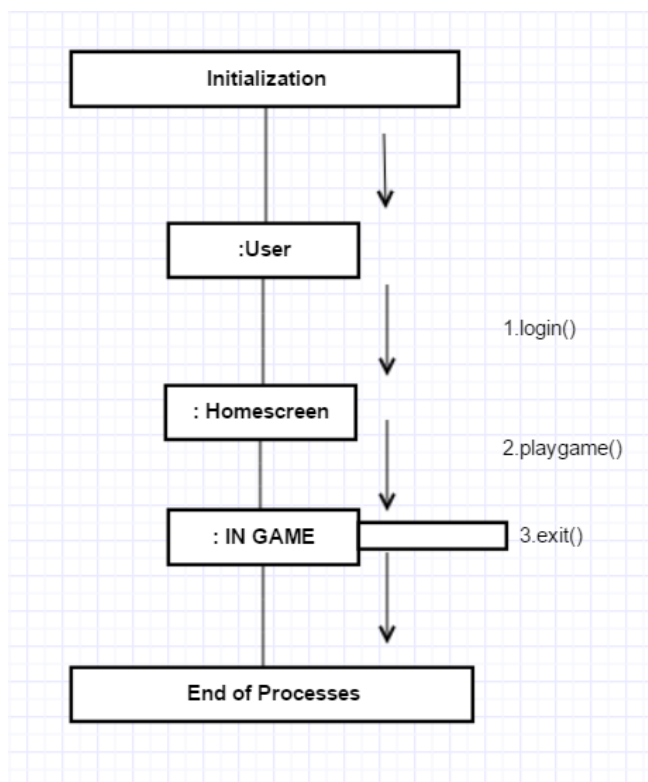
2.4 STATE CHART DIAGRAM



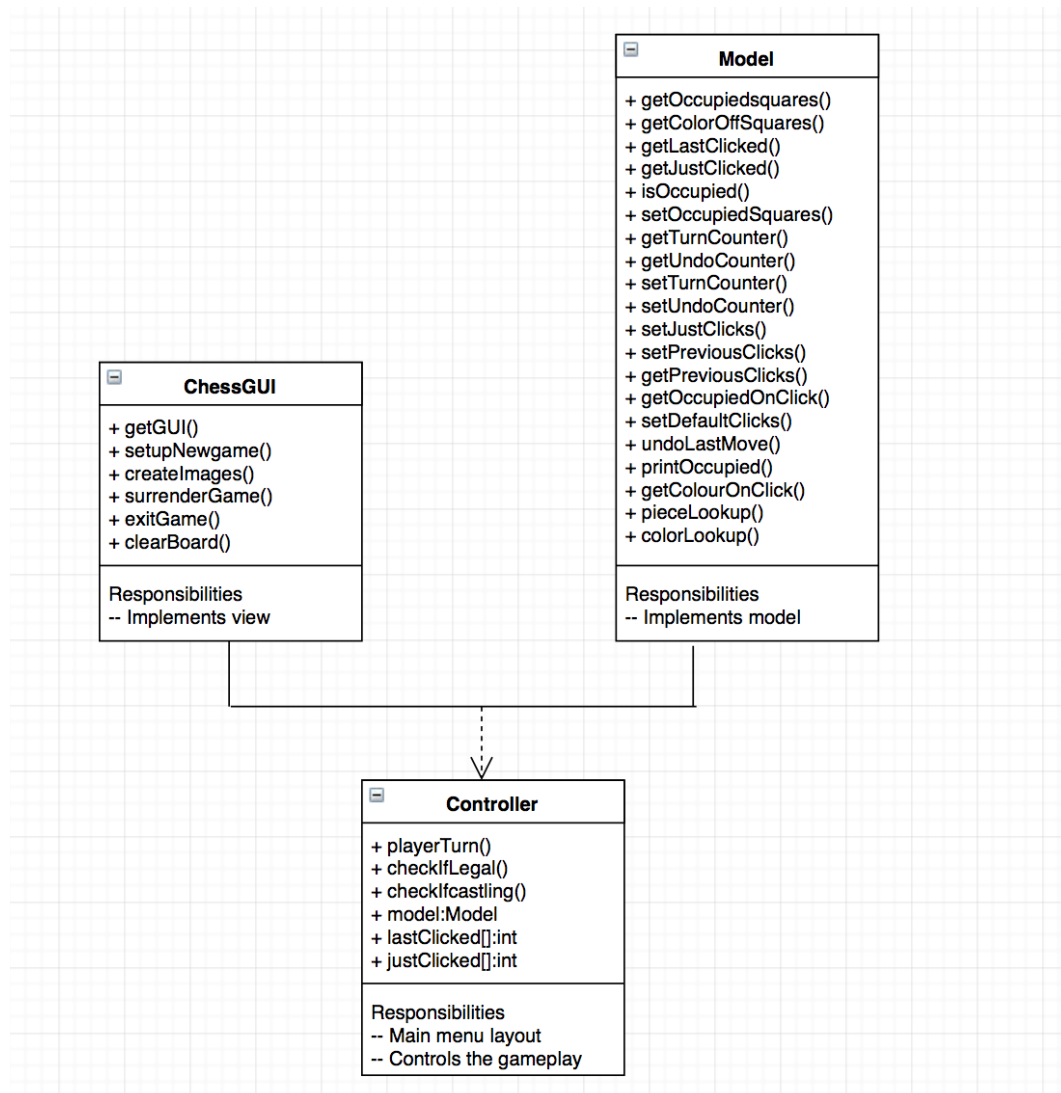
2.5 SEQUENCE DIAGRAM



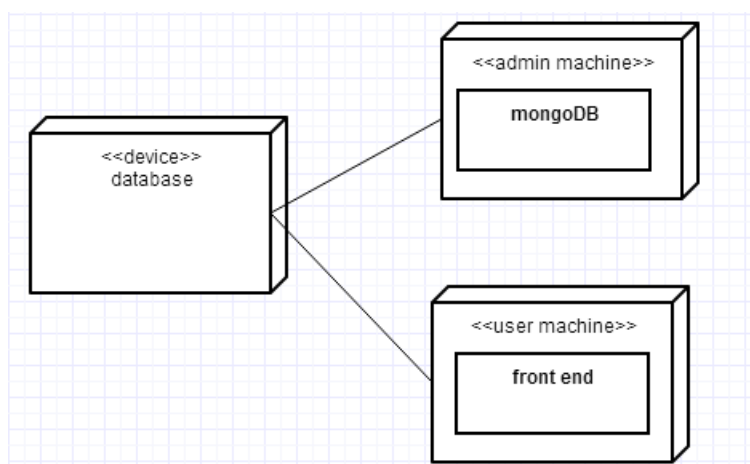
2.6 COLLABORATION DIAGRAM



2.7 COMPONENT DIAGRAM



2.8 DEPLOYMENT DIAGRAM



Software Testing

for

CHECKMATE

Version 1.0

Table of Contents

1) Compilation.....	28
1.1) Test Case 1.....	28
2) Login	29
2.1) Test Case 2.....	29
2.2) Test Case 3.....	30
3) Socket Connection.....	31
3.1) Test Case 4.....	31
4) ChessBoard	32
4.1) Test Case 5.....	32
4.2) Test Case 6.....	33
5) Game Play	34
5.1) Test Case 7.....	34
5.2) Test Case 8.....	35
5.3) Test Case 9.....	36
5.4) Test Case 10	37
6) Movement	38
6.1) Test Case 11.....	38
6.2) Test Case 12.....	39
6.3) Test Case 13.....	40
6.4) Test Case 14.....	41
6.5) Test Case 15.....	42
6.6) Test Case 16.....	43
6.7) Test Case 17.....	44
6.8) Test Case 18.....	45
6.9) Test Case 19.....	46
6.10) Test Case 20.....	47
6.11) Test Case 21.....	48

1) Compilation

1.1) Test Case 1

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00C	Test Designed by: Nirut
Test Priority (Low/Medium/High): High	Test Designed date: 19/3/2018
Module Name: Chess	Test Executed by: Parth
Test Title: Compilation	Test Execution date: 19/3/2018
Description: Verify whether the code is running successfully without errors.	
Pre-conditions: Appropriate database and compiler.	
Dependencies: none	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Navigate to terminal	mongod	The program is compiled properly	As same as expected result		Pass
3	Open another terminal	mongo				
4	Open another terminal					
5	Go to the directory where code is present	Node app				

Post Condition: App in execution.

```
C:\WINDOWS\system32\cmd.exe - node app
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\nirut>cd..

C:\Users>cd..

C:\>d:

D:\>cd "college/fourth sem/software engineering/project"

D:\college\fourth sem\software engineering\project>cd n

D:\college\fourth sem\software engineering\project\n>node app
listening on *: 3000
```

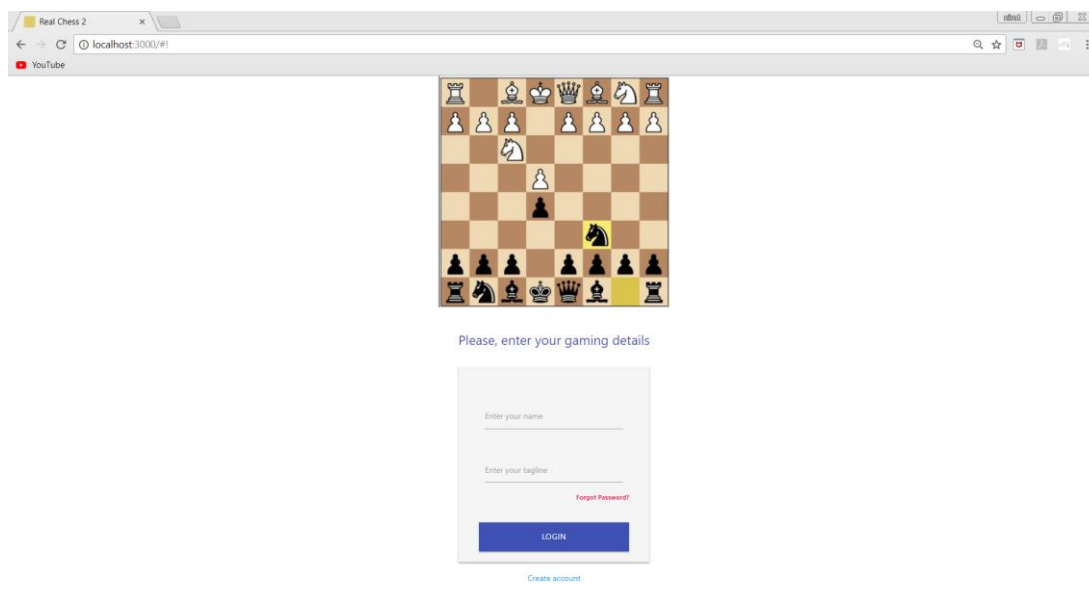
2) Login

2.1) Test Case 2

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00L.1	Test Designed by: Lakshay
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Login	Test Executed by: Nirut
Test Title: Create account	Test Execution date: 19/3/2018
Description: Test the online chess create account page.	
Pre-conditions: User should enter all the details as asked.	
Dependencies: none	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Navigate to create account page	User: Nirut	User should be able to create account	Unable to create account for the specified user		Fail
2	Provide valid username	Password: hey				
3	Provide valid password					
4	Click on Create account button					

Post Condition: User remains on the create account page.

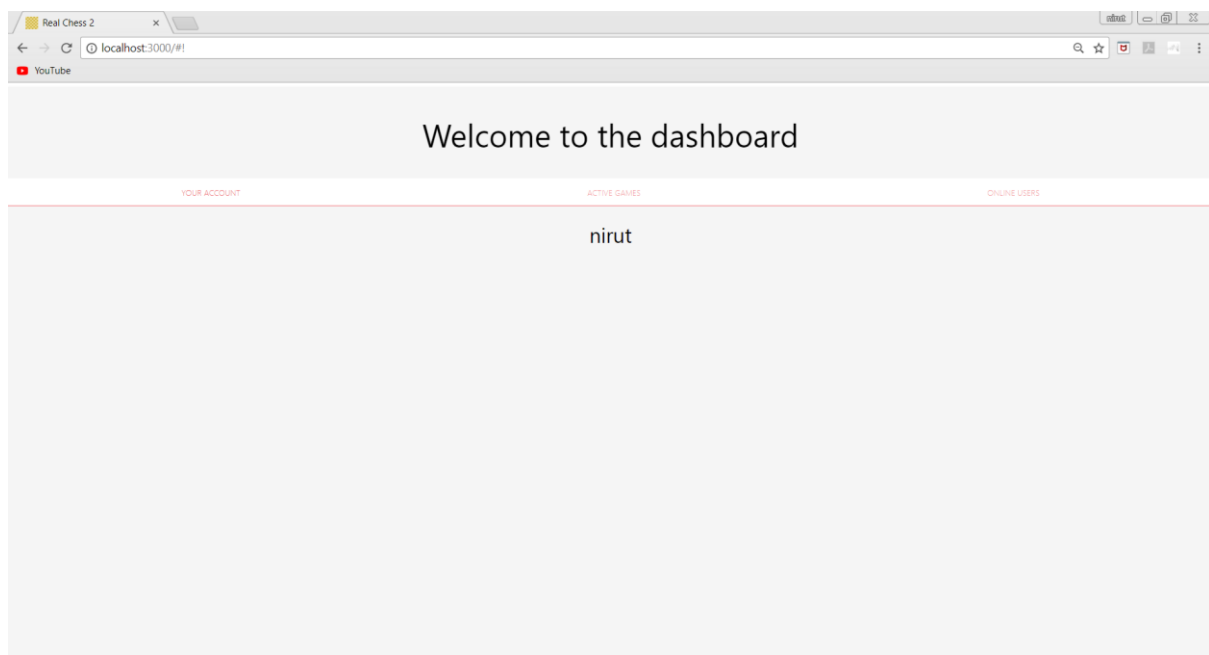


2.2) Test Case 3

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00L.2	Test Designed by: Nirut
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Login	Test Executed by: Lakshay
Test Title: Verify login with valid username and password	Test Execution date: 19/3/2018
Description: Test the online chess login page	
Pre-conditions: User has valid username and password	
Dependencies: none	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Navigate to login page	User: Nirut	User should be able to login	User is navigated to		Pass
2	Provide valid username	Password: hey		dashboard with successful		
3	Provide valid password			login		
4	Click on Login button					

Post Condition: User entered into the dashboard.



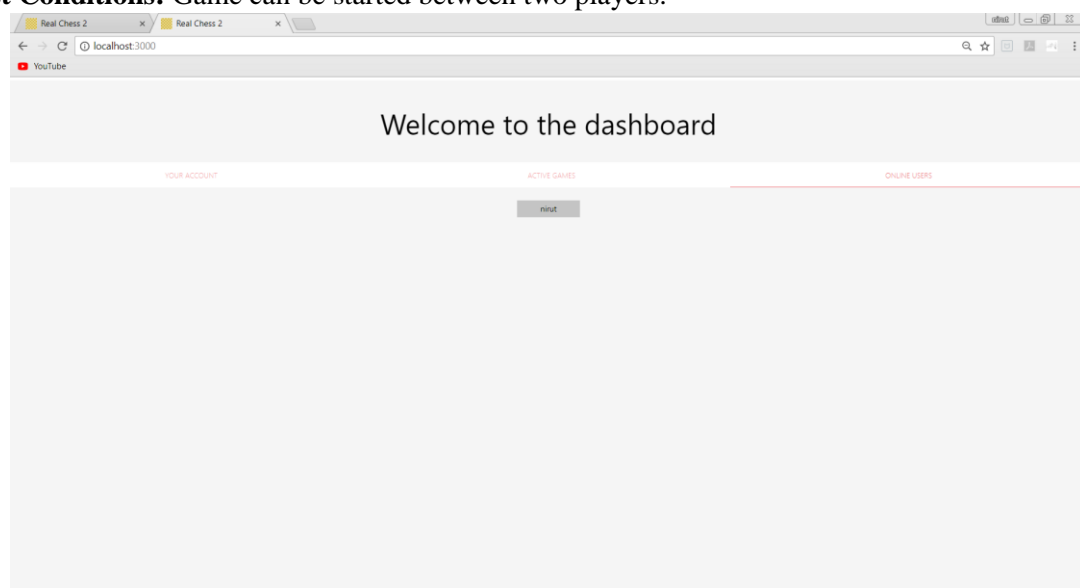
3) Socket Connection

3.1) Test Case 4

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00S	Test Designed by: Parth
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Socket Connection	Test Executed by: Lakshay
Test Title: Number of active players	Test Execution date: 19/3/2018
Description: Test the number of online players	
Pre-conditions: One user is active prior to the second login	
Dependencies: login	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Navigate to login page	User: Lakshay	User should be able to login	User is navigated to		Pass
2	Provide valid username	Password: hello		dashboard with successful		
3	Provide valid password			login		
4	Click on Login button					
5	Check active players page	Shows Nirut online	User should be able to see which player is online	As expected		Pass

Post Conditions: Game can be started between two players.



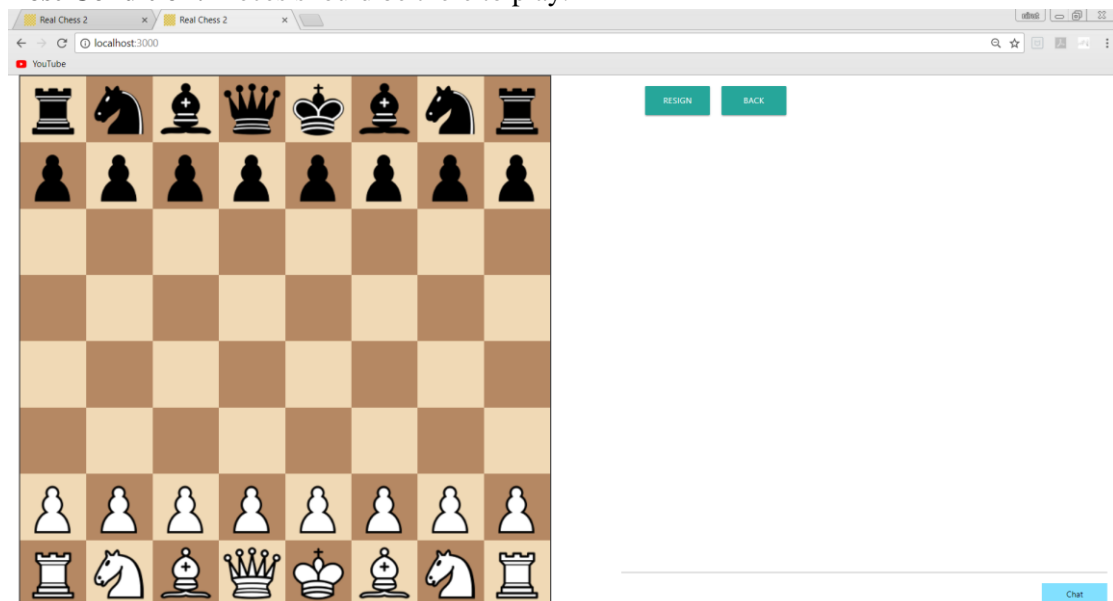
4) Board

4.1) Test Case 5

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00B.1	Test Designed by: Parth
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Chessboard	Test Executed by: Lakshay
Test Title: Dimension and interface of chessboard	Test Execution date: 19/3/2018
Description: Verify whether the dimension of the chessboard is 8x8 and also check whether the color of each box is alternate black and white.	
Pre-conditions: A game should be started between 2 players	
Dependencies: Socket connection between the two players	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Check active players page	Shows Nirut online	User should be able to start a new game or be a spectator	As expected		Pass
2	Check whether all the pieces are available on both sides or not	Go to the game page and perform the test	Dimension of the chess board is 8x8 and colour of the box is alternating black and white thus satisfying the rules of the chess.	Result as expected		Pass

Post Condition: Pieces should be there to play.

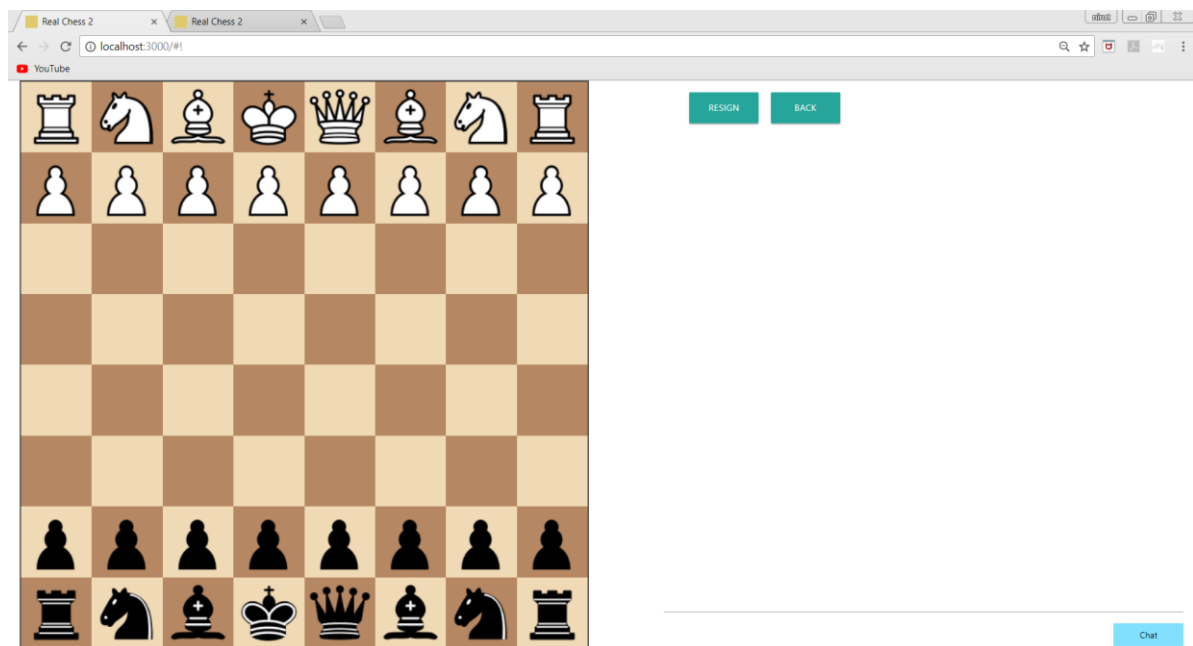


4.2) Test Case 6

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00B.2	Test Designed by: Nirut
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Chessboard	Test Executed by: Lakshay
Test Title: All pieces on board are available	Test Execution date: 19/3/2018
Description: Verify whether all pieces on both sides are available or not	
Pre-conditions: User has valid username and password; A game should be started between 2 players	
Dependencies: Two players must be in a game	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Check active players page	Shows Nirut online	User should be able to start a new game or be a spectator	As expected		Pass
2	Check whether all the pieces are available on both sides or not	Go to the game page and perform the test	All the pieces are available and fairly organised for both sides which are satisfying the basic rules of the game is shown below.	Result as expected		Pass

Post Condition: All the pieces are available and fairly organized for both sides.



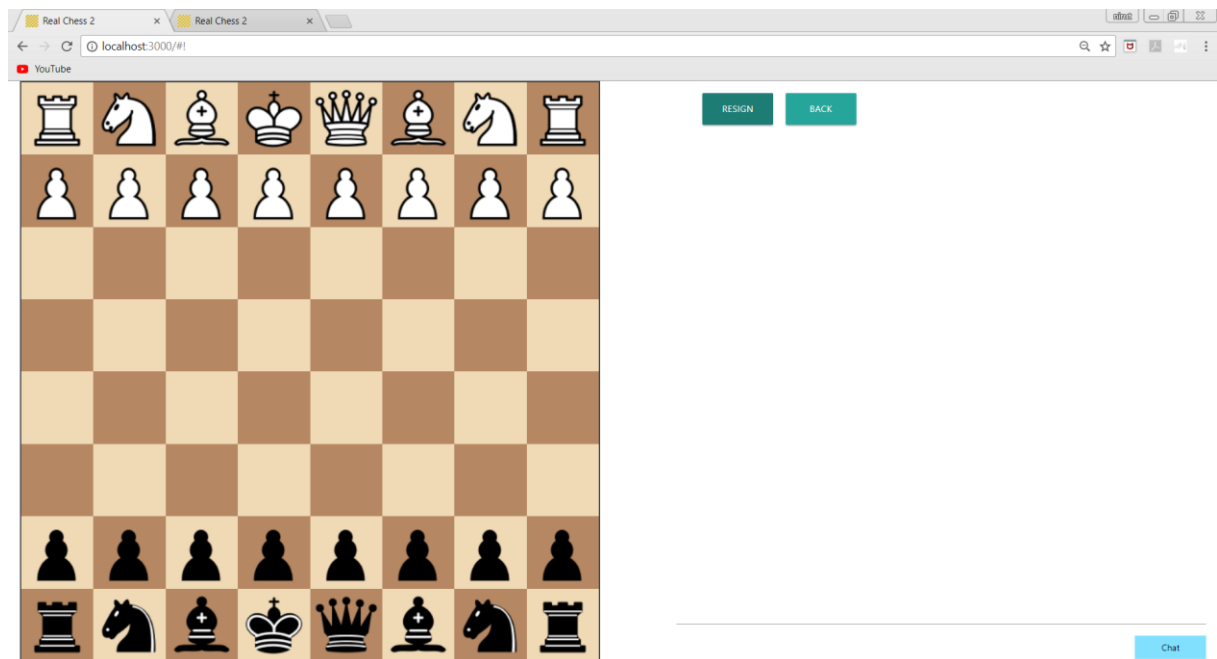
5) Game Play

5.1) Test Case 7

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00G.1	Test Designed by: Nirut
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Gameplay	Test Executed by: Lakshay
Test Title: Resign the game	Test Execution date: 19/3/2018
Description: Verify whether the user can resign the current going game.	
Pre-conditions: User should be playing a game.	
Dependencies: none	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Get into the game	Connect with the other player online	Connection completed	As expected		
2	Click on the button resign	Gameplay to NULL	Game can be discarded and other user can view that the first user left the game.	User remains in the gameplay		Fail

Post Condition: User remains in the gameplay.

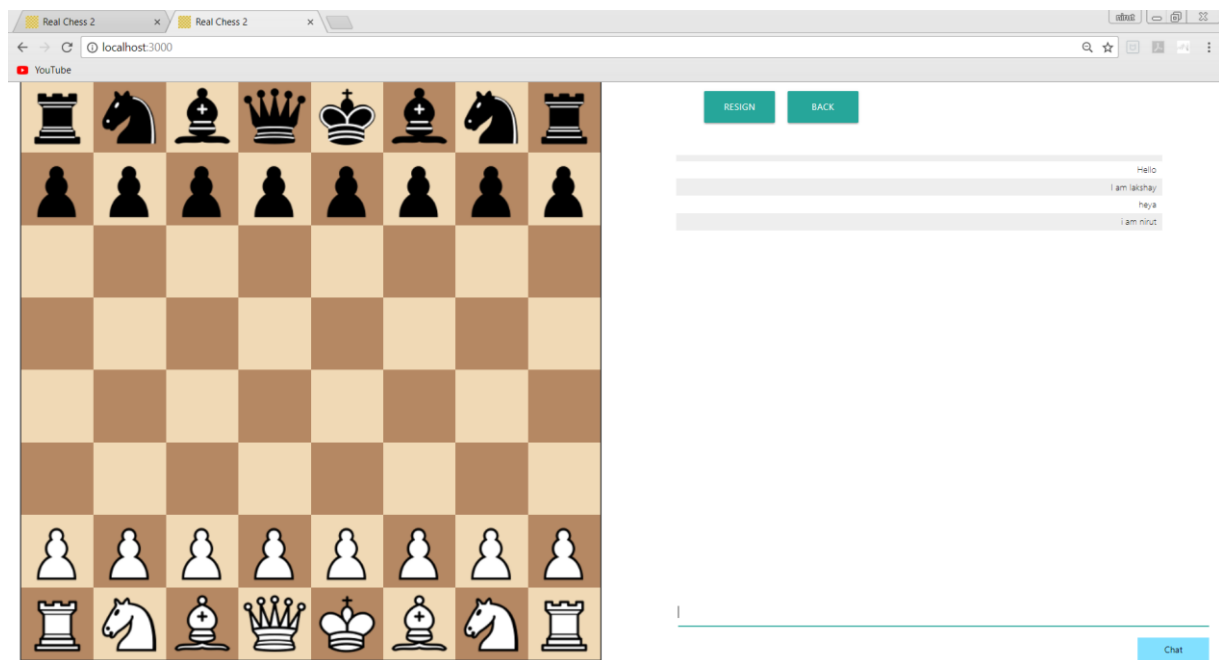


5.2) Test Case 8

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00G.2	Test Designed by: Parth
Test Priority (Low/Medium/High): Low	Test Designed date: 19/3/2018
Module Name: Gameplay	Test Executed by: Lakshay
Test Title: Chat	Test Execution date: 19/3/2018
Description: Verify whether connected players can chat.	
Pre-conditions: User should be playing a game.	
Dependencies: none	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Get into the game	Connect with the other player online	Connection completed	As expected		
2	Click on chat button	Chat button				
3	Type message and press enter	Entered message	Another user can view the sent message	As expected		Pass

Post Condition: User can chat with the connected player.

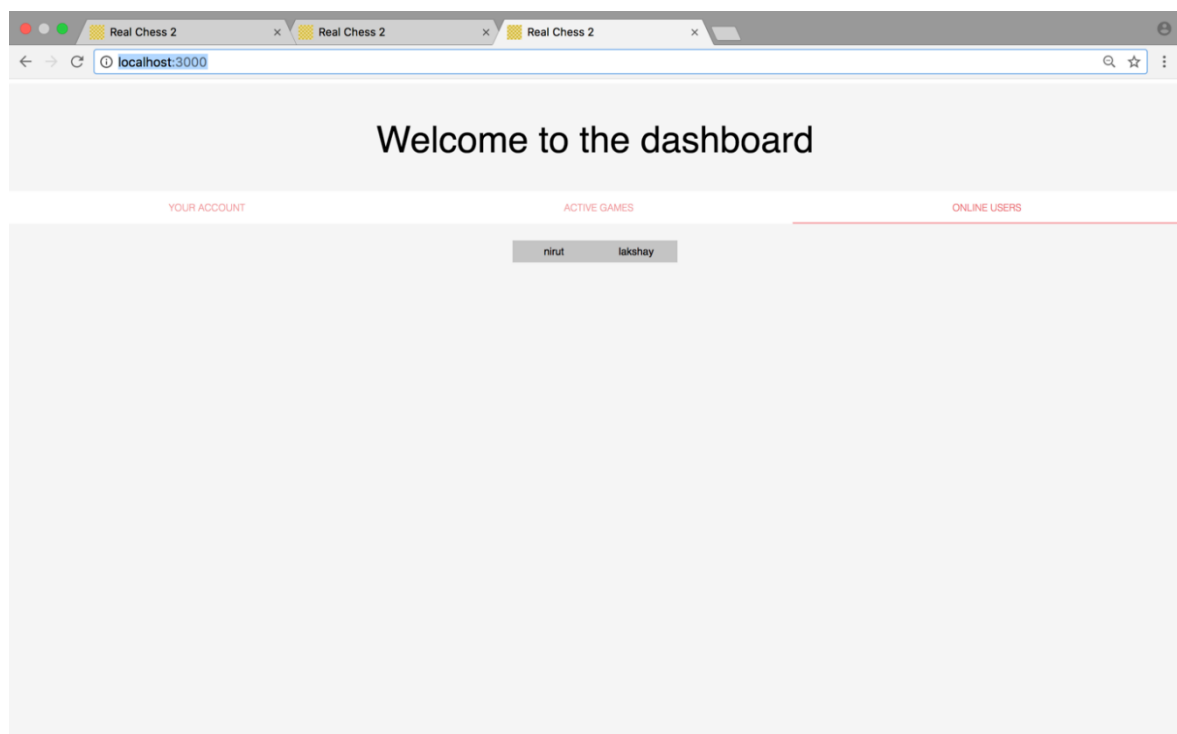


5.3) Test Case 9

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00G.3	Test Designed by: Parth
Test Priority (Low/Medium/High): High	Test Designed date: 19/3/2018
Module Name: Gameplay	Test Executed by: Nirut
Test Title: End Game	Test Execution date: 19/3/2018
Description: Verify whether the game is ending properly	
Pre-conditions: User should be playing a game.	
Dependencies: none	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Strategy for checkmate	movements				
2	Capture of king	movements	End of a game	As expected		Pass
3			Pop up displayed	NO popup to show		Fail

Post Condition: User can play another game.

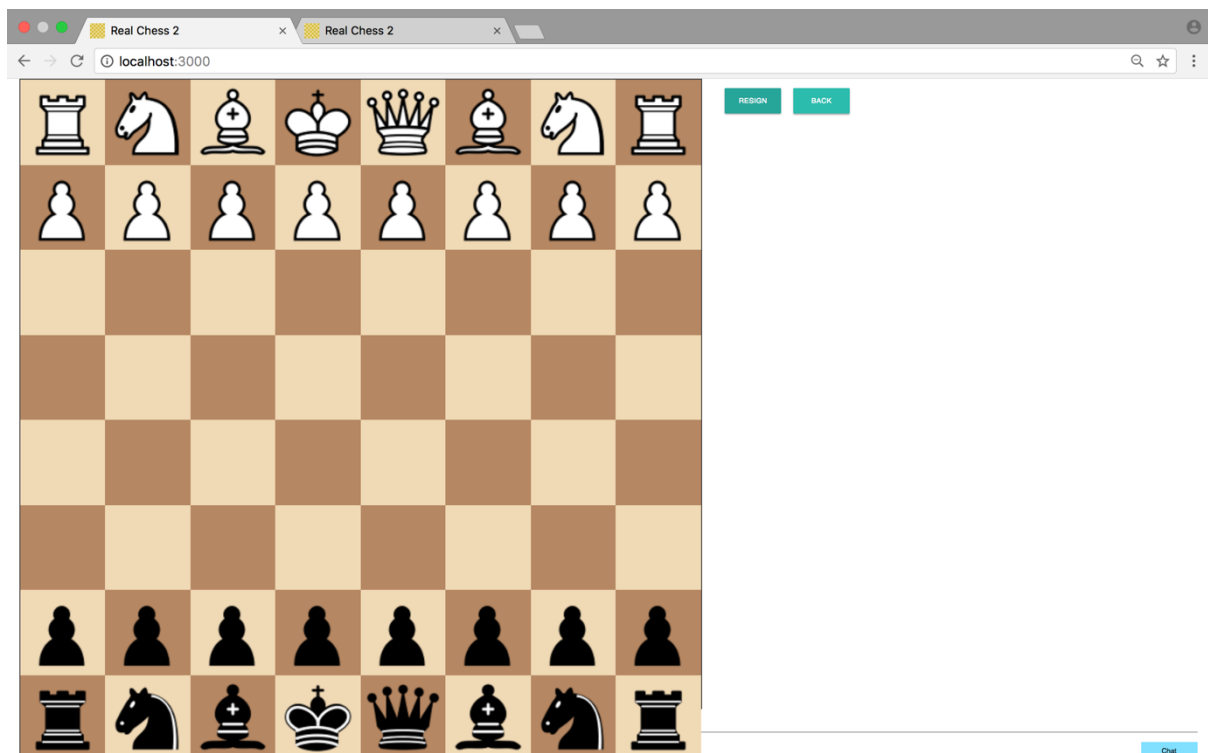


5.4) Test Case 10

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00G.4	Test Designed by: Nirut
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Gameplay	Test Executed by: Parth
Test Title: Back to dashboard	Test Execution date: 19/3/2018
Description: Verify whether the user can switch to dashboard while a game is going on.	
Pre-conditions: User should be playing a game.	
Dependencies: none	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	User press the back button	Back button	User goes back to dashboard	As expected		Pass

Post Condition: User can play another game.



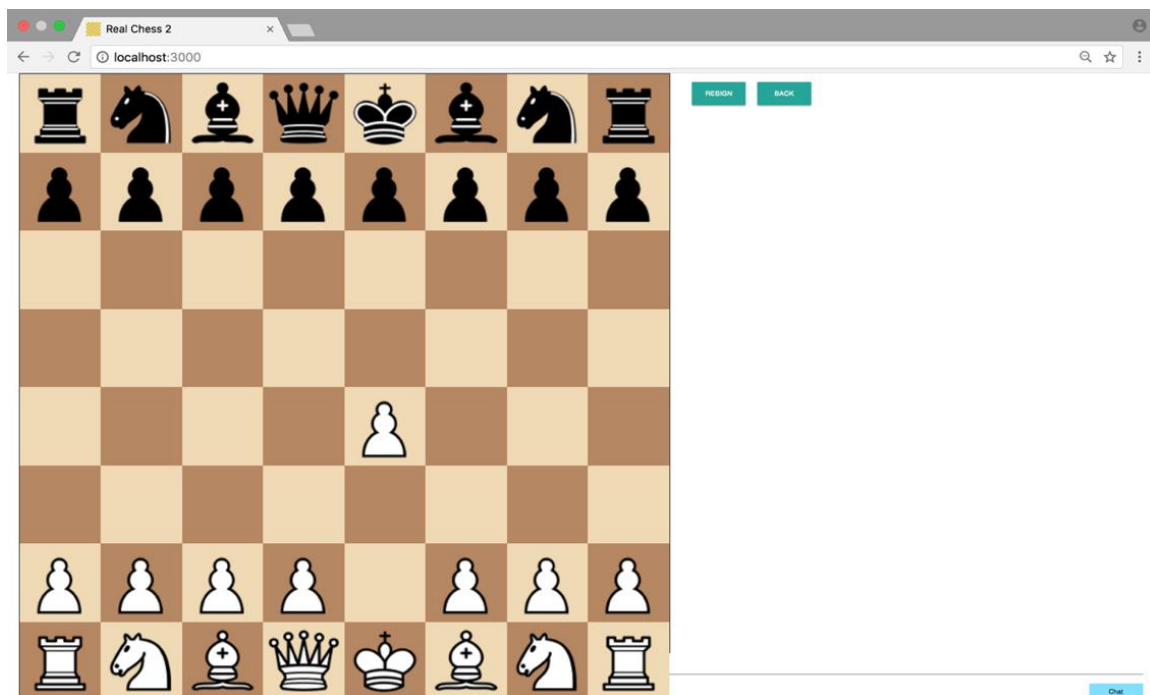
6) Movements

6.1) Test Case 11

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00M.1	Test Designed by: Lakshay
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Chessboard: Movement	Test Executed by: Nirut
Test Title: Movement of Pawn	Test Execution date: 19/3/2018
Description: Verify whether the pawn moves to a valid position	
Pre-conditions: User has valid username and password; A game should be started between 2 players	
Dependencies: Two players must be in a game	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Check active players page	Shows Lakshay online	User should be able to start a new game or be a spectator	As expected	Yes R5.1	Pass
2	Move Pawn to a location	Pawn in the chess game	Pawn should move to the valid position. If the position is invalid it should come back to its original position.	Result as expected		Pass

Post Condition: Pawn moves to required position

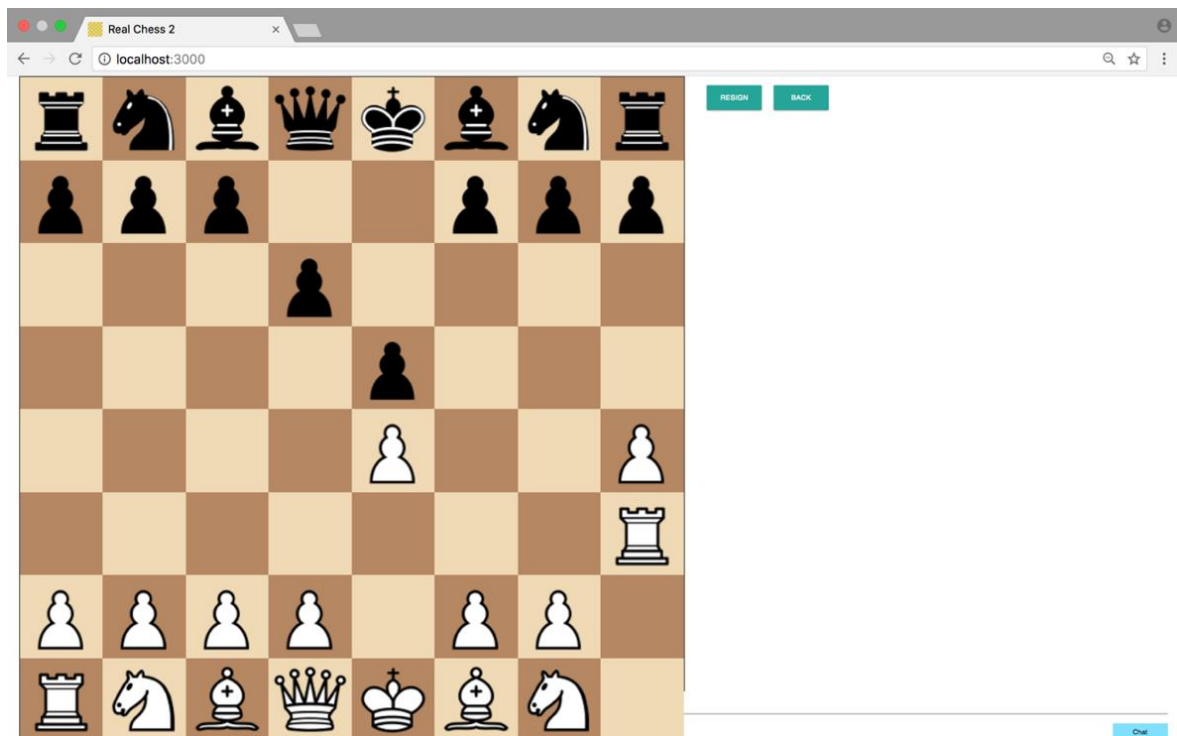


6.2) Test Case 12

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00M.2	Test Designed by: Lakshay
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Chessboard: Movement	Test Executed by: Parth
Test Title: Movement of Rook	Test Execution date: 19/3/2018
Description: Verify whether the rook moves to a valid position	
Pre-conditions: User has valid username and password; A game should be started between 2 players	
Dependencies: Two players must be in a game	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Check active players page	Shows Lakshay online	User should be able to start a new game or be a spectator	As expected	YES R5.2	Pass
2	Move Rook to a location	Rook in the chess game	Rook should move to the valid position. If the position is invalid it should come back to its original position.	Result as expected		Pass

Post Condition: Rook moves to required position

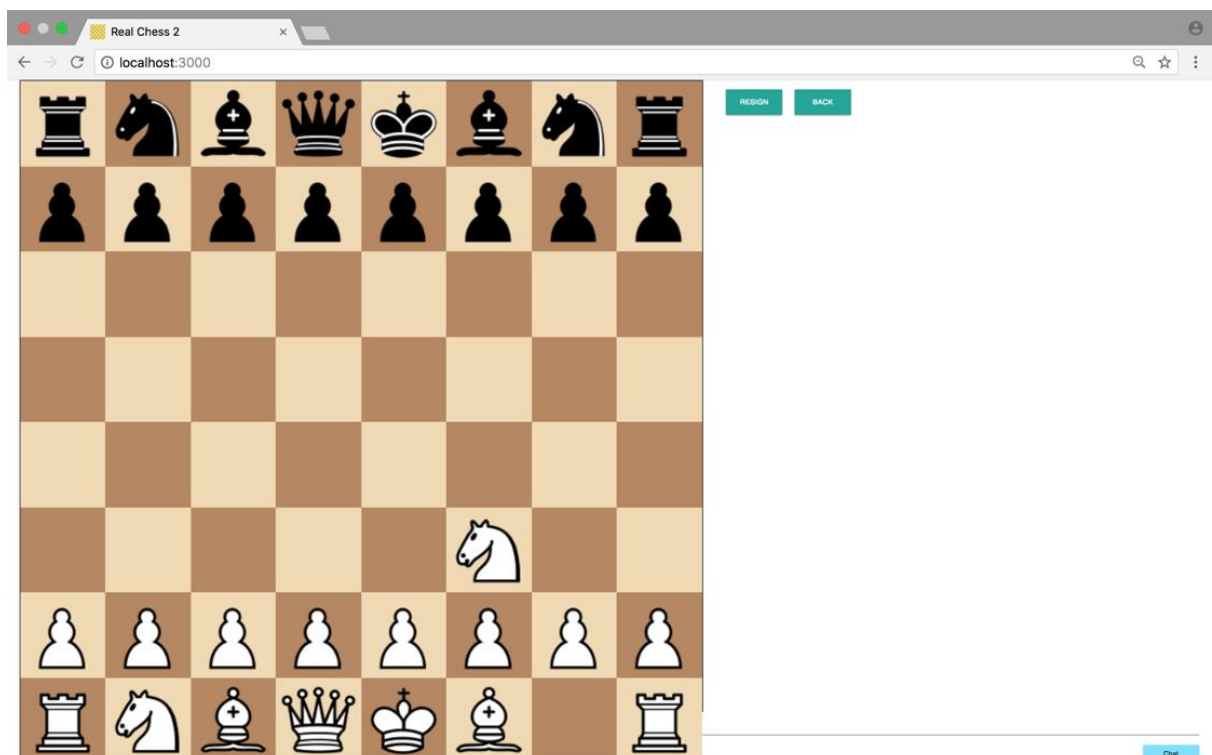


6.3) Test Case 13

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00M.3	Test Designed by: Lakshay
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Chessboard: Movement	Test Executed by: Parth
Test Title: Movement of Knight	Test Execution date: 19/3/2018
Description: Verify whether the knight moves to a valid position	
Pre-conditions: User has valid username and password; A game should be started between 2 players	
Dependencies: Two players must be in a game	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Check active players page	Shows Lakshay online	User should be able to start a new game or be a spectator	As expected	YES R5.3	Pass
2	Move Knight to a location	Knight in the chess game	Knight should move to the valid position. If the position is invalid it should come back to its original position.	Result as expected		Pass

Post Condition: Knight moves to required position

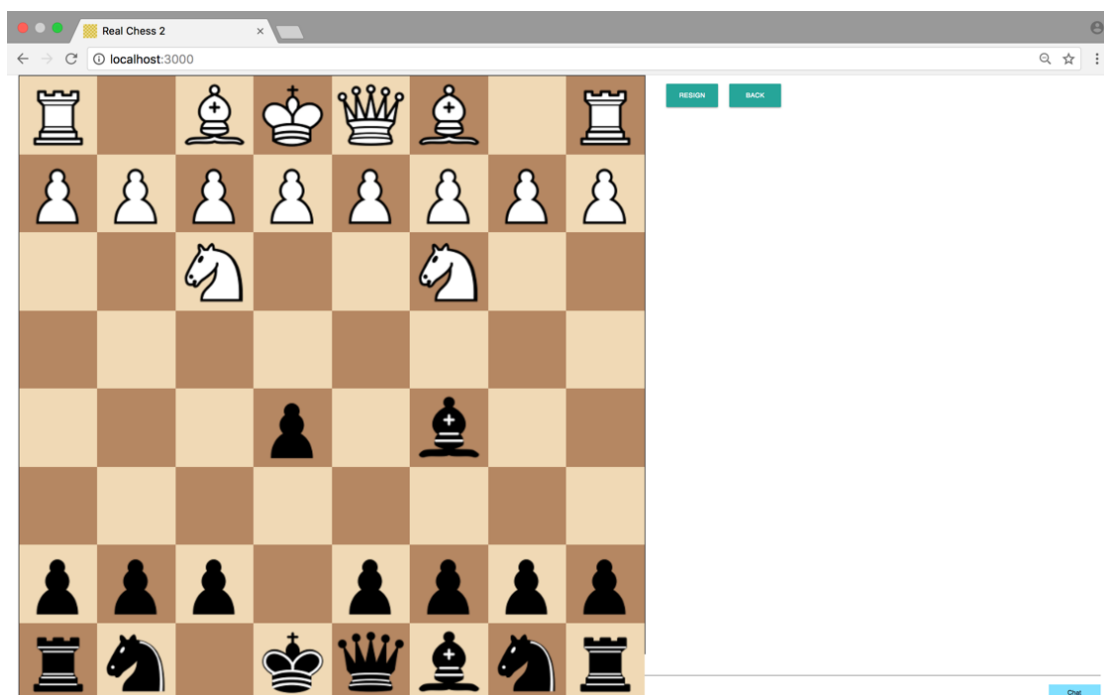


6.4) Test Case 14

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00M.4	Test Designed by: Lakshay
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Chessboard: Movement	Test Executed by: Nirut
Test Title: Movement of Bishop	Test Execution date: 19/3/2018
Description: Verify whether the bishop moves to a valid position	
Pre-conditions: User has valid username and password; A game should be started between 2 players	
Dependencies: Two players must be in a game	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Check active players page	Shows Lakshay online	User should be able to start a new game or be a spectator	As expected	YES R5.4	Pass
2	Move Bishop to a location	Bishop in the chess game	Bishop should move to the valid position. If the position is invalid it should come back to its original position.	Result as expected		Pass

Post Condition: Bishop moves to required position

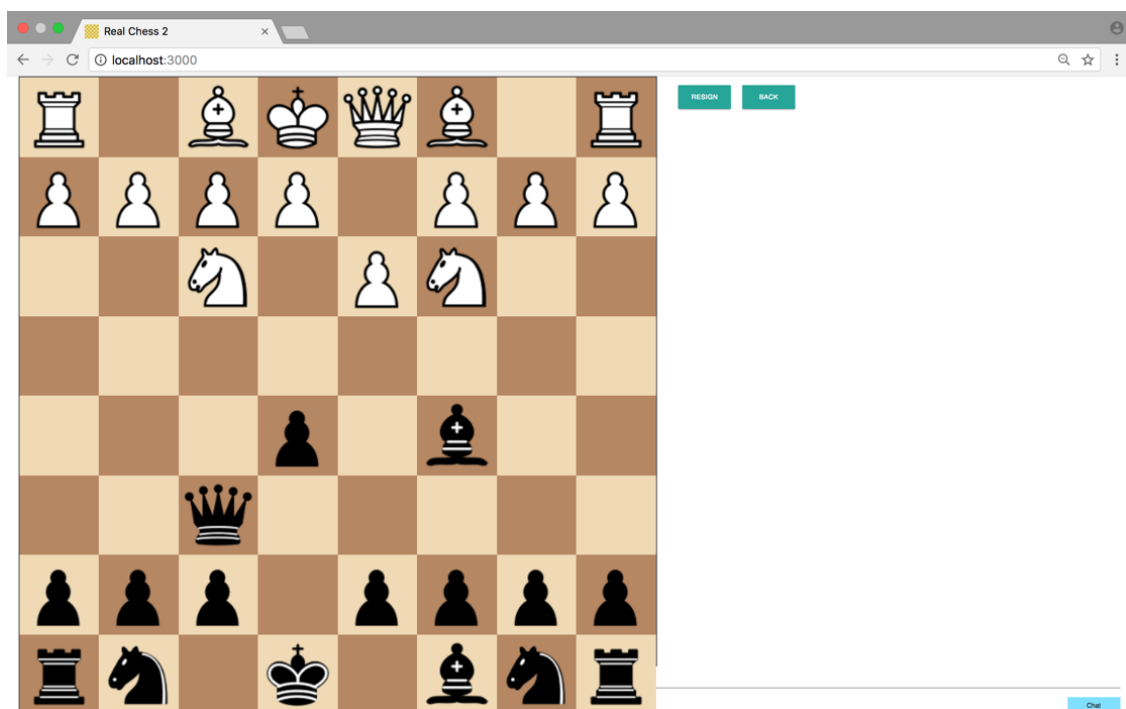


6.5) Test Case 15

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00M.5	Test Designed by: Parth
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Chessboard: Movement	Test Executed by: Nirut
Test Title: Movement of Queen	Test Execution date: 19/3/2018
Description: Verify whether the queen moves to a valid position	
Pre-conditions: User has valid username and password; A game should be started between 2 players	
Dependencies: Two players must be in a game	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Check active players page	Shows Lakshay online	User should be able to start a new game or be a spectator	As expected	YES R5.5	Pass
2	Move Queen to a location	Queen in the chess game	Queen should move to the valid position. If the position is invalid it should come back to its original position.	Result as expected		Pass

Post Condition: Queen moves to required position



6.6) Test Case 16

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00M.6	Test Designed by: Lakshay
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Chessboard: Movement	Test Executed by: Nirut
Test Title: Movement of King	Test Execution date: 19/3/2018
Description: Verify whether the king moves to a valid position	
Pre-conditions: User has valid username and password; A game should be started between 2 players	
Dependencies: Two players must be in a game	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Check active players page	Shows Lakshay online	User should be able to start a new game or be a spectator	As expected	YES R5.6	Pass
2	Move King to a location	King in the chess game	King should move to the valid position. If the position is invalid it should come back to its original position.	Result as expected		Pass

Post Condition: King moves to required position

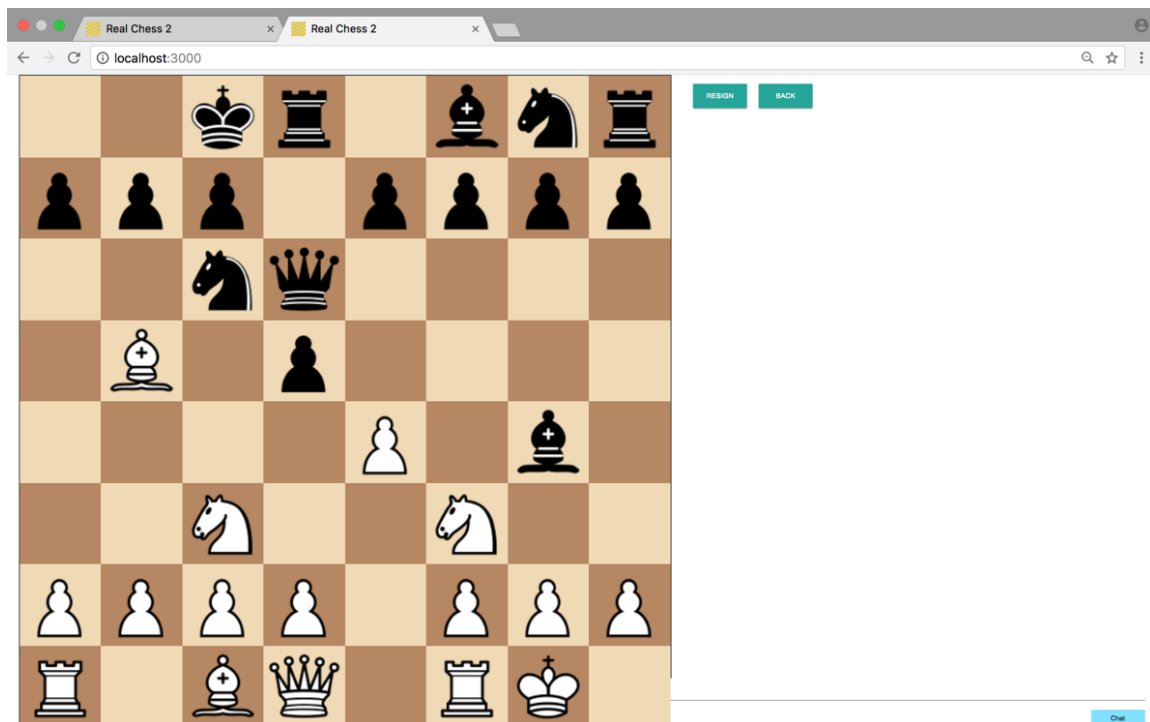


6.7) Test Case 17

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00M.7	Test Designed by: Lakshay
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Chessboard: Movement	Test Executed by: Nirut
Test Title: Castling	Test Execution date: 19/3/2018
Description: Verify whether Castling is possible	
Pre-conditions: User has valid username and password; A game should be started between 2 players	
Dependencies: Two players must be in a game	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Check active players page	Shows Lakshay online	User should be able to start a new game or be a spectator	As expected	YES R5.7	Pass
2	Perform Castling	King, Rook in the chess game	King and Rook should move to their respective positions.	Result as expected		Pass

Post Condition: Castling is performed

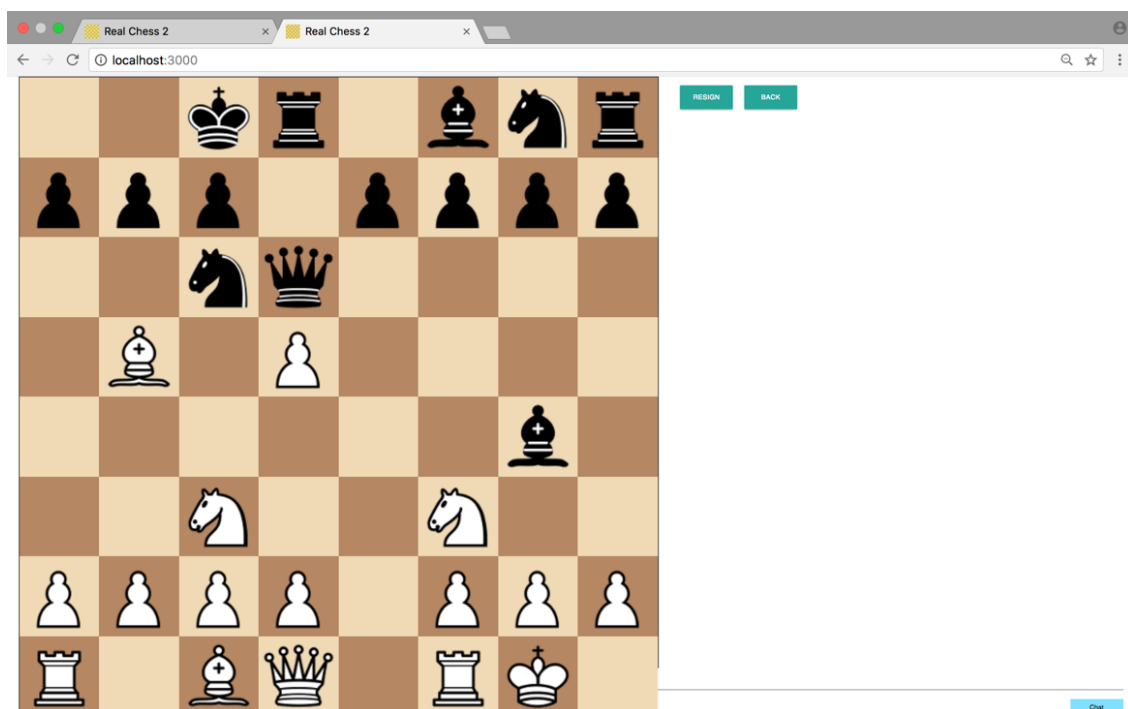


6.8) Test Case 18

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00M.8	Test Designed by: Lakshay
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Chessboard: Movement	Test Executed by: Parth
Test Title: General Capture	Test Execution date: 19/3/2018
Description: Verify whether general capturing is working.	
Pre-conditions: User has valid username and password; A game should be started between 2 players	
Dependencies: Two players must be in a game	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Check active players page	Shows Lakshay online	User should be able to start a new game or be a spectator	As expected	YES R5.8	Pass
2	Capture a piece	Two opponent pieces	One piece should capture another piece and that piece should be removed from the chess board.	Result as expected		Pass

Post Condition: Piece is captured.

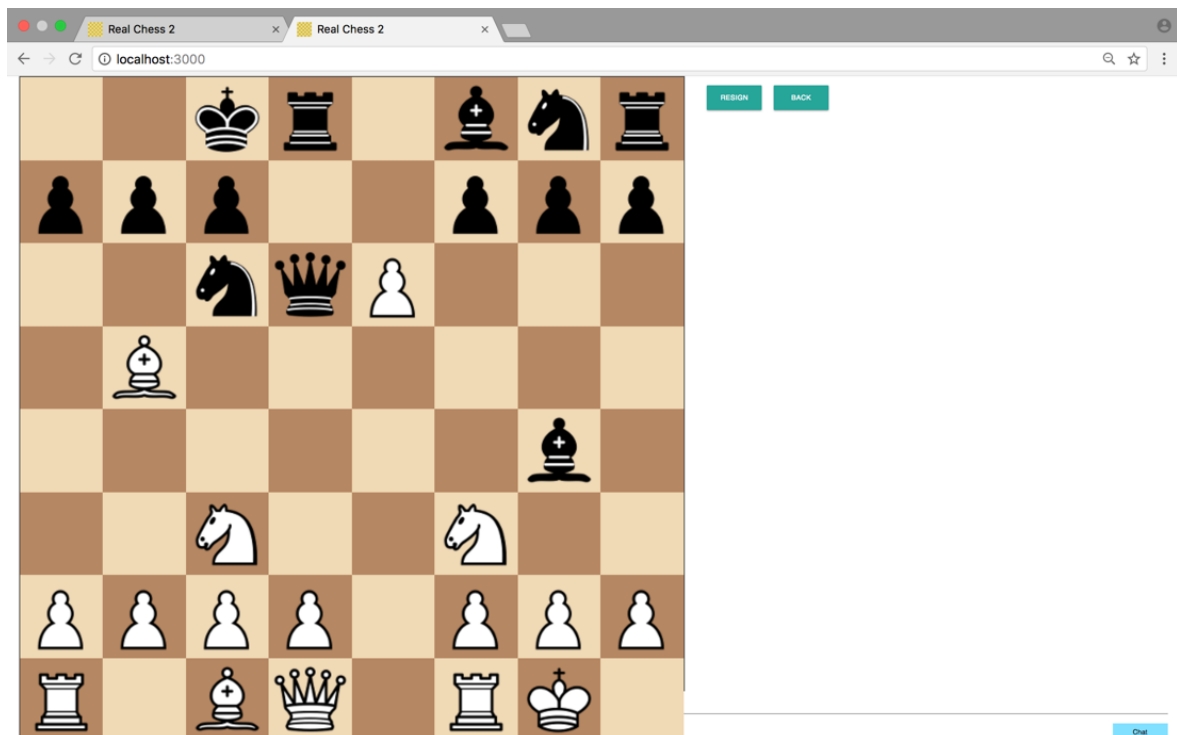


6.9) Test Case 19

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00M.9	Test Designed by: Lakshay
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Chessboard: Movement	Test Executed by: Parth
Test Title: En Passant	Test Execution date: 19/3/2018
Description: Verify whether En Passant is possible	
Pre-conditions: User has valid username and password; A game should be started between 2 players	
Dependencies: Two players must be in a game	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Check active players page	Shows Lakshay online	User should be able to start a new game or be a spectator	As expected	YES R5.10	Pass
2	Perform En Passant	Pawn in the chess game	Pawn captures an opponent piece using En Passant and moves to its respective position.	Result as expected		Pass

Post Condition: En Passant is performed

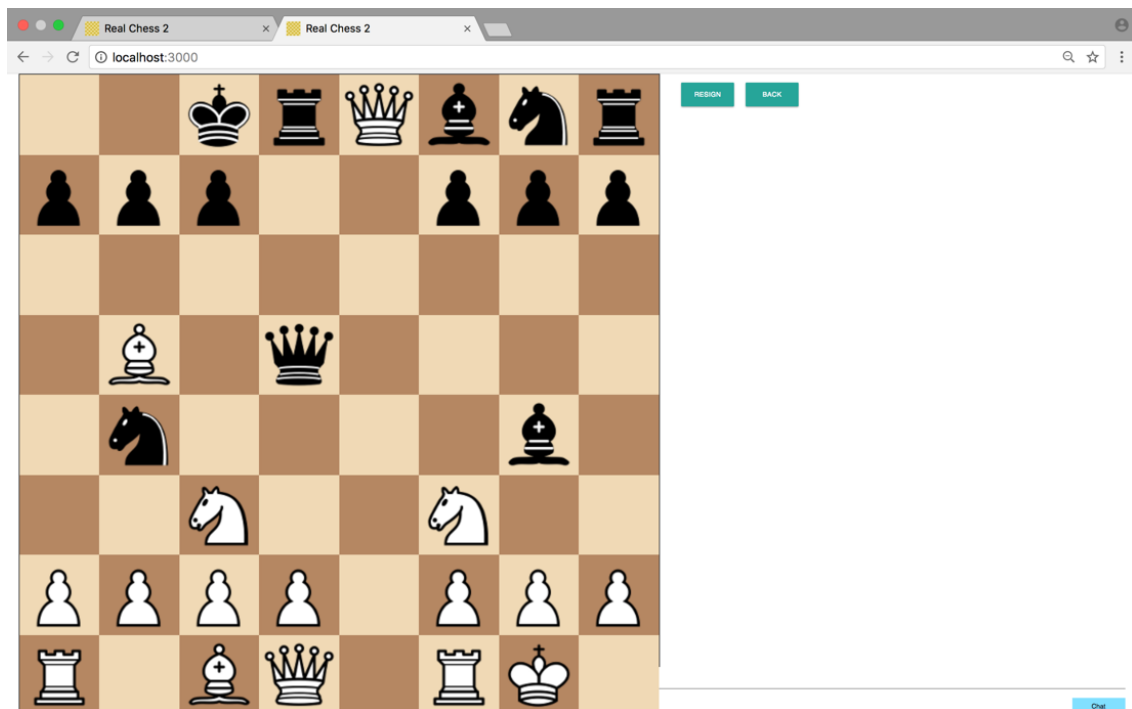


6.10) Test Case 20

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00M.10	Test Designed by: Lakshay
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Chessboard: Movement	Test Executed by: Nirut
Test Title: Promotion	Test Execution date: 19/3/2018
Description: Verify whether Promotion is possible	
Pre-conditions: User has valid username and password; A game should be started between 2 players	
Dependencies: Two players must be in a game	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Check active players page	Shows Lakshay online	User should be able to start a new game or be a spectator	As expected	YES R 5.11	Pass
2	Perform promotion of piece	Pawn in the chess game	Pawn gets promoted to any piece the user wants when it reaches the last row.	Result as expected		Pass

Post Condition: Promotion is performed

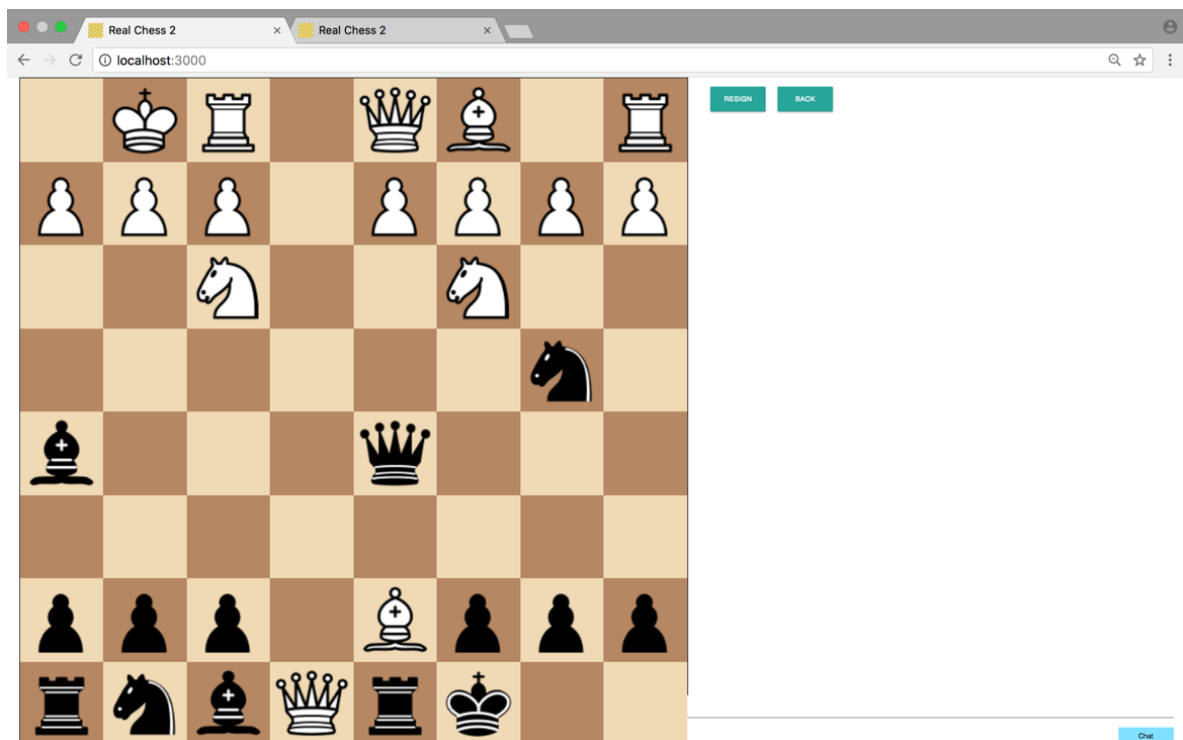


6.11) Test Case 21

Project Version: 1.0	
CHECKMATE	
Test Case ID: x00M.11	Test Designed by: Nirut
Test Priority (Low/Medium/High): Medium	Test Designed date: 19/3/2018
Module Name: Chessboard: Movement	Test Executed by: Parth
Test Title: Legality	Test Execution date: 19/3/2018
Description: Verify whether Legality is implemented	
Pre-conditions: User has valid username and password; A game should be started between 2 players	
Dependencies: Two players must be in a game	

Step	Test Steps	Test Data	Expected Result	Actual Result	Requirements Validated	Status (Pass/Fail)
1	Check active players page	Shows Lakshay online	User should be able to start a new game or be a spectator	As expected	YES, R 5.12	Pass
2	Check Legality	Any piece in the chess game	A move which doesn't stop the check on the king is an illegal move and should be illegal to perform	Result as expected		Pass

Post Condition: Legality is checked



User Manual

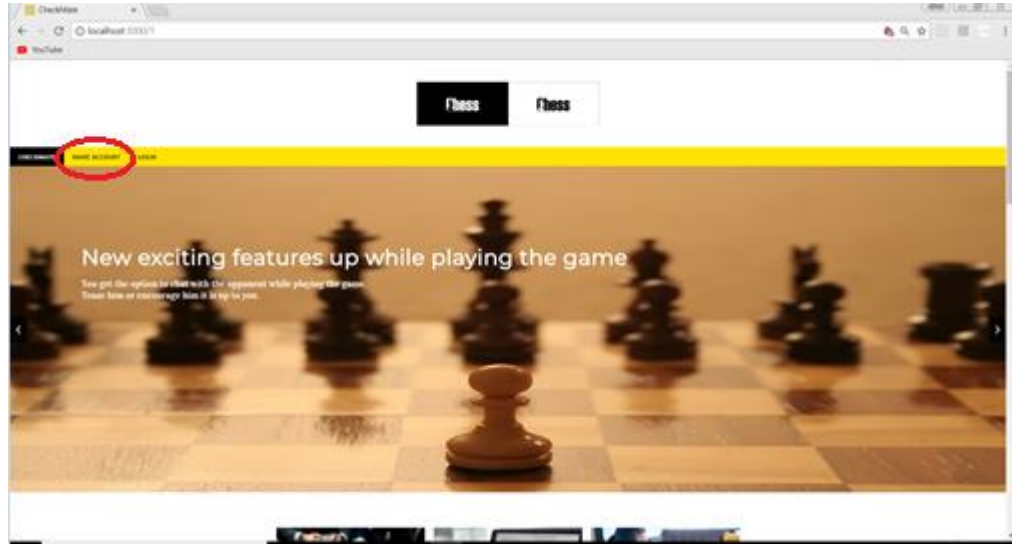
for

CHECKMATE

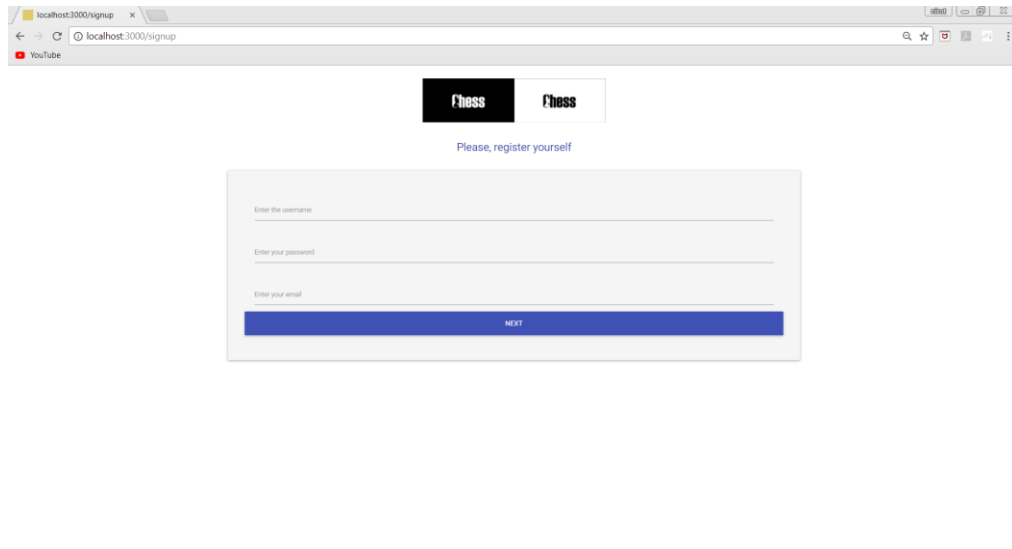
Version 1.0

How to Play a Live Chess Game?

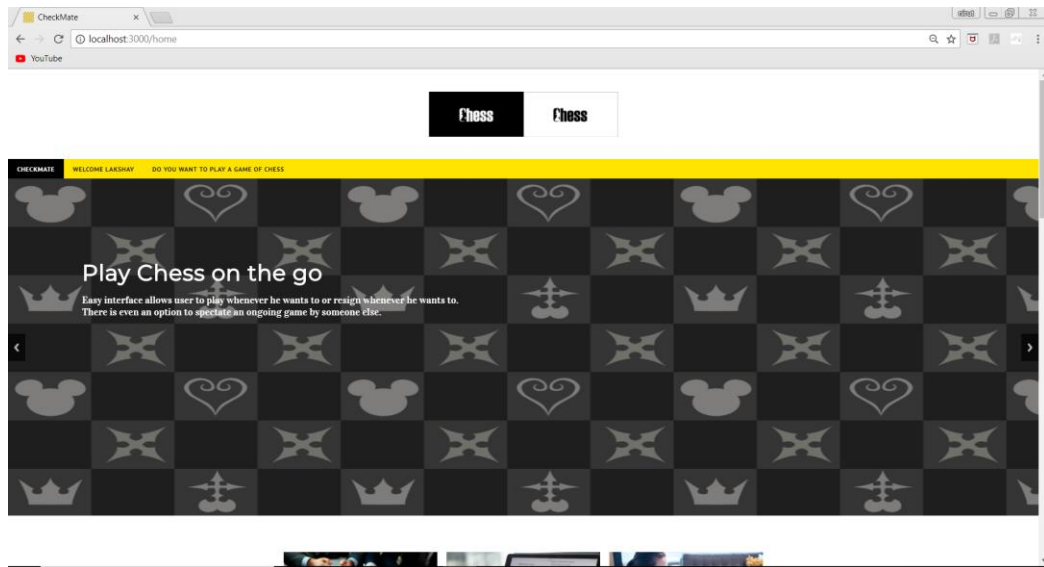
1. If you are a new user press make account button.



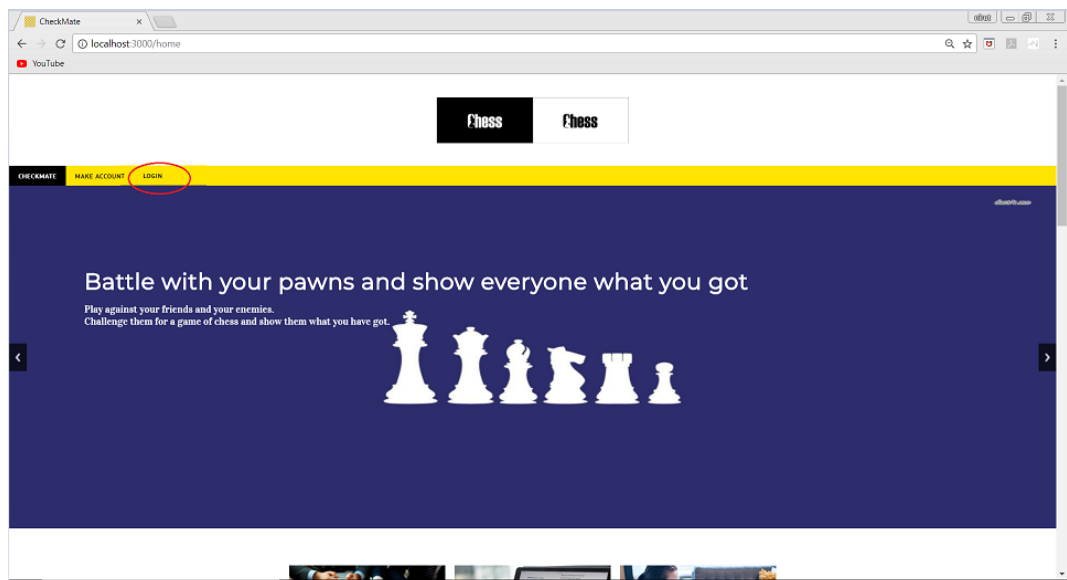
2. Enter your credentials and press next



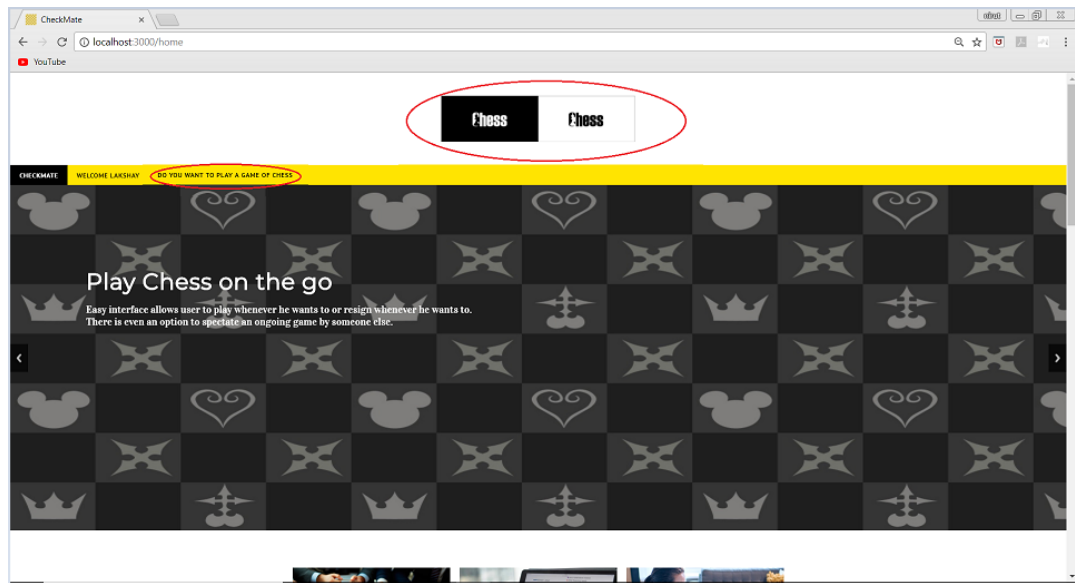
3. You will now enter to this screen



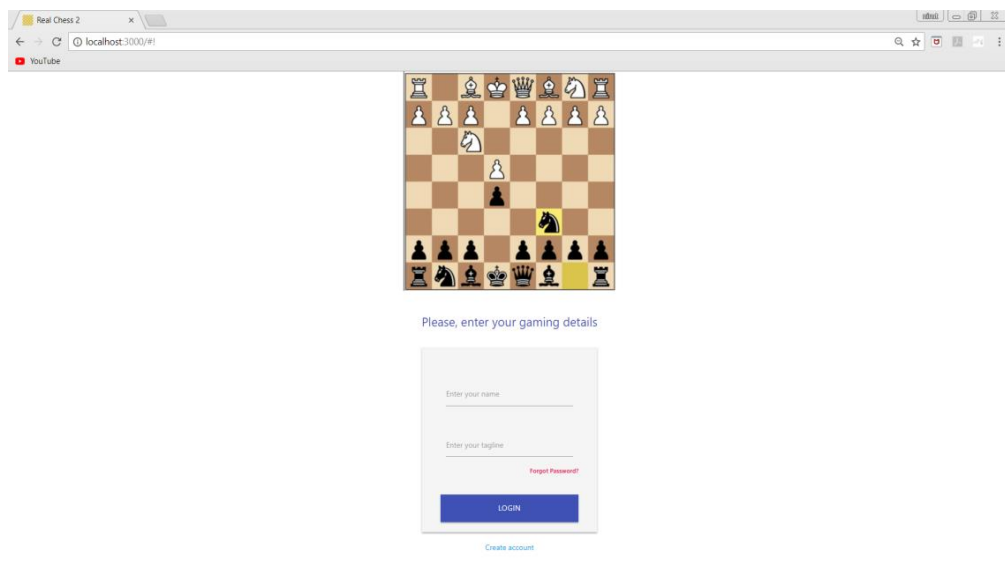
4. Else you can press login button if you already a registered user.



5. Press “Do you want to play a game of chess” button to play

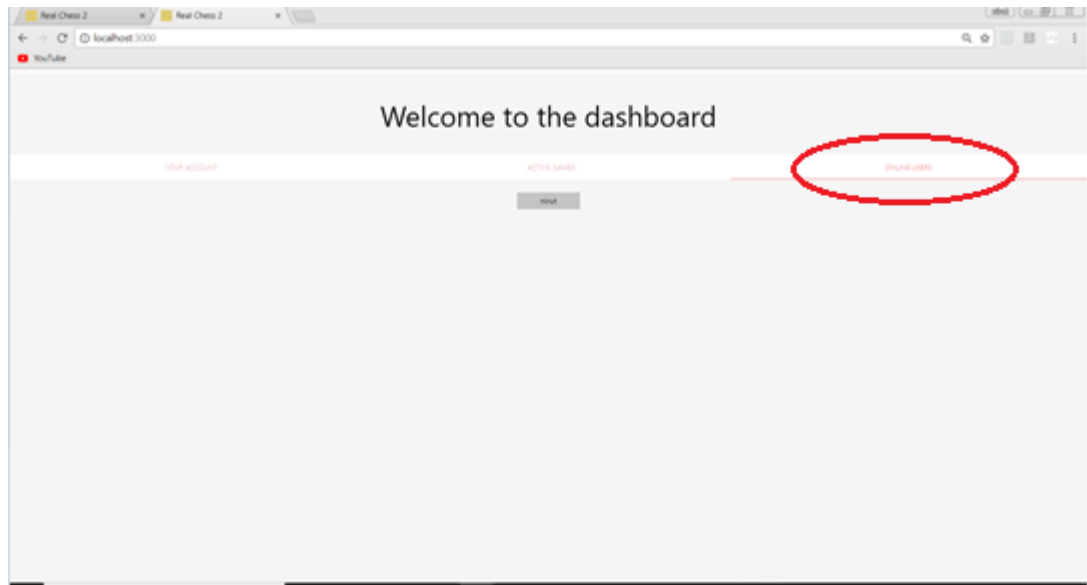


- Or you can press chess with black to play as a black and vice versa for the white one.
- The next screen displayed to you will look like the picture below.

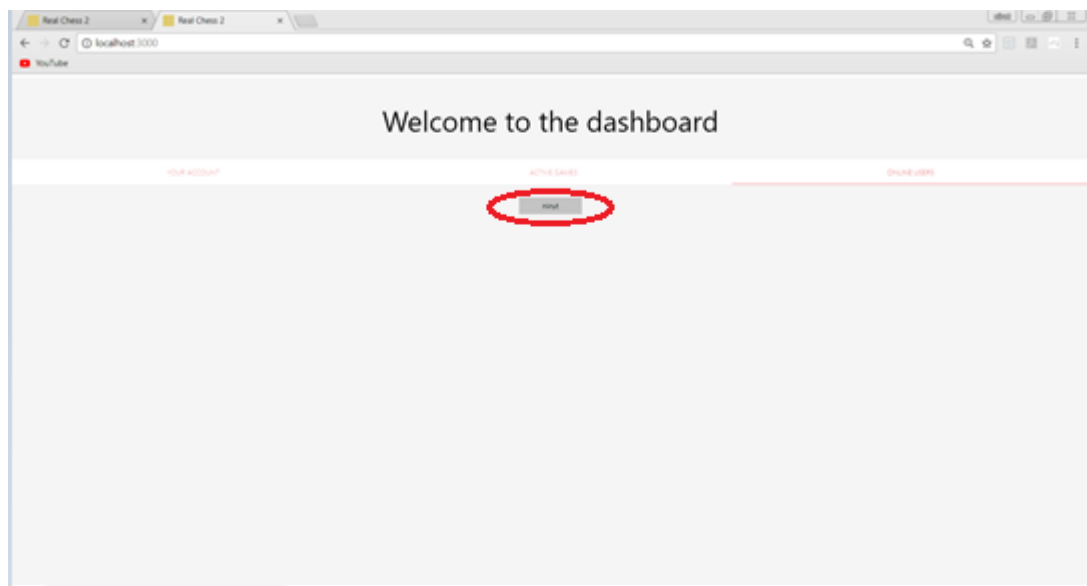


- Enter your name and tagline.
- Press login button.
- Get into the game.

6. On the home screen press online users to see who is online.



7. Then select the name of the user you want to play with.



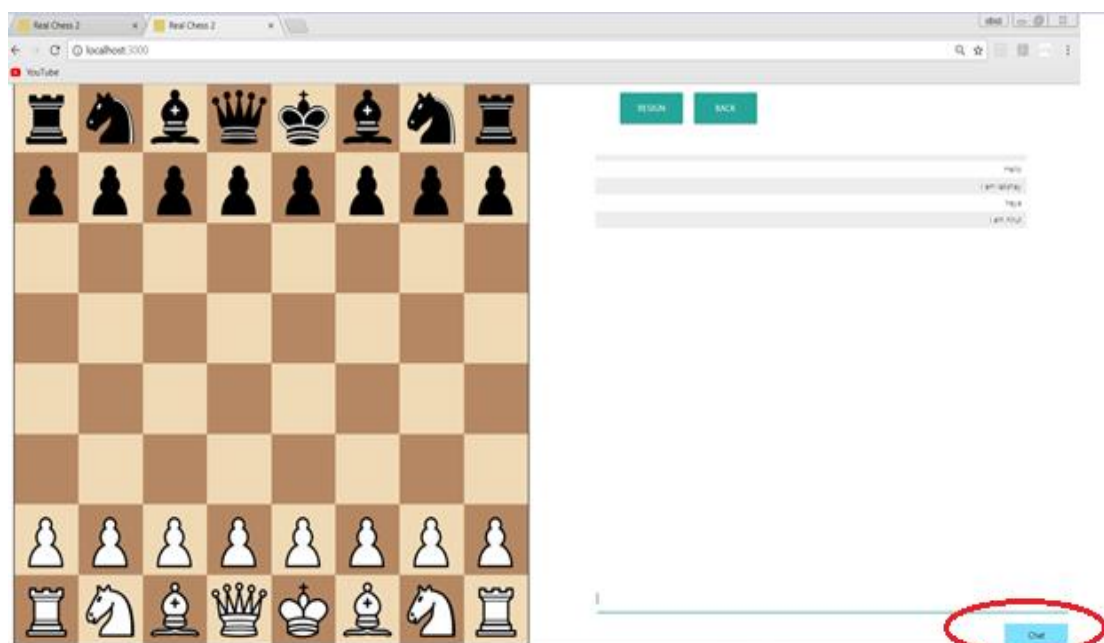
- For example, if I am interested to play with nirut I press nirut.
- Then the game will start between you and nirut.

8. Your opponent will be in the list of active users when they are online and not playing any game.
9. Once the screen has loaded you will be able to play the chess game with your opponent.

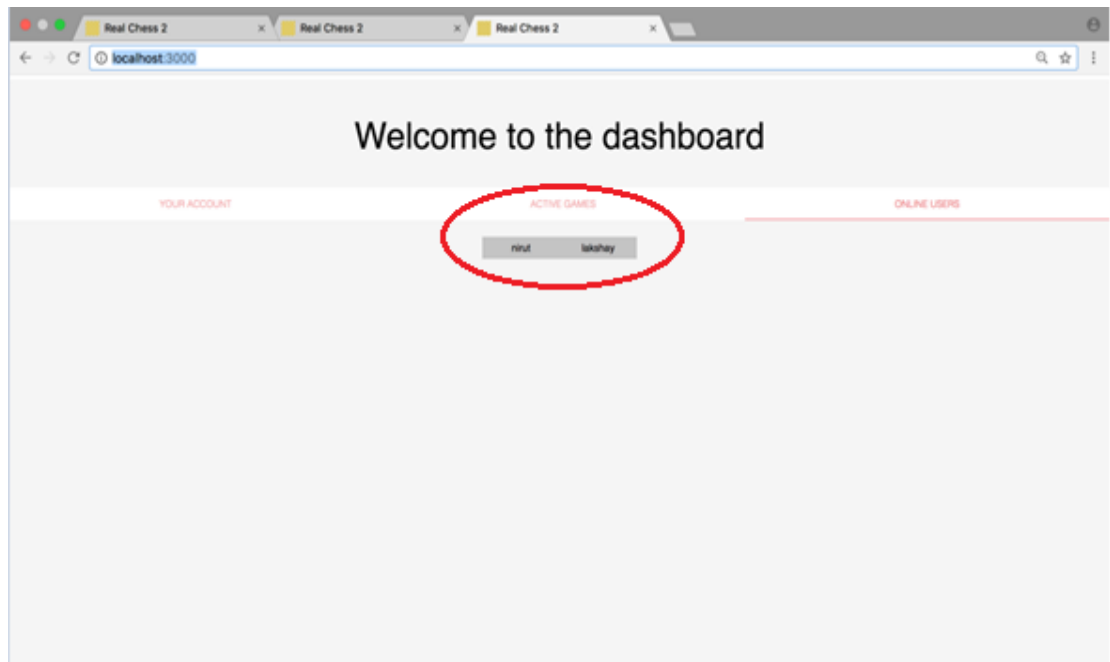


- This is how the page looks like.

10. You can also chat with the opponent by entering the message in the box and then pressing send button.



11. You can also go back and see active games.



- Have fun and remember nobody ever won a chess game by resigning.

CONCLUSION

Concluding the project, we were able to implement a real time web-based chess application using Sockets. Using sockets, we are able to minimize the time for the move to actually reach the opponent over the internet. The scope of this project is to make this website a base for chess players.

This project can also be extended in various ways. One of them is conduct tournaments in the site. Other features which can be added are creating a discussion forum within the site. Also, ratings can be given to players based on their wins and other factors. Tutorials can be provided to beginners by professionals. It can be one of the most interactive site where many people, keen on playing chess with others in the world visit.