Team Project Sprint #3

Team Name: Java4Fun
Team Members: Mohamad Abboud, Javkhlan Enkhjin, Brahian Ramon, Pratheek Reddy Patnam

I. Updated User Stories

ID	User Story Name	User Story Description	Priority	Estimated effort (hours)	Actual effort (if completed)	Status (completed, toDo, inProgress)	Developer names
1	Registration	As a new user, I want to be able to register for the checkers game.	High	1 day	Completed	Completed	Javkhlan Enkhjin, Brahian Ramon
2	Login	As an existing user, I need to log in so I can play the checkers game.	High	1 day	Completed	Completed	Javkhlan Enkhjin, Brahian Ramon
3	Display the board	As a player, after logging in, I need a 8*8 board grid with pieces so that I can start the game.	High	1 day	Completed	Completed	Mohamad Abboud, Pratheek Patnam
4	Start the game	As a player, when the board is displayed, I will click the start button to start the game.	High	1 day	Completed	Completed	Mohamad Abboud, Pratheek Patnam
5	Make a move 1	As a black piece player, I want to be able to move my piece diagonally forward to a valid empty cell so that I can make a move.	High	2 days	Completed	Completed	Javkhlan Enkhjin, Brahian Ramon, Mohamad Abboud, Pratheek Patnam
6	Make a move 2	As a white piece player, I want to be able to move my piece diagonally forward to a valid empty cell so that I can make a move.	High	2 days	Completed	Completed	Javkhlan Enkhjin, Brahian Ramon, Mohamad Abboud, Pratheek Patnam
7	Capture move 1	As a black piece player, I want to be able to move my piece diagonally forward to a valid empty cell so that I can capture a white piece.	High	1 day	Completed	Completed	Brahian Ramon, Mohamad Abboud

8	Capture move 2	As a white piece player, I want to be able to move my piece diagonally forward to a valid empty cell so that I can capture a black piece.	High	1 day	Completed	Completed	Brahian Ramon, Mohamad Abboud
9	Jump move 1	As a black piece player, after capturing a white piece, I should check if other white pieces are diagonally forward with the space beyond them empty so that I can continue to capture more white pieces.	Medium	2 days	Completed	Completed	Javkhlan Enkhjin, Pratheek Patnam
10	Jump move 2	As a white piece player, after capturing a black piece, I should check if other black pieces are diagonally forward with the space beyond them empty so that I can continue to capture more black pieces.	Medium	2 days	Completed	Completed	Javkhlan Enkhjin, Pratheek Patnam
11	King Piece 1	As a black/white piece player, I want my piece to become a king if it reaches the end of the board so that I can move both forwards and backwards diagonally.	Low	1 day	Completed	Completed	Javkhlan Enkhjin, Brahian Ramon, Mohamad Abboud, Pratheek Patnam
12	Game over	As a black or white piece player, I need to know if the game is over after each move.	Medium	1 day	Started	In progress	Javkhlan Enkhjin, Brahian Ramon, Mohamad Abboud, Pratheek Patnam
13	Game process	As a black or white piece player, I need to know the chance of winning.	Low	1 day	Started	Completed	Javkhlan Enkhjin, Brahian Ramon, Mohamad Abboud, Pratheek Patnam
14	Player side	As a black or white piece player, I need to know which pieces I am playing.	Low	1 day	Started	Completed	Javkhlan Enkhjin, Brahian Ramon,

							Mohamad Abboud, Pratheek Patnam
15	Computer VS Human	As a player, I want to play against the computer, because I don't want to play alone.	High	1 week	Started	Completed	Javkhlan Enkhjin, Brahian Ramon, Mohamad Abboud, Pratheek Patnam

II. Updated Acceptance Criteria (AC)

User Story ID and Name	AC ID	Description of Acceptance Criterion	Status (completed, toDo, in progress)	Developer Names
1	1.1	**AC 1.1** New user successfully registers **Given:** A new, valid username that does not exist. **When:** I register a new account with this username **AND** password is valid and re-entered correctly **AND** FirstName is valid **AND** LastName is valid **AND** Email is valid **AND** Phone number is valid **Then**: The new user is successfully registered	Completed	Javkhlan Enkhjin, Mohamad Abboud, Brahian Ramon, Pratheek Patnam
1	1.2	**AC 1.2** Unsuccessful registration due to username **Given:** A invalid username **When:** I register a new user with this username **AND** password is valid and re-entered correctly **AND** FirstName is valid **AND** LastName is valid **AND** Email is valid **AND** Phone number is valid	Completed	Javkhlan Enkhjin, Mohamad Abboud, Brahian Ramon, Pratheek Patnam

		Then: The user is not registered **AND** an error message is displayed		
1	1.3	**AC 1.3** Unsuccessful registration due to invalid password **Given:** An invalid password **When:** I register a new user with a valid username that does not exist **AND** password is re-entered correctly **AND** FirstName is valid **AND** LastName is valid **AND** Email is valid **AND** Phone number is valid **Then:** The user is not registered **AND** an error message is displayed	Completed	Javkhlan Enkhjin, Mohamad Abboud, Brahian Ramon, Pratheek Patnam
1	1.4	**AC 1.4** Unsuccessful registration due to invalid First Name **Given**: A invalid first name **When**: I register a new user with valid username that does not exist **AND** password is valid and re-entered correctly **AND** LastName is valid **AND** Email is valid **AND** Phone number is valid **Then**: The user is not registered **AND** an error message is displayed	Completed	Javkhlan Enkhjin, Mohamad Abboud, Brahian Ramon, Pratheek Patnam
1	1.5	**AC 1.5** Unsuccessful registration due to invalid last name **Given**: An invalid last name **When**: I register a new user with a valid username that doesn't exist **AND** password is valid and re-entered correctly **AND** FirstName is valid	Completed	Javkhlan Enkhjin, Mohamad Abboud, Brahian Ramon, Pratheek Patnam

		AND Email is valid **AND** Phone number is valid **Then**: The user is not registered **AND** an error message is displayed		
1	1.6	**AC 1.6** Unsuccessful registration due to invalid phone number **Given:** An invalid phone number **When**: I register a new user with a valid username that does not exist **AND** password is valid and re-entered correctly **AND** FirstName is valid **AND** LastName is valid **AND** Email is valid **Then**: The user is not registered **AND** an error message is displayed	Completed	Javkhlan Enkhjin, Mohamad Abboud, Brahian Ramon, Pratheek Patnam
1	1.7	**AC 1.7** Unsuccessful registration due to invalid email address **Given**: An invalid email address **When**: I register a new user with a valid username that does not exist **AND** password is valid and re-entered correctly **AND** FirstName is valid **AND** LastName is valid **AND** Phone number is valid **Then**: The user is not registered **AND** an error message is displayed	Completed	Javkhlan Enkhjin, Mohamad Abboud, Brahian Ramon, Pratheek Patnam
2	2.1	**AC 2.1** Successful Login **Given**: A valid username **When**: I login with this username	Completed	Brahian Ramon, Pratheek Patnam

		AND password is valid and is correct **Then**: the user is successfully logged in **And** I can play Checkers		
2	2.2	**AC 2.2** Unsuccessful login due to invalid username **Given**: An invalid username that does not exist **When**: I login with this username AND password is valid **Then:** The user is not logged in AND an error message is displayed	Completed	Brahian Ramon, Pratheek Patnam
2	2.3	**AC 2.3** Unsuccessful login due to a username that does not exist **Given:** A valid username that does not exist **When:** I login with this username **AND** password is valid **Then:** The user is not logged in **AND** an error message is displayed	Completed	Brahian Ramon, Pratheek Patnam
2	2.4	**AC 2.4** Unsuccessful login due to invalid password **Given:** An invalid password **When:** I login with this password AND username is valid and does exist **Then:** The user is not logged in **AND** an error message is displayed	Completed	Brahian Ramon, Pratheek Patnam
2	2.5	**AC 2.5** Unsuccessful login due to incorrect password **Given:** A valid password that is incorrect for the username **When:** I login with this password **AND** username is valid and does exist **Then:** The user is not logged in	Completed	Brahian Ramon, Pratheek Patnam

		AND an error message is displayed		
3	3.1	**AC 3.1:** Display Board **Given** a successful login **When** the player logs in. **Then** the board with pieces will be displayed to start the game.	Completed	Brahian Ramon, Pratheek Patnam
4	4.1	**AC 4.1:** Start the game **Given** the board is displayed **When** the player clicks the start button. **Then** the game will start. **AND** it is black piece's turn.	Completed	Brahian Ramon, Pratheek Patnam
5	5.1	**AC 5.1:** a valid black piece move on the board **Given** an ongoing game with black piece's turn **When** the black piece makes a move diagonally forward to a valid empty space **Then** the selected black piece will be moved to the selected cell by the user **And** the turn is changed to white piece.	Completed	Brahian Ramon, Pratheek Patnam
5	5.2	**AC 5.2:** a invalid black piece move on the board **Given** an ongoing game with black piece's turn **When** the black piece makes a move diagonally forward to a invalid space. **Then** the selected black piece will be remain in position **And** the turn will not change.	Completed	Brahian Ramon, Pratheek Patnam
5	5.3	**AC 5.3:** a invalid black piece move outside the board **Given** an ongoing game with black piece's turn	Completed	Brahian Ramon, Pratheek Patnam

		When the black piece makes a move outside the board **Then** the selected black piece will be remain in position **And** the turn will not change.		
6	6.1	**AC 6.1:** a valid white piece move on the board **Given** an ongoing game with white piece's turn **When** the white piece makes a move diagonally forward to a valid empty space **Then** the selected white piece will be moved to the selected cell by the user **And** the turn is changed to black pieces.	Completed	Javkhlan Enkhjin, Mohamad Abboud
6	6.2	**AC 6.2:** a invalid white piece move on the board **Given** an ongoing game with white piece's turn **When** the white piece makes a move diagonally forward to a invalid space. **Then** the selected white piece will be remain in position **And** the turn will not change.	Completed	Javkhlan Enkhjin, Mohamad Abboud
6	6.3	**AC 6.3:** a invalid white piece move outside the board **Given** an ongoing game with white piece's turn **When** the white piece makes a move outside the board **Then** the selected white piece will be remain in position **And** the turn will not change.	Completed	Javkhlan Enkhjin, Mohamad Abboud
7	7.1	**AC 7.1:** a white piece with a black piece diagonally next to it **Given** an ongoing game with white piece's turn **When** the white piece makes a move diagonally forward to a valid empty space beyond the black piece that is next to the selected white piece.	Completed	Javkhlan Enkhjin, Mohamad Abboud

		Then the selected white piece will be moved to the selected empty cell **And** the black piece will be captured **And** the turn is changed to black pieces.		
8	8.1	**AC 8.1:** a black piece with a white piece diagonally next to it **Given** an ongoing game with black piece's turn **When** the black piece makes a move diagonally forward to a valid empty space beyond the white piece that is next to the selected black piece. **Then** the selected black piece will be moved to the selected empty cell **And** the white piece will be captured **And** the turn is changed to white pieces.	Completed	Brahian Ramon, Pratheek Patnam
9	9.1	**AC 9.1:** a black piece player after capturing a white piece **Given** an ongoing game with black piece's turn **When** the black piece captured a white piece and moves to an empty cell where another white piece can be captured **Then** the selected black has to continue capturing until in no viable white pieces can be captured **And** all viable white pieces will be captured **And** the turn is changed to white pieces.	Completed	Brahian Ramon, Pratheek Patnam
10	10.1	**AC 10.1:** a white piece player after capturing a black piece **Given** an ongoing game with white piece's turn **When** the white piece captured a black piece and moves to an empty cell, if another black piece can be captured **Then** the selected white has to continue capturing until in no viable black pieces can be captured **And** all viable black pieces will be captured **And** the turn is changed to black pieces.	Completed	Javkhlan Enkhjin, Mohamad Abboud

11	11.1	**AC 11.1:** a black/white piece reaches the end of the board **Given** an ongoing game with black/white piece's turn **When** the black/white piece performs a valid move to reach the end of the board **Then** the piece will become a king piece **And** the piece will be able to move both diagonally forward and backwards	Completed	Javkhlan Enkhjin, Mohamad Abboud
12	12.1	**AC 12.1:** a win by black piece **Given** an ongoing game with black piece's turn **When** all the white pieces are captured **Then** the game is over. **And** black piece player wins the game.	Completed	Javkhlan Enkhjin, Mohamad Abboud
12	12.2	**AC 12.2:** a win by white piece **Given** an ongoing game with white piece's turn **When** all the black pieces are captured **Then** the game is over. **And** white piece player wins the game.	Completed	Javkhlan Enkhjin, Mohamad Abboud
12	12.3	**AC 12.3:** a win by white piece **Given** an ongoing game with black piece's turn **When** the black piece cannot make a valid move **Then** the game is over. **And** white piece player wins the game.	Completed	Brahian Ramon, Pratheek Patnam
12	12.4	**AC 12.4:** a win by black piece **Given** an ongoing game with white piece's turn **When** the white piece cannot make a valid move **Then** the game is over.	Completed	Brahian Ramon, Pratheek Patnam

		And black piece player wins the game.		
13	13.1	**AC 13.1:** a players can see their percentage of winning chance **Given** an ongoing game **When** an after or before making a move **Then** players can see their and opponents' chances of winning based on how many pieces they have.	Completed	Javkhlan Enkhjin, Mohamad Abboud
14	14.1	**AC 14.1:** as a players can see their username **Given** an ongoing game **When** for all duration of playing **Then** players can see their username on the side which they are playing.	Completed	Javkhlan Enkhjin, Mohamad Abboud
15	15.1	**AC 15.1:** Start the game human vs computer **Given** the board is displayed **When** the player clicks the start button. **Then** the game will start **AND** the black piece's will be controlled by the human **AND** the white piece's will be controlled by the Computer.. **AND** it is black piece's turn.	Completed	Mohamad Abboud, Javkhlan Enkhjin Brahian Ramon, Pratheek Patnam
	15.2	**AC 15.2:** Computer makes a move **Given** an ongoing game **When** the turn switches to white pieces. **Then** the computer will select a valid white piece and move it to a valid empty space on the board **AND** it is black piece's turn	Completed	Mohamad Abboud, Javkhlan Enkhjin Brahian Ramon, Pratheek Patnam
	15.3	**AC 15.3:** Computer has an opportunity to capture a black piece **Given** an ongoing game and it is white piece's turn	Completed	Mohamad Abboud, Javkhlan Enkhjin

		When the computer will select a valid white piece and move it to a valid empty space diagonally beyond a black piece on the board **Then** the black piece will be captured **AND** it is black piece's turn		Brahian Ramon, Pratheek Patnam
	15.4	**AC 15.4:** Computer has an opportunity to jump after capturing a black piece **Given** an ongoing game and it is white piece's turn **When** the computer has already captured a black piece and another black piece can be captured. **Then** the turn will not be changed **AND** the computer has to continue to capture the other black piece using the same white piece used to capture the first black piece.	Completed	Mohamad Abboud, Javkhlan Enkhjin Brahian Ramon, Pratheek Patnam
	15.5	**AC 15.5:** Computer piece reaches the last row. **Given** an ongoing game and it is white piece's turn **When** the computer moves a white piece to the last row on the board. **Then** the white piece will become a king piece **AND** the turn will be changed to black piece.	Completed	Mohamad Abboud, Javkhlan Enkhjin Brahian Ramon, Pratheek Patnam

III. Updated Implementation Tasks

Include the tasks from the previous report and highlight the new tasks with a different color.

Summary of production code

User Story ID and Name	AC ID	Class Name(s)	Method Name(s)	Developer Name(s)	Status	Notes (optional)
1	1.1	LoginController	registerNewAccount()	Javkhlan Enkhjin	Completed	
	1.2	LoginController	registerationUsername()	Javkhlan Enkhjin	Completed	
	1.3	LoginController	registerationPassword()	Javkhlan Enkhjin	Completed	
	1.4	LoginController	registerationFName()	Javkhlan Enkhjin	Completed	
	1.5	LoginController	registerationLName()	Javkhlan Enkhjin	Completed	
	1.6	LoginController	registerationPhonenumber()	Javkhlan Enkhjin	Completed	

	1.7	LoginController	registerationEmail()	Javkhlan Enkhjin	Completed	
2	2.1	LoginController	actionOnClickLoginButton()	Javkhlan Enkhjin	Completed	
	2.2	LoginController	actionOnClickLoginButton()	Javkhlan Enkhjin	Completed	
	2.3	LoginController	actionOnClickLoginButton()	Javkhlan Enkhjin	Completed	
	2.4	LoginController	loginPassword()	Javkhlan Enkhjin	Completed	
	2.5	LoginController	actionOnClickLoginButton()	Javkhlan Enkhjin	Completed	
3	3.1	CheckerGUI	InitializeBoard()	Brahian Ramon, Pratheek Patnam	Completed	
4	4.1	CheckerGUI	InitializeBoard()	Brahian Ramon, Pratheek Patnam	Completed	
5	5.1	CheckerGUI	UpdateGame()	Mohamad Abboud, Javkhlan Enkhjin	Completed	
	5.2	CheckerGUI	UpdateGame()	Mohamad Abboud, Javkhlan Enkhjin	Completed	
	5.3	CheckerGUI	UpdateGame()	Mohamad Abboud, Javkhlan Enkhjin	Completed	
6	6.1	CheckerGUI	UpdateGame()	Mohamad Abboud, Javkhlan Enkhjin	Completed	
	6.2	CheckerGUI	UpdateGame()	Mohamad Abboud, Javkhlan Enkhjin	Completed	
	6.3	CheckerGUI	UpdateGame()	Mohamad Abboud, Javkhlan Enkhjin	Completed	
7	7.1	CheckerGUI	DisplayPossibleMoves()	Mohamad Abboud, Javkhlan Enkhjin	Completed	
8	8.1	CheckerGUI	DisplayPossibleMoves()	Mohamad Abboud,	Completed	

				Javkhlan Enkhjin		
9	9.1	CheckerGUI	PerfomOvertake()	Mohamad Abboud, Javkhlan Enkhjin	Completed	
10	10.1	CheckerGUI	PerfomOvertake()	Mohamad Abboud, Javkhlan Enkhjin	Completed	
11	11.1	CheckerGUI	DoMove()	Mohamad Abboud, Javkhlan Enkhjin	Completed	
12	12.1	CheckerGUI	GameState()	Brahian Ramon, Pratheek Patnam	Completed	
	12.2	CheckerGUI	GameState()	Brahian Ramon, Pratheek Patnam	Completed	
	12.3	CheckerGUI	GameState()	Brahian Ramon, Pratheek Patnam	Completed	
	12.4	CheckerGUI	GameState()	Brahian Ramon, Pratheek Patnam	Completed	
13	13.1	CheckerGUI	PerfomOvertake()	Mohamad Abboud, Javkhlan Enkhjin	Completed	
14	14.1	CheckerGUI	playerNames()	Mohamad Abboud, Javkhlan Enkhjin	Completed	
15	15.1	CheckerGUIComputer	InitializeBoard()	Mohamad Abboud, Javkhlan Enkhjin Brahian Ramon, Pratheek Patnam	Completed	
	15.2	CheckerGUIComputer	UpdateGameComputer()	Mohamad Abboud, Javkhlan Enkhjin Brahian Ramon,	Completed	

				Pratheek Patnam		
	15.3	CheckerGUIComputer	UpdateGameComputer()	Mohamad Abboud, Javkhlan Enkhjin Brahian Ramon, Pratheek Patnam	Completed	
	15.4	CheckerGUIComputer	UpdateGameComputer()	Mohamad Abboud, Javkhlan Enkhjin Brahian Ramon, Pratheek Patnam	Completed	
	15.5	CheckerGUIComputer	CheckComputerPieceOutcome(Node piece)	Mohamad Abboud, Javkhlan Enkhjin Brahian Ramon, Pratheek Patnam	Completed	

Summary of automated test code (directly corresponding to some acceptance criteria)

User Story ID and Name	**Acceptance Criterion ID**	**Class Name (s) of the Test Code**	**Method Name(s) of the Test Code**	**Description of the Test Case (input & expected output)**	**Status**	**Developer Name(s)**
1	1.1	RegisterTest	SuccessfulRegistration	After successfully validating the registration form and making sure that all the input is validated, the user will be registered in the database and can be used to login.	Done	Javkhlan Enkhjin, Brahian Ramon
	1.2	RegisterTest	UnsuccessfulRegistration	Unsuccessful registration due to an invalid username	Done	Javkhlan Enkhjin, Brahian Ramon
	1.3	RegisterTest	UnsuccessfulRegistrationPassword	Unsuccessful registration due to an invalid password	Done	Javkhlan Enkhjin, Brahian Ramon
	1.4	RegisterTest	UnsuccessfulRegistrationFName	Unsuccessful registration due to an invalid first name	Done	Javkhlan Enkhjin, Brahian Ramon
	1.5	RegisterTest	UnsuccessfulRegistrationLName	Unsuccessful registration due to an invalid last name	Done	Javkhlan Enkhjin,

						Brahian Ramon
	1.6	RegisterTest	UnsuccessfulRegistrationPhonenumber	Unsuccessful registration due to an invalid phone number	Done	Javkhlan Enkhjin, Brahian Ramon
	1.7	RegisterTest	UnsuccessfulRegistrationEmail	Unsuccessful registration due to an invalid email	Done	Javkhlan Enkhjin, Brahian Ramon
2	2.1	LoginTest	SuccessfulTestLogin	Successfully logging into the application using a valid username and password	Done	Javkhlan Enkhjin, Brahian Ramon
	2.2	LoginTest	UnsuccessfulTestLogin	Unsuccessful login due to an invalid username	Done	Javkhlan Enkhjin, Brahian Ramon
	2.3	LoginController	actionOnClickLoginButton1	Unsuccessful login due to an non existing username	Done	Javkhlan Enkhjin, Brahian Ramon
	2.4	LoginTest	UnsuccessfulTestLoginPassword	Unsuccessful login due to an invalid password	Done	Javkhlan Enkhjin, Brahian Ramon
	2.5	LoginController	actionOnClickLoginButton1	Unsuccessful login due to an incorrect password to an already existing username	Done	Javkhlan Enkhjin, Brahian Ramon
3	3.1	CheckersBoardTest	testBoardContainsPieces()	I go over the board to make sure that each piece is found on the board in the beginning.	Done. Note: works with JUNIT5 when code was merged with teammates project it fails since it only works with Junit5	Mohamad Abboud, Pratheek Patnam
	3.1	CheckersBoardTest	testValidRow()	Testing a valid row where a piece can be placed on a particular row on the board. 1 means a piece is there, 0 means the cell is empty.	Done. Note: works with JUNIT5 when code was merged with	Mohamad Abboud, Pratheek Patnam

					teammates project it fails since it only works with Junit5	
	3.1	CheckersBoardTest	testValidColumn()	Same as the row, but testing a valid column where a piece can be placed on a particular column on the board. 1 means a piece is there, 0 means the cell is empty.	Done. Note: works with JUNIT5 when code was merged with teammates project it fails since it only works with Junit5	Mohamad Abboud, Pratheek Patnam
	3.1	CheckersBoardTest	testInvalidRow()	Testing an invalid row where a piece cannot be moved to, invalid rows will be represented as -1.	Done. Note: works with JUNIT5 when code was merged with teammates project it fails since it only works with Junit5	Mohamad Abboud, Pratheek Patnam
	3.1	CheckersBoardTest	testInvalidColumn()	Testing an invalid column where a piece cannot be moved to, invalid columns will be represented as -1.	Done. Note: works with JUNIT5 when code was merged with teammates project it fails since it only works with Junit5	Mohamad Abboud, Pratheek Patnam
5	5.1	NewCheckersBoardTest	testValidBlackPieceMove	Testing that a black piece can move to a	Completed	Javkhlan Enkhjin,

				valid empty cell diagonally forward		Brahian Ramon
	5.2	NewCheckersBoardTest	testInvalidNonEmptyCellBlackPieceMove	Testing that a black piece cannot move to a non empty cell	Completed	Javkhlan Enkhjin, Brahian Ramon
	5.3	NewCheckersBoardTest	testInvalidOutOfBoundsCellBlackPieceMove	Testing that a black piece cannot move to a cell that is out of bounds	Completed	Javkhlan Enkhjin, Brahian Ramon
6	6.1	NewCheckersBoardTest	testValidWhitePieceMove	Testing that a white piece can move to a valid empty cell diagonally forward	Completed	Mohamad Abboud, Pratheek Patnam
	6.2	NewCheckersBoardTest	testInvalidNonEmptyCellWhitePieceMove	Testing that a white piece cannot move to a non empty cell	Completed	Mohamad Abboud, Pratheek Patnam
	6.3	NewCheckersBoardTest	testInvalidOutOfBoundsCellWhitePieceMove	Testing that a white piece cannot move to a cell that is out of bounds	Completed	Mohamad Abboud, Pratheek Patnam

Summary of manual test cases (directly corresponding to some acceptance criteria)\

User Story ID and Name	**Acceptance Criterion ID**	**Test Case Input**	**Test Oracle (Expected Output)**	**Status**	**Notes**	**Developer Name(s)**
4	4.1	The user will click the start button	The game will start, and the black piece player can make the first move	Completed	-	Mohamad Abboud, Javkhlan Enkhjin
7	7.1	A white piece user will have a piece with a black piece positioned diagonally forward to it, the white piece user will click on his/her piece and click on the empty square that is diagonally adjacent to the black piece.	The black piece will be captured by the white and removed from the board, then the turn is changed to black piece	Completed	-	Brahian Ramon, Pratheek Patnam
8	8.1	A black piece user will have a piece with a white piece positioned diagonally forward to it, the black piece user will click on the piece and click on the empty square that is diagonally adjacent to the white piece.	The white piece will be captured by the black piece and removed from the board, then the turn is changed to white piece.	Completed	-	Brahian Ramon, Pratheek Patnam

9	9.1	After a black piece has captured a white piece and there are more viable white pieces that can be captured.	The black piece user has to jump and capture all viable white pieces on the board, then when all the pieces that can be captured are captured, the captured pieces will be removed from the board then the turn will change to white piece.	Completed	-	Brahian Ramon, Pratheek Patnam
10	10.1	After a white piece has captured a black piece there are more viable white pieces that can be captured.	The white piece user has to jump and capture all viable black pieces on the board, then when all the pieces that can be captured are captured, the captured pieces will be removed from the board then the turn will change to black piece.	Completed	-	Mohamad Abboud, Javkhlan Enkhjin
11	11.1	When a white or black piece enters the end of the board in their respective turns	The white or black piece will turn into a King white or black piece, which will be able to move both diagonally forward and backward	Completed	-	Mohamad Abboud, Javkhlan Enkhjin
12	12.3	When it is the black piece's turn and there are no more possible moves for the black pieces to move	Black piece player will forfeit the game and white piece player will win	Completed	-	Brahian Ramon, Pratheek Patnam
12	12.4	When it is the white piece's turn and there are no more possible moves for the white pieces to move	White piece player will forfeit the game and white piece player will win	Completed	-	Brahian Ramon, Pratheek Patnam
13	13.1	When a black/whites piece's captured by the opponent.	The percentage of the white or black piece's chances to win decreases. (the percentage will decrease in value)	Completed	-	Mohamad Abboud, Javkhlan Enkhjin
14	14.1	When both users log in successfully.	Player 1's username will be displayed at the bottom right of the user interface and the Player 2's	Completed	-	Mohamad Abboud, Javkhlan Enkhjin

			username at the top right of the user interface			
15	15.1	When the human user clicks on the computer radio button and logs in successfully.	The checkers board will appear with the black and white pieces and the game will start once the user clicks the start game button. The user will control the black pieces and the computer will control the white pieces.	Completed	-	Mohamad Abboud, Javkhlan Enkhjin Brahian Ramon, Pratheek Patnam
	15.2	When the game is ongoing and it is white piece's turn (computer's turn)	The computer will choose a valid piece at random and move it to a valid empty square on the board	Completed	-	Mohamad Abboud, Javkhlan Enkhjin Brahian Ramon, Pratheek Patnam
	15.3	When the game is ongoing and it is white piece's turn (computer's turn)	The computer will choose a valid piece that is able to capture a black piece and then move the white piece to a valid empty square beyond that black piece to capture it.	Completed	-	Mohamad Abboud, Javkhlan Enkhjin Brahian Ramon, Pratheek Patnam
	15.4	When the game is ongoing and it is white piece's turn (computer's turn) and the white piece has just captured a black piece	The computer will check if there is another piece that is viable to be captured, if so the computer will jump again and check again if there is another possible jump move, if not the turn will switch to black piece after the capture and jump is completed	Completed	-	Mohamad Abboud, Javkhlan Enkhjin Brahian Ramon, Pratheek Patnam
	15.5	When the game is ongoing and it is white piece's turn (computer's turn)	When the computer moves a white piece to the last row on the board the white piece will become a king piece and change shape to have a red circle inside of it, indicating that it is a king piece. That particular king piece will be able to move and capture both	Completed	-	Mohamad Abboud, Javkhlan Enkhjin Brahian Ramon, Pratheek Patnam

			diagonally forward and backward.			

Summary of other automated or manual tests (not corresponding to the acceptance criteria)

Number	**Test Input**	**Expected Result**	**Class Name of the Test Code**	**Method Name of the Test Code**	**Status**	**Developer Name(s)**
1	1.1	Displaying the Login Page GUI	The login page will be displayed and the user will able to login or register to start a game	Completed	-	Javkhlan Enkhjin
4	4.1	Displaying the board GUI with pieces on the board.	The checkers board is displayed to the user with the white and black pieces that will be used to play the game.	Completed	-	Mohamad Abboud

IV. Design Documentation

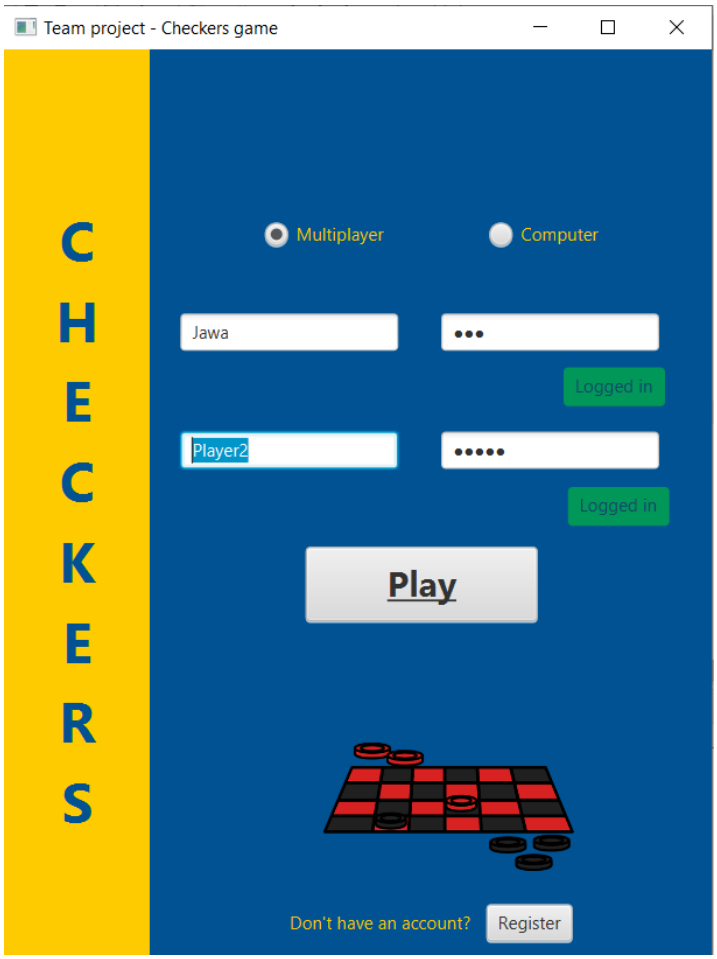
1. User Interface Design

List the names of the team members who contributed to this section.
Summarize the user-interface design, using a combination of screenshots and textual descriptions.

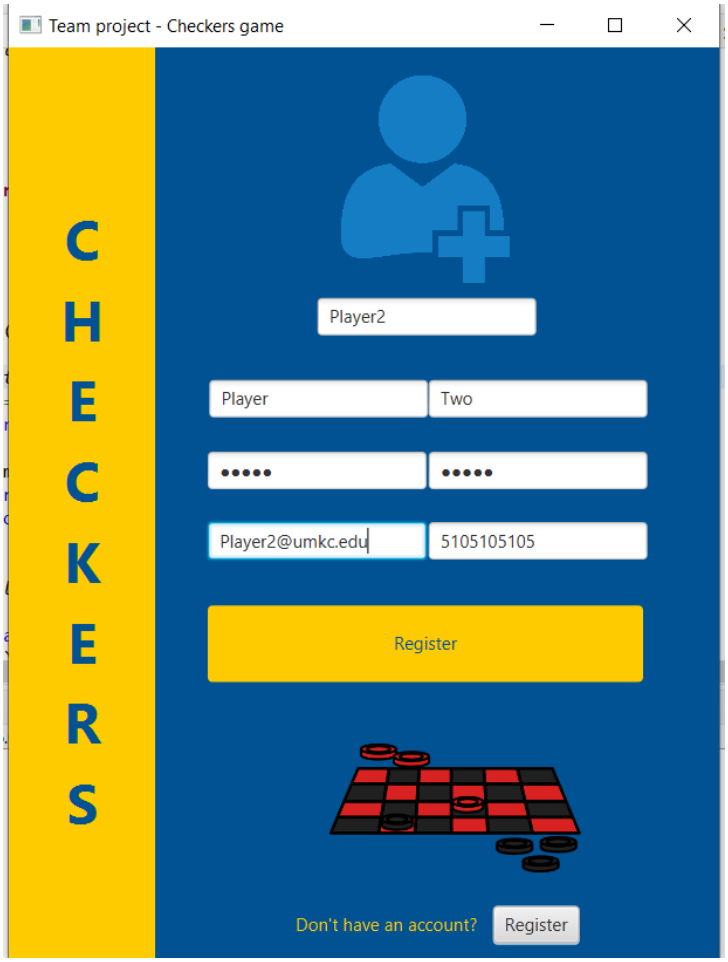
Contributors: Mohamad Abboud, Javkhlan Enkhjin, Brahian Ramon, Pratheek Reddy Patnam

Our checkers app is made up of 3 main pages, Login page, Register page, Checkers Game page.

The first is the login page, which contains 2 radio buttons and text fields, the radio buttons represent whether the user would like to play against a human or against the computer. If the human radio button is clicked, the user will be prompted with 2 login sections, one for the first player and another for the second player. Both logins have to be authenticated before the login can be successful. If the computer radio button is clicked the user will be prompted with only 1 login section indicating that there won't be a second human player and that the user will play against the computer. The user will then be authenticated and will login successfully to start the game.

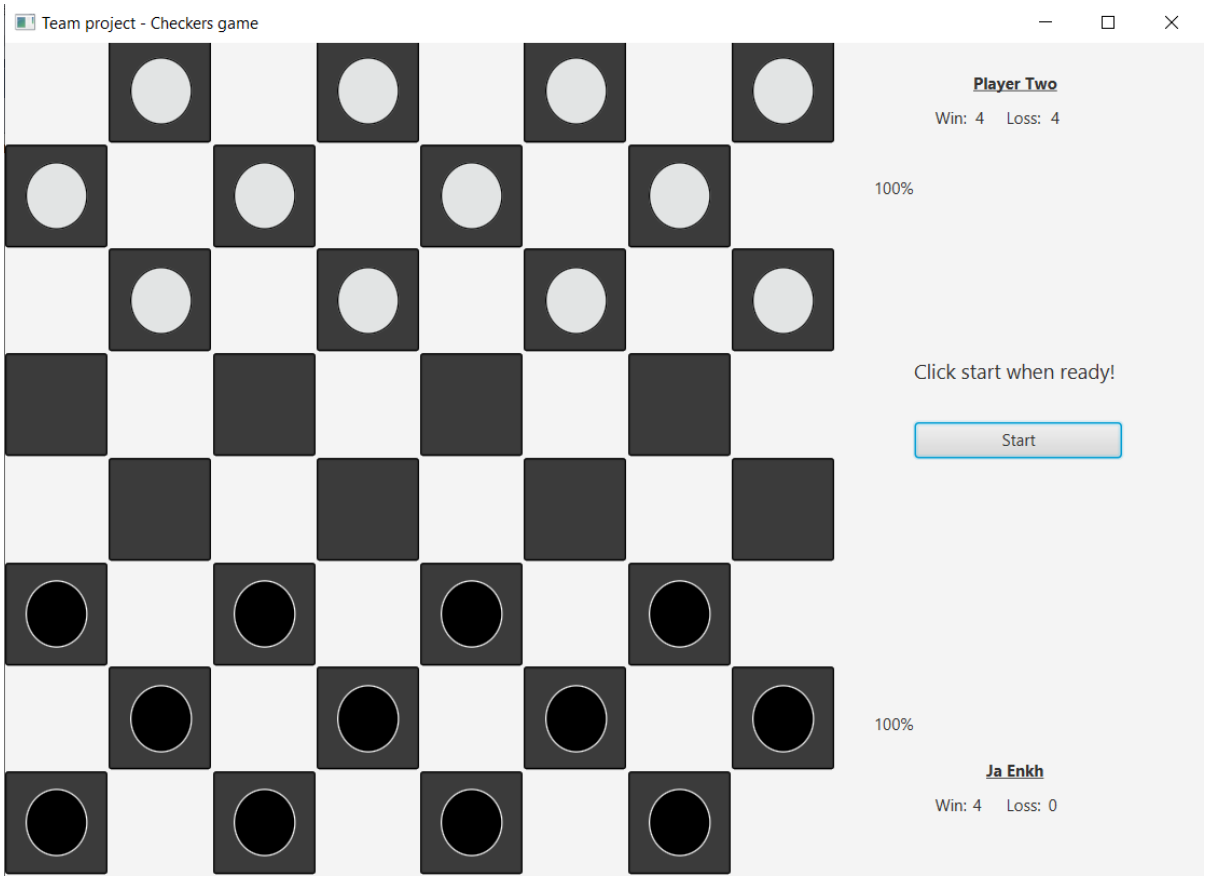


The second page is the register page, the user will have to enter a username, password, first, last name, phone number and email that they will use in order to login into the Checkers game. For user management, we store users password after an encrypted sha2 algorithm(hashbyte function).



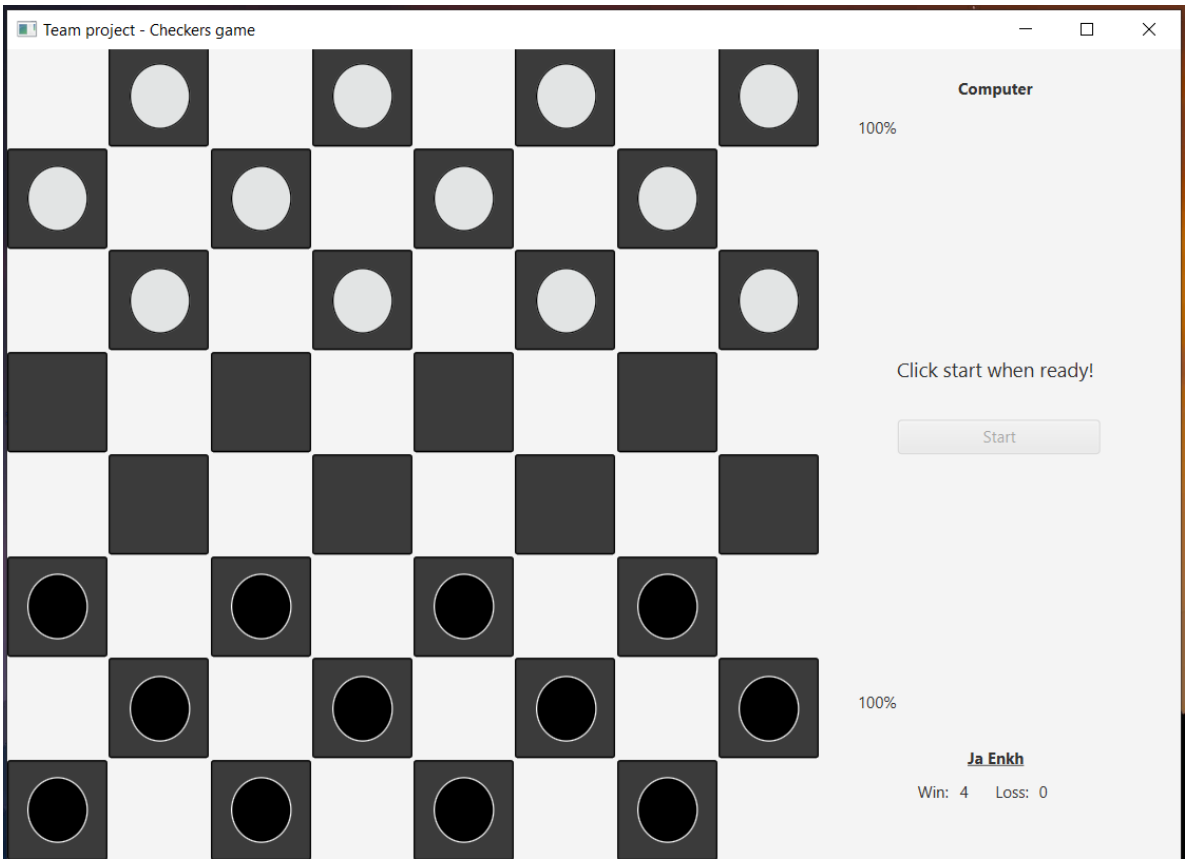
The third page is the Checkers game page. This page contains the user information section to the right and the checkers board to the left. The user information contains the names of both users respectively depending on which side they are on, also along with the name there is a win percentage that will let the user know how good their odds are to win. In the middle of the user information section there is a start game button that is used to start the game.

The game has 2 different states, playing against a human or computer.



vs Human: Once the start game button is clicked. The turn will start with black pieces, the black piece player will be able to start the game by clicking on any piece that can perform a legal move and then click on an empty square on the board. If the empty square click is a valid square the black piece will be moved to that square. The turn then will be switched to white piece and the process repeats. If there is an opportunity for either white or black piece player to capture an opponent, the black or white piece player will select the viable checker's piece and then place it diagonally beyond the opponent piece which will capture the opponent piece. After the capture is complete, if there is an opportunity to capture another opponent piece using the piece that was used earlier, the turn will not be changed and the current turn piece player will have to continue to capture all possible opponent pieces to end the turn.

The game can end in two ways, either black piece wins or a white piece wins in 2 situations, one side will win if the other side does not have any more pieces available or the other side cannot make any moves in their respective turn.

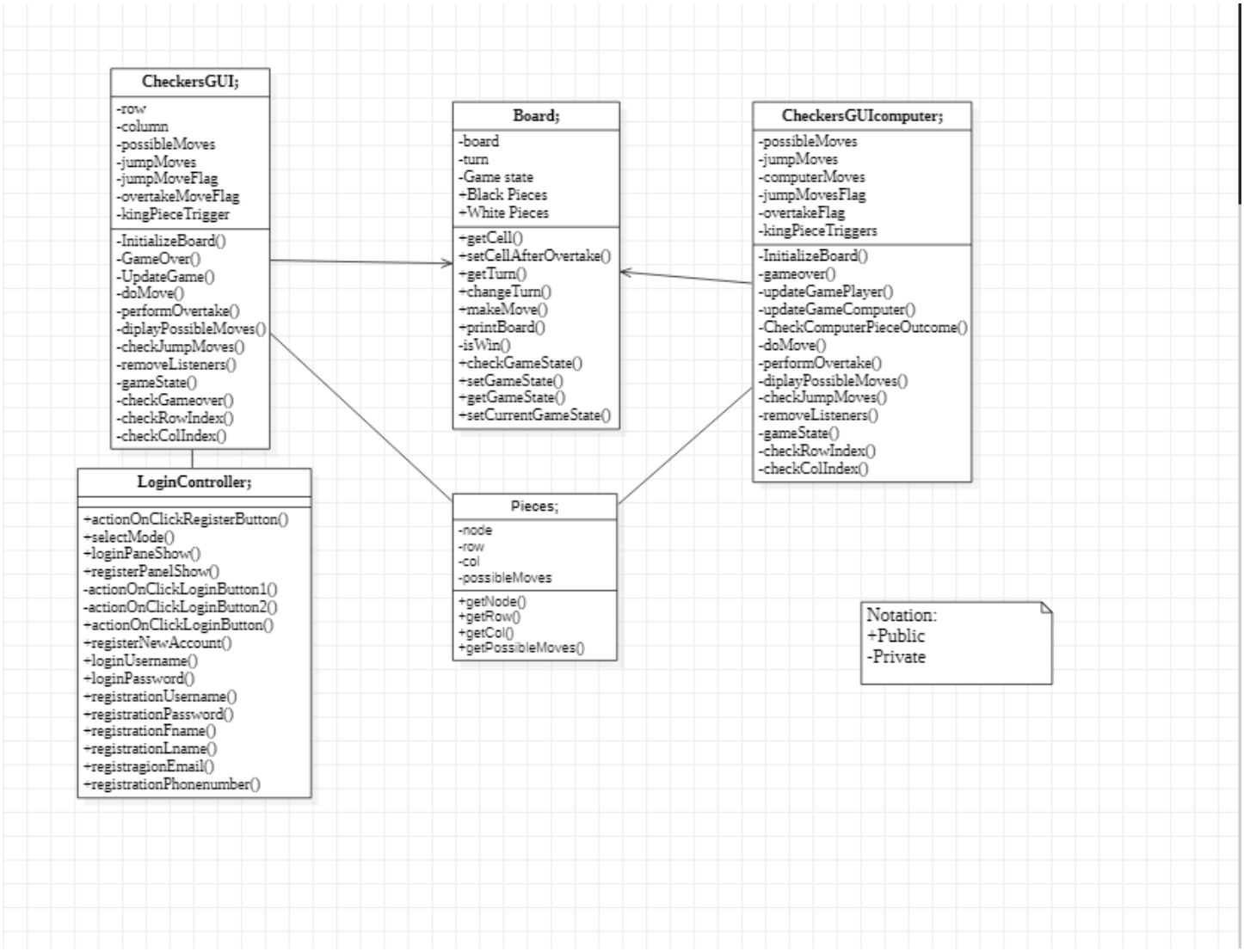


vs Computer: Similarly as the vs Human state, the game will start once the start game button is clicked. The turn will start with black pieces who are the human player, the black piece player will make a move and the turn will switch to white pieces who are controlled by the computer and the game is underway.

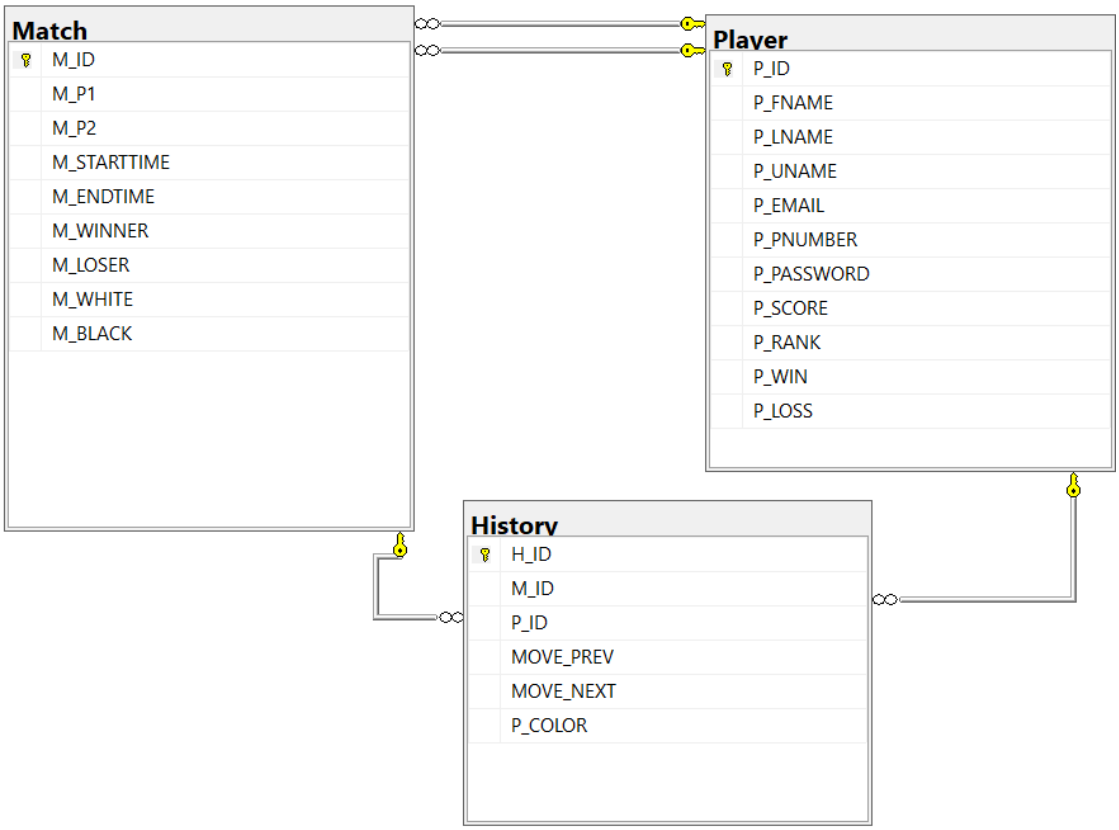
2. Software Architecture

List the names of the team members who contributed to this section.
Provide a class diagram that captures the main classes and their relationships in your final program.

Contributors: Mohamad Abboud, Javkhlan Enkhjin, Brahian Ramon, Pratheek Reddy Patnam



- Database Design:



For database design, we use Player table for user information and login credential, P_ID (primary key), P_FNAME - first name, P_LNAME - last name, P_UNAME - user name (login credential), P_EMAIL - email, P_PNUMBER - phone number, P_PASSWORD - password (login credential), P_SCORE - earned score (further development), P_RANK - user rank (further development), P_WIN - user win record and P_LOSS - user loss record. For further development, both tables match and history would be used to show who made which move when, this is to show a record/history of each match, also it would show play-by-play of each player.

3. Algorithm Design

List the names of the team members who contributed to this section.
Describe the algorithm design of the computer opponent (e.g., using pseudo code). The description should be understandable without referring to the source code.

Contributors: Mohamad Abboud, Javkhlan Enkhjin

At the start of the computer's turn, a function was built to traverse the white pieces which will determine if there exists a piece that can perform a move or overtake. If not then the computer must forfeit and the human player will win. However, if there is one move or many moves, they will be sorted into 2 variables, the overtake moves or the possible moves. The algorithm will prioritize the overtake moves more than the possible moves, if there are overtake moves, the computer will select the first move and perform that overtake move and capture a human piece. After overtaking, the algorithm will also check if there is a possible jump move before changing turns so that all legal jump moves are completed and opponent pieces are captured. If there were no overtake moves in the respective turn, the algorithm will choose one possible move at random and the turn will be changed to the human player.

4. Extensibility

List the names of the team members who contributed to this section.
Discuss how your code can be extended for the variants of American Checkers, such as national and regional variants (https://en.wikipedia.org/wiki/Draughts#National_and_regional_variants). What classes and methods need to be changed, and how? How was the Open-Closed Principle applied (i.e., which functions or classes are open for extension, but closed for modification)?

Contributors: Mohamad Abboud, Javkhlan Enkhjin, Brahian Ramon, Pratheek Reddy Patnam

- Our game can be extended to allow a larger board size such as 10x10 or 12x12. We do not have any fixed limitations on the current board size which would make it very simple to convert to a larger board size. The code depends on what size the board was initialized in the board class which is currently set to 8. Also the board constructor needs to be modified to include the extra rows that were added in order to add more pieces to the board with the increase of the board size. Moreover, in order to convert the changes to the user interface, adjustments have to be made to the CheckersBoard.fxml and CheckersBoardComputer.fxml by increasing the grid size, and adding more ellipses to cover the remaining spaces.
- The piece that will possess the first move can also be changed to white (it is currently set to black), to change this feature, in the board class, the turn variable must be changed to white meaning white piece player will start.
- These options make this application extensibile to the following checker variants: Polish Draughts, Ghanian Draughts, Canadian Checkers, Brazilian Draughts, and Russian Draughts.
- The Open-Closed principle is not applied in this situation, as the current board size and turn needs to be modified in the board class to satisfy the requirements of the checker's variants. The 2 elements are initialized at the top of the class and once modified will lead to the extensibility.

V. Findings from the Code Review Exercise

Use the following template to document the findings from the code review of **each** class.

Participant names: Mohamad Abboud, Javkhlan Enkhjin, Brahian Ramon, Pratheek Reddy Patnam
Class that was reviewed: LoginController, CheckersGUI, CheckersGUIComputer, Board, Piece, Main, JavaConnection2SQL

Checklist	Checklist Item	Findings
Coding Standards	Naming conventions	Method and variable naming are suitable
	Ordering convention of method arguments	Method argument ordering is appropriate and meaningful.
	Meaningful and valid comments	Comments to describe complicated algorithms and code are found.
	Consistent style of code blocks	All code blocks are in place.
	Consistent indentation	Every indentation is the same and in place.
Design Principles	Good class abstraction and interface	There is good class abstraction as all the methods and variables in the Board, Piece, CheckersGUI, CheckersGUIComputer, logincontroller respectively possess high cohesion and contain methods and variables only related to their context, with no extra methods that do not play an important role in the code.
	Appropriate visibility of each variable, method, and class	All classes, methods, and variables possess appropriate visibility.

	Design by contract (pre/post-conditions)	The CheckerGUI and CheckersGUIComputer do not require pre or post condition as they are a non demanding design that is mainly event driven and depend on the user clicks, Moreover, we are constantly validating the pre condition by using if statements to make sure the input is correct. LoginController includes registering new users and logging processes. For register, it's checking users valid, username, full name, phone number, and email address. For login, it's checking valid username and password before logging in to the next page.	
	Is the Open-Closed Principle violated?	Yes, because any change to be made to extend the game to a different variant requires a minor modification to the existing code but there won't be any need to extend the code in most cases.	
	Is the Single Responsibility Principle violated?	No, as each class has only one responsibility and has only one reason to change.	
Code Smells	Magic numbers	There is a large number of magic numbers that represent an invalid cell (-1) or a cell that contains a black piece (1) or a white piece (2)	
	Unnecessary global / class variable	There are no unnecessary global variables	
	Duplicate code	There is a small use of duplicate code	
	Long methods	there is 1 very long method (UpdateGamePlayer in CheckersGUI and CheckersGUIComputer)	
	Long parameter list	There are no long parameter list	
	Over-complex expression	There are no over-complex expression	
	Switch or if-then-else that needs to be replaced with polymorphism	There are a large number of if and else statements that need to be replaced with polymorphism	
	Variable or method name whose intent is unclear	There are no variable or method names whose intent is unclear.	
	Any similar methods in other classes?	Yes, the page CheckersGUIComputer class contains several similar methods that are found in the CheckersGUI class.	
Bugs	**Buggy code snippet**	**What is the bug?**	**Why is it a bug?**

VI. Source Code Summary

You must submit all source code. The team will receive no credit for sprint 3 if the source code is not submitted or the following tables are incomplete. A team member will receive no credit for sprint 3 if his/her name is not shown in the following tables.

Summary of all source code files and individual contributions

Source Code File Name	# Lines of code	Javkhlan Enkhjin: #Lines of code	Mohamd Abboud: #Lines of code	Brahian Ramon: #Lines of code	Pratheek Patnam: #Lines of code
LoginController	331	331	-	-	-
CheckersGUI	713	-	713	-	-
CheckersGUIComputer	816	78	295	231	212
Board	125	-	-	51	74
Piece	36	36	-	-	-
Main	34	34	-	-	-
JavaConnection2SQL	44	44	-	-	-
RegisterTest	99	-	-	99	

LoginTest	42	-	-	42	-
CheckersBoardTest	75	-	-	75	-
NewCheckersBoardTest	373	-	-	-	373
UserValidation	99	-	-	99	-
Total Lines	2787	523	1008	597	659

Summary of each member's contribution

Team member	# total number of lines of code contributed	Program files to which the team member contributed
Mohamd Abboud	1008	CheckersGUI, CheckersGUIComputer
Javkhlan Enkhjin	523	LoginController, CheckersGUIComputer, Piece, Main, JavaConnection2SQL
Brahian Ramon	597	CheckersGUIComputer, Board, RegisterTest, LoginTest, CheckersBoardTest
Pratheek Patnam	659	CheckersGUIComputer, Board, NewCheckersBoardTest

VII. Meeting Minutes (only during this sprint)

Report the minutes of all meetings, including, but not limited to: project/sprint planning meeting, stand-up meeting, backlog grooming, retrospective meeting, and pair programming session.

Date	**Time and Duration**	**Place**	**Participant Names**	**Purpose of the Meeting**	**Specific Action Items**
11/30/2021	2:15 pm - 3:15 pm	Library	All Team Members	Assign Tasks.	Similarly, like sprint 1 and 2, we decided to split the tasks again, Brahian and Pratheek were responsible for the Design Documentation, Javkhlan and Mohamad were responsible for the implementation of the production code and test code. We also agreed to join efforts together in reviewing each other's work performing updates and adjustments wherever we see fit. We used the same meeting technique as last time as it worked very well in order to deliver the sprint 2 requirements.
12/2/2021	2:15 pm - 5:00 pm	Library	All Team Members	Review Tasks and Build Report.	This meeting was conducted after all the elements of the report were done, the code was completed so was the design documentation. We sat down to discuss the approach for the code review exercise and provided a constructive review of our code. Which was completed and we were able to finalize the report.
12/7/2021	2:15 pm - 5:00 pm	Library	All Team Members	Build and Finalize Report.	We discovered a few flaws in our report so we conducted a meeting to review and address the problems and fix the report in time for submission.

VIII. Buddy Ratings

If you don't feel comfortable to include your ratings in this report, you may email your ratings to the instructor or grader.

Rating receiver

Rating giver	Mohamad Abboud	Brahian Ramon De La Rosa	Jawa Enkhjin	Pratheek Reddy Patnam
Mohamad Abboud	X	1	1	1
Brahian Ramon De La Rosa	1	X	1	1
Jawa Enkhjin	1	1	X	1
Pratheek Reddy Patnam	1	1	1	X
Average	1	1	1	1