**Class: Login**

Voidvalidate();

Method Descriptions:

Login

Method: validate

Description: Checks to see if the Email and Password matches with a player account in the database.

Preconditions: The login pages correctly loads and is ready to accept user input.

PostConditions: The user is either able to access the main menu of the game page or they are shown an error message if login is unsuccessful.

**Class: Blackjack**

List of methods:

**void populate();**

**void push(var cardType);**

**void hitMe();**

**void stop();**

**void startGame();**

**void indexOf(var addEle) ;**

**void dealerPlay();**

**void splice(var index1, var index2);**

**int burst(var arr);**

**Method**: populate

Description: add all 52 card objects inside the deck

Preconditions: deck should be empty

Postconditions: deck is full

Signature: void populate();

**Method**: push()

Description: add each card into the deck

Preconditions: the deck is not full

Postconditions: one more card added to the deck

Signature: void push(var cardType);

**Method**: hitMe

Description: This method is called whenever a player asks additional card from the dealer, one at a time

Preconditions: There are enough cards in the deck

The player already got two cards

Postconditions: Player has Blackjack

Player stands

Player busts

Signature: void hitMe();

**Method**: stop

Description: This method is called whenever the player clicks stop button to stop withdrawing card from the deck.

Preconditions: There are enough cards in the deck

Player has no Blackjack

Postconditions: Dealer gets a chance to play

Dealer hidden cards are displayed

Signature: void stop();

**Method**: startGame

Description: This method is called when the users starts the blackjack game and the dealer provides two cards to the player and the computer AI.

Preconditions: There are enough cards in the deck.

The cards are shuffled.

Postconditions: The game continues

Signature: void startGame();

**Method**: indexOf

Description: check index to delete card from the deck

Preconditions: the deck is not empty

Postconditions: the index card is deleted from the deck

Signature: void indexOf(var addEle) ;

**Method**: dealerPlay()

Description: This method is called whenever the player clicks the stop button to let the computer AI play against it

Preconditions: the player clicks the stop button

Postconditions: winner and loser are declared

Signature: void dealerPlay();

**Method**: splice

Description: delete card from the deck

Preconditions: the deck is not empty

Postconditions: the card is deleted from the deck

Signature: void splice(var index1, var index2);

**Method**: burst

Description: checks the value of those cards whose value is greater than ten and assigns them as value 10

Preconditions: the array is not empty

Postconditions: manipulates the cards in the array in such a way that favors both player and dealer to win the game

Signature: var burst(var arr);