Final Project: Spades

# Introduction

Project: Implement a two-player version of Spades

Project members: Derek Luong and Jamie Bowers

Class: SWE 681- 001

Date: 2019-04-27

# Design/Architecture

# Installation Instructions

**External Dependencies –** The following dependencies are required in order to install and play the spades game. This paper assumes the user knows how to install, configure, and run these dependencies.

* PostgreSQL 11
* Any Java IDE that supports Maven 3.5.2 (We used IntelliJ) with the 10.0.2 JDK
* JDK 10.0.2
* Java 1.8
* Two Separate Internet Browsers (We tested against Firefox and Google Chrome) or two separate computers.

**Initial Setup -** In order to run the Spades game, the user must set up the initial database this will create the necessary tables for both Spring Security as well as tables we created to run the spades game. Inside the Java project there is a folder called “sql\_scripts” and a file called “create\_database.sql” run this inside the Postgres’s pgAdmin tool using the Query tool.

# Operating Instructions

The first step in order to play this game requires that two user accounts are created

# Game Rules

**Objective –** Be the first player to reach 500 points. If both players reach 500 points in the same round the player with the higher number of points between the two wins.

**Setup** – Each “Round” a 52 Card Deck is shuffled. Two players take turns drawing cards. Player 1 draws a card and chooses to keep it or discard it, if the first card is kept the second card must still be discarded before Player 2 takes their first card otherwise Player 1 takes the second card drawn and discards the first card. The process is repeated by Player 2. This process keeps going until each player has 13 cards. (((1 card kept + 1 card discarded) \* 2 players) \* 13 cards per player = 52 Total Cards in Deck)

**Card Rank –** Ace, King, Queen, Jack, 10, 9, 8, 7, 6, 5, 4, 3, 2. Spades beats every other suited card. E.g. 2S (2 of Spades) beats AH (Ace of Hearts)

**Bidding –** Each player will decide how many “Tricks” they think they can win in a round. They can also choose “nil” if they want to try to lose every trick that hand to try to get 100 points for that round (See Scoring) A “Trick” is simply a winning hand. Bidding occurs each round.

**Scoring –**

* Each winning trick is worth 10 times the bid amount and every trick that is won that goes over the bid amount is referred to as a “Sandbag” or “bag” and is worth 1 point. Example: Player 1 bids 6 tricks and wins 8. They are awarded 62 points, 60 for their bid and 2 for their bag.
* If a player fails to meet the number of tricks they bid the player is deducted the entire bid times 10 points for each bid. Example: Player 2 bids 8 tricks but only gets 6 tricks in the round, they are deducted 80 points.
* Sandbags are kept track of for each player and once a player hits 10 bags they are deducted 100 points, and their bags are reset to 0.
* Nil Bid – A player can bid 0 if they believe they can lose every trick in order to get 100 points for the round. If they fail to do this and get even 1 trick they are deducted 100 points.

**Gameplay –** Player 1 starts the game and plays a card, player 2 then follows. Spades are not allowed to be played unless they have already been “broken”. In order to “Break Spades” a player may play a spade if they do not have any cards of the suit that the opening player plays. Once spades is broken it can be opened with in any following round.

Example: Player 1 has 5S, 4C, 2D and 3D, player 2 has QH, 10D, 3S, and 4S. Player 1 plays 2D, player 2 must play a Diamond, in the next hand Player 1 Plays 3D because Player 2 no longer has any Diamonds they can play their 3 of Spades to win the trick, spades are now broken and Player 1 can play their 5S in the next hand.

The winner of the trick starts the next hand. Each hand is played until the players are out of their initial 13 cards. The round is then scored and a new round is started. This process goes on until one of the players wins the game.

# Why This Application is Secure