Five Card Stud Poker with Betting

Linus W. Freeman III & Brian Woltemate

2015-12-16

ABSTRACT: This is our SWE-681 Semester Project Final Report for our secure multiplayer Internet (web) gaming system for Five Card Stud Poker with Betting.

# 1. Introduction

We designed/implemented our Five Card Stud Poker with Betting project to allow users to:

1. create a new registration of username and password,

2. logon securely to our secure multiplayer Internet (web) gaming system.

3. start a new game.

4. join a game.

5. logoff.

6. resume a game by joining again.

7. take turns betting (raising), calling, or folding for each 2 through fifth card dealt.

8. Listing completed games.

This web application uses SSL/TLS for encrypted communications between Web Browser clients (Chrome and Internet Explorer) and the web server. This application was built using Apache Web Server, Tomcat Server 8.x with JSP/Struts 2.x, MySQL DBMS Server 5.6.x, FindBugs for defect resolution, JQuery, AJAX, Java J2EE, and GitHub server & GitHub Desktop.

After logging in you are directed to a WEB\_INF\games.jsp page where strictly secure asynchronous AJAX mini-http calls are made to allow the users to take turns with game state management.

# 2. Design/Architecture

# 4. Installation Instructions

**Here are the current software download products and versions (version number and 64-bit):**

**MySQL Server Development Tool:**

http://sqldeveloper.solyp.com/download/

**Static Source Analysis Case Tool:**

findbugs-noUpdateChecks-3.0.1

**Integrated Development Environment:**

eclipse-jee-mars-1-win32-x86\_64.zip

**Model View Controller (MVC2) System Libraries:**

struts-2.3.24.1

**Java Web Application Server:**

apache-tomcat-8.0.27

**Database Management System (DBMS):**

mysql-5.6.27-winx64.zip

# 5. Operating Instructions

Here are the steps to running the Five Card Stud with Betting application:

1.) Start the MySQL DMBS Server 5.6.x by running [ bin\mysqld ]

2.) Using the MySQL bin\mysql client run the following from the source code zip file:

Source c:\StrutsAjax\MSQL\_Server\_DDL\fcs\_2.sql

3.) Start the Tomcat J2EE JSP/Struts Server with a valid self-signed server certificate.

Build and deploy the WAR file either using Eclipse with a embedded Tomcat Server, or an a separately installed Tomcat instance.

4.) Using either Google Chrome or/and Internet Explorer go to this web address:

localhost:8443/StrutsAjax/index.jsp

5.) An existing user may simply just logon. Or click on the New User Registration link and after creating a new account, logon.

6.) After player 1 is logged on he may a.) List Completed games, b.) Start a New Game, or c.) Join an existing game.

7.) If the user Starts a New Game then he waits for a player to join his game and then the first card is dealt down to both players and the next card up. Player 1 my raise or fold for each turn and player 2 may call or fold.

8.) Cards dealt second to fifth are bet on each turn with only cards 1 and 5 dealt facing down.

9.) At the conclusion of the game winner information with what each players hand ranks with total betting amount displayed.

# 6. Game rules

**Here are the five card stud poker and betting rules we will design and implement:**

https://www.pagat.com/poker/variants/5stud.html

**Here is a summary of the rules for five-card stud poker and betting:**

1. The dealer rotates counter clockwise and the application deals with secure shuffling of a deck of cards.

2. The first card is dealt "down" to each player.

3. Then each player receives the next three cards "up", one at a time with betting (up to $1.00 max bet for our project) for each new card dealt.

4. Each player can bet by doing one of the following: (1) call, (2) raise, or (3) fold.

5. After the fifth card is dealt “down" and bet on, a winner is determined based on the following card-hand rankings and receives the pot of money:

**The rank of hands from lowest to highest is:**

1.high card

2. pair

3. 4-straight

4. 4-flush

5.two pair

6.three of a kind

7.straight

8.flush

9.full house

10.four of a kind

11.straight flush

# 7. Assurance Case Why we believe it’s secure