Team 6 Blackjack - Design Document

Angelo Rosario, Cory Yang, Guodong Lu, Michael Maddux

V 1.1

7/15/2020

**1.0 Introduction**

This design document details the overall design for an online Blackjack game.

**1.1 Goals and Objectives**

This document describes important aspects of the implementation of the server, client, and game modules of the Blackjack game.

**1.2 Statement of Scope**

Decisions in this document are made based on the following priorities: Usability, efficiency, maintainability and portability.

**1.3 Software Context**

Program will be written in Java, User/game data will be stored in text files. Commands will be available for users to access/modify their information. The client will allow users to access the game. The server will maintain game rooms and user data.

**1.4 Major Constraints**

Issue 1: Users have login information and are only able to access authorized information.

Option 1.1: Information stored User object include username, password, bank. Usernames will be checked to avoid duplicates.

Issue 2: Storage of user information.

Option 2.1: Information stored in text file, but loaded to an arraylist for manipulation purposes.

Issue 3: Must scale to accommodate many clients/games.

Option 3.1: Server will only be able to support so many at once, though we’ll probably never hit peak traffic. Will still limit maximum number of logged in users/available games.

**2.0 Game Design**

This online blackjack game will be a simple blackjack game which gives the players basic actions such as hit, stand, and bet.

**2.1 Client Side**

* Client will keep track of client sessions
* Client will be able to access User data
* Allows User to log in

**2.2 Server Side**

* Server maintains security and logs
* Server maintains User login information and data
* Scales to multiple clients
* Can maintain multiple games at once (clone)
* Server processes actions done by the client

**2.3 Game**

* The game deck will consist of 52 cards
* For 52 cards, there are 4 suits and 13 values
* Dealer will act according to specific rules, will always take a card at 16 and under but will stand at 17+
* Game will have up to 4 players

**3.0 Architectural and Component Level Design**

**3.1 Program Structure**

**3.1.1 Architecture Diagram**

****

**3.2 Description of Client**

The Client program will run on independent machines to connect to the singular Server. The Client will provide an intuitive interface for users to use in order to access account information and to play interactive games of Blackjack with other users using the Client program.

**3.2.1 Client Processing Narrative (functions and processes described)**

Prior to login, the Client will establish a connection with the singular Server using a known IP address and port number. A private key will be established, whereafter all Messages shall be encrypted. The Client will display Messages from the Server in a meaningful manner, in the form of readable information about one’s account or in the form of a display of a game of a Blackjack table. User input will be used to determine the Messages sent by the Client. In a Blackjack game with multiple Clients, the Messages sent by the Client will be used to update the game state, whereafter the Server will send Messages to the connected Clients of that particular instance in order to update the display with the new information.

**3.2.2 Client Interface Description (input and output interfaces described)**

User input will be through text fields and buttons. The input of text fields or the actions of buttons will be used to populate a Message, or subclass of a Message, with the correct type and relevant data. Upon hitting the corresponding Confirm or Ok button, the Message will be sent to the Client. Invalid entry that can be validated by the Client shall be handled by the Client, before a Message is sent. User output will be displayed in a GUI using Swing and AWT. Static information, button labels, and automated responses can be generated by the Client to display. The Client will use Server Messages to display the dynamic information, such as account information and game state.

**3.2.3 Client processing details (algorithmic description)**

The Client will use a Diffie-Hellman key exchange in order to provide a 128 bit private key for encryption. After which, this key will be used to encrypt messages using AES. Each Client shall correspond to one thread for the singular Server.

**3.3 Description of Server**

The Server is a singular program that maintains a multithreaded socket to communicate with multiple connecting Clients. The Server has access to account information for users stored in a local file, protected with a one-way-hash. The Server creates instances of Blackjack games for Clients, of which multiple independent games may be running simultaneously. The Server keeps track of game state and handles game logic.

**3.4 Software Interface Description**

**3.4.1 External interfaces**

Each computer will have their own separate client running on their own respective network. They will be connecting to a server and communicating with each other through that server using a known IP and port.

**3.4.2 Internal Interfaces**

The clients within each computer will be set up so that it can connect to the server and join a table to start a game of blackjack between it’s players connected to the server. Users will be prompted to set up an account or login using their username and password. There will be buttons for users to select their commands throughout the application for anything account related or in-game related.

**3.4.3 Human Interfaces**

The interface will be simple and consistent for the user to keep it easily understandable for all players. Buttons will be easy to read and straightforward for the user.

**4.0 User Interface Design**

For the user interface, we will use the command pattern to encapsulate the id, password, register, etc. We also want to store the buttons by using the facade and command pattern in the user interface.

**5.0 Restrictions, Limitations, and constraints**

**5.0.1** One account can only login with one device each time.

**5.0.2** The game only uses one deck of cards. (52 cards)

**5.0.3** The minimum bet amount must be one game coin. (A number between 0 and player bank)

**5.0.4** Password has to contain at least eight characters, which include at least one number one upper case letter and one lower case letter.

**5.05** Account ID cannot be repeated.

**6.0 Testing Issues**

**6.0.1** The maximum login-users test.

**6.0.2** Judgment test.

**6.0.3** Game's rule logic test.

**6.0.4** A single table reaches the maximum test.

**6.0.5** Log out test.

**6.0.6** Log in test.

**7.0 Appendices**

SRS document

UML diagrams





