



Eötvös Loránd Tudományegyetem
Informatikai Kar
Információs Rendszerek Tanszék

Többasztalos és -felhasználós póker játék adatbázis modellezése

Ács Zoltán
tanársegéd

Fehér Valentin
Programtervező Informatikus BSc

Budapest, 2016

Tartalomjegyzék

1. Bevezetés	3
1.1. A témaválasztás indoklása	3
1.2. A megoldandó feladat leírása	3
1.2.1. A szerver alkalmazás leírása	3
1.2.2. A kliens alkalmazás leírása	4
2. Felhasználói dokumentáció	5
2.1. Témaválasztás indoklása	5
2.2. Feladat leírás	5
2.3. Telepítés	5
2.3.1. Java SE Runtime Environment 8	5
2.3.2. MySQL Community Server 5.6	5
2.3.3. Az adatbázis használatba vétele	6
2.3.4. A póker szerver elindítása	7
2.3.5. A póker kliens elindítása	7
2.4. Felhasznált technológiák	7
2.5. Adatbázis séma	8
2.6. Modulok	9
2.7. Funkciók	10
3. Tovább fejlesztési lehetőségek	11
4. Tesztelés	12
4.1. Funkcionális tesztelés	12
5. Hivatkozások	13

1. Bevezetés

1.1. A témaválasztás indoklása

Mindenképpen egy online, többfelhasználós játékot szerettem volna megvalósítani. Később leszűkítettem a kört kártyajátékokra, és végül az ulti és a póker között vaciláltam. A döntésem a pókerre esett, ugyanis az ulti viszonylag bonyolultabb, mint a póker, több szabály, több megszorítás a partykra vonatkozólag, ráadásul ahány ház annyi szokás alapon könnyen nézet eltérések szoktak keletkezni az ultizás során. Pókerezni egyszerűbb - bár ezt sokan vitatják - és jó formán mindenki könnyen megérti a játék lényegét.

1.2. A megoldandó feladat leírása

A játék általam elképzelt reprezentációja egy szerver- és egy kliens alkalmazásból áll, melyeknek jól meghatározott feladata van. Ezek egymástól függetlenül kerülnek fejlesztése úgy, hogy a kliens csak egy jól megfogalmazott API-t ismer a szervertől, melyen keresztül kommunikálni tudnak. Ennek az API-nak a feladata, hogy egy felületet biztosítson arra, hogy a felhasználók a kliens alkalmazás használatával:

- be tudjanak jelentkezni a szerverre, vagy arra regisztrálni tudjanak
- megtalálják a szerveren elérhető játékosokat, barátokat
- megtekinthessék a ranglistát
- új játékot tudjanak indítani, vagy egy már megkezdett játékot folytatni, amennyiben az abban játszó összes játékos elérhető
- a játékban lehetséges műveleteket végre tudják hajtani, a változásokról értesítést kaphassanak
- játék közben egymásnak szöveges üzenetet küldhessenek
- egy véget ért játékot le tudjanak menteni

Az API pontos megfogalmazása után a szerver- és a kliens alkalmazást egymástól teljesen külön kezelhetjük.

1.2.1. A szerver alkalmazás leírása

A szerver alkalmazás egyrészt egy webes felületet nyújt az adminisztrátorok számára, melyen keresztül a felhasználókat és a játékokat, eredményeket valamint a játékhöz tartozó naplókat kezelhetik.

Másrészt pedig a szerver alkalmazás a teljes játékot (a játék logikáját és a lekérdezett játék aktuális állapotát), mint szolgáltatást nyújtja a fent említett API-n keresztül, melyet a kliensek autentikálás után használhatnak.

A szerver párhuzamosan egyszerre több játékot is tud futtatni és egy játék a játékosok kérésére bármikor felfüggesztésre, majd később folytatásra kerülhet.

A szerver rendelkezik egy ranglistával is, melyen a játékosok az alábbi módon kerülhetnek feljebb:

- 10 pontot kap az a játékos, aki megnyer egy játékot
- 5 pontot kap az a játékos, aki ugyan tudta a megoldást, de egy másik játékos megelőzte a vádolás folyamatában

1.2.2. A kliens alkalmazás leírása

A szakdolgozatom kliens alkalmazása egy olyan asztali alkalmazás, amely egy jól átlátható felületet nyújt a játékhöz. Elsődleges célja, hogy a szerver API-ját használva lekérdezze a játék állapotát, majd a kapott állapotot megjelenítse, valamint, hogy a felhasználói interakciót a szerver felé továbbítsa és az eredményt megjelenítse.

Magán a játék felületen kívül a kliens alkalmazáson keresztül a felhasználó új játékot hozhat létre barátok, vagy online játékosok kiválasztásával, kezelheti saját profilját és beállításait, valamint megtekintheti a szerver ranglistáját.

A kliens alkalmazás ezen felül lehetőséget ad a felhasználónak arra, hogy lementett játékokat játsszon vissza.

2. Felhasználói dokumentáció

2.1. Témaválasztás indoklása

Mindenképpen egy online, többfelhasználós játékot szerettem volna megvalósítani. Később leszűkítettem a kört kártyajátékokra, és végül az ulti és a póker között vaciláltam. A döntésem a pókerre esett, ugyanis az ulti viszonylag bonyolultabb, mint a póker, több szabály, több megszorítás a partykra vonatkozólag, ráadásul ahány ház annyi szokás alapon könnyen nézet eltérések szoktak keletkezni az ultizás során. Pókerezni egyszerűbb - bár ezt sokan vitatják - és jó formán mindenki könnyen megérti a játék lényegét.

2.2. Feladat leírás

2.3. Telepítés

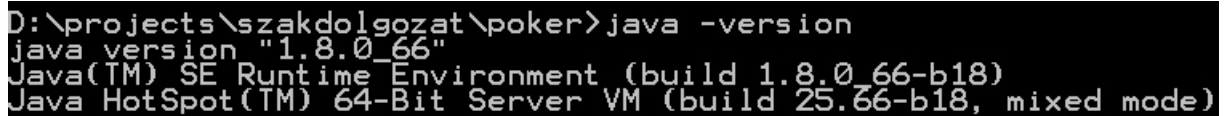
2.3.1. Java SE Runtime Environment 8

A programcsomag futtatásához legalább Windows XP operációs rendszer szükséges, amelyen Java SE Runtime Environment 8 futtató környezet [1] (a továbbiakban: JRE) fut. A JRE feltelepítését követően manuálisan ellenőrizzük, hogy a rendszer felvette-e környezeti változóként az installációs könyvtárat. Navigáljunk az operációs rendszerben a környezeti változók módosítása panelhez, majd ellenőrizzük le, hogy a PATH nevű környezeti változóhoz hozzá lett-e adva az installációs könyvtár: C:\ProgramFiles\Java\jdk1.8.0_60\bin. Ha nem, akkor pontosvesszővel (;) elválasztva egészítsük ki a változó értékét, majd indítsuk el a promptot (ha nyitva van, akkor indítsuk újra). Ha a

```
java -version
```

utasítás hatására az 1. ábrán látható szöveg jelenik meg a konzolon, akkor sikeres

1. ábra. JRE verzió



```
D:\projects\szakdolgozat\poker>java -version
java version "1.8.0_66"
Java(TM) SE Runtime Environment (build 1.8.0_66-b18)
Java HotSpot(TM) 64-Bit Server VM (build 25.66-b18, mixed mode)
```

volt a JRE telepítése és beállítása. További instrukciókért ld. melléklet.

2.3.2. MySQL Community Server 5.6

A programcsomag megköveteli a MySQL Community Server 5.6 adatbázis-kezelő rendszer [2] (a továbbiakban: MySQL Server) használatát is. Letöltés után csomagoljuk ki a zip állományt egy tetszőleges könyvtárba, majd a fentiekkel megegyező

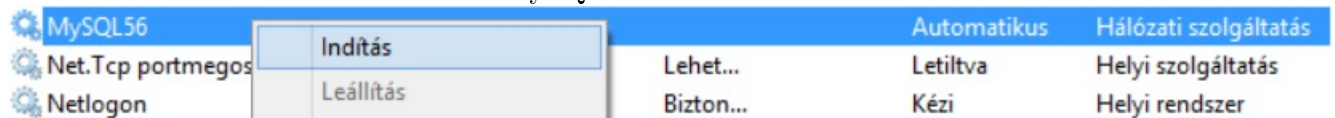
módon adjuk hozzá a PATH nevű környezeti változó értékéhez a MySQL Server bin könyvtár elérési útvonalát. Ha ezzel végeztünk, akkor nyissuk meg a promptot (ha nyitva van, akkor indítsuk újra), majd navigáljunk a MySQL Server bin könyvtárba, ott pedig adjuk ki a

```
mysqld --install
```

parancsot. A parancs végrehajtása után navigáljunk a szolgáltatások panelhez, amelyet a legkönnyebben a promptban a

```
services.msc
```

2. ábra. MySQL Service



kiadott utasítással lehet elérni. Majd járjunk el a 3. ábrának megfelelően. Térjünk vissza a konzolra, ahol adjuk ki a

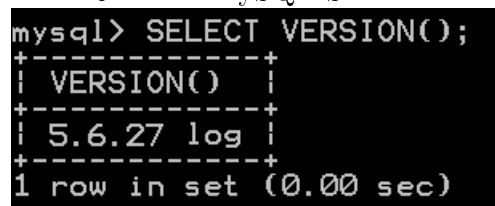
```
mysql -u root -p
```

parancsot, amely jelszót fog kérni. A beviteli sort hagyjuk üresen, nyomjunk entert. Ha sikeresen beléptünk az adatbázis-kezelő rendszerbe, akkor adjuk ki a

```
SELECT VERSION();
```

utasítást, és ha a 3. ábrának megfelelő képernyőképet kapunk, akkor sikeresen felte-

3. ábra. MySQL Service



lepítettük az adatbázis-kezelő rendszert.

2.3.3. Az adatbázis használatba vétele

Ha sikeresen elindítottuk a MySQL Servert, akkor szükségünk lesz egy új adatbázis sémára (és demo adatokra), amelyet a X:\poker\release\poker-db.sql állományban találunk. Ezt a fílet kell lefuttatni az adatbázison, a hatása idempotens. A promptban adjuk ki a

```
mysql -u root -p < X:\poker\release\poker-db.sql
```

utasítást, amely jelszót fog kérni. A beviteli sort ugyancsak hagyjuk üresen. Ha sikeresen lefutott a parancs, akkor az adatbázis séma „felhúzása” megtörtént.

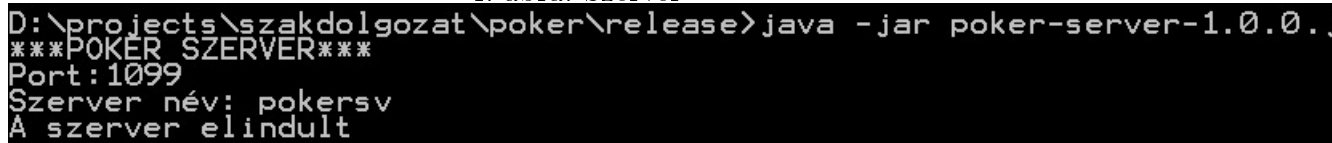
2.3.4. A póker szerver elindítása

A DVD lemezen a `\poker\release\` mappában található meg a `poker-server-1.0.0.jar` file. Nyissunk egy terminált a kijelölt könyvtárban, és adjuk ki a

```
java -jar poker-server-1.0.0.jar
```

parancsot. Ha a 4. ábrának megfelelő konzol loglistát látunk, akkor a szervert siker-

4. ábra. Szerver



```
D:\projects\szakdolgozat\poker\release>java -jar poker-server-1.0.0.jar
***POKER SZERVER***
Port:1099
Szerver név: pokersv
A szerver elindult
```

resen elindítottuk.

2.3.5. A póker kliens elindítása

A kliens futtatása hasonló módon történik, mint a szerveré. Navigáljunk a `\release\poker\kliens` mappába, és a konzolon adjuk ki a megfelelő parancsot. Jöhet az ábra, meg a kódot kicsit átírni, hogy logoljon konzolra, mint a szerver...

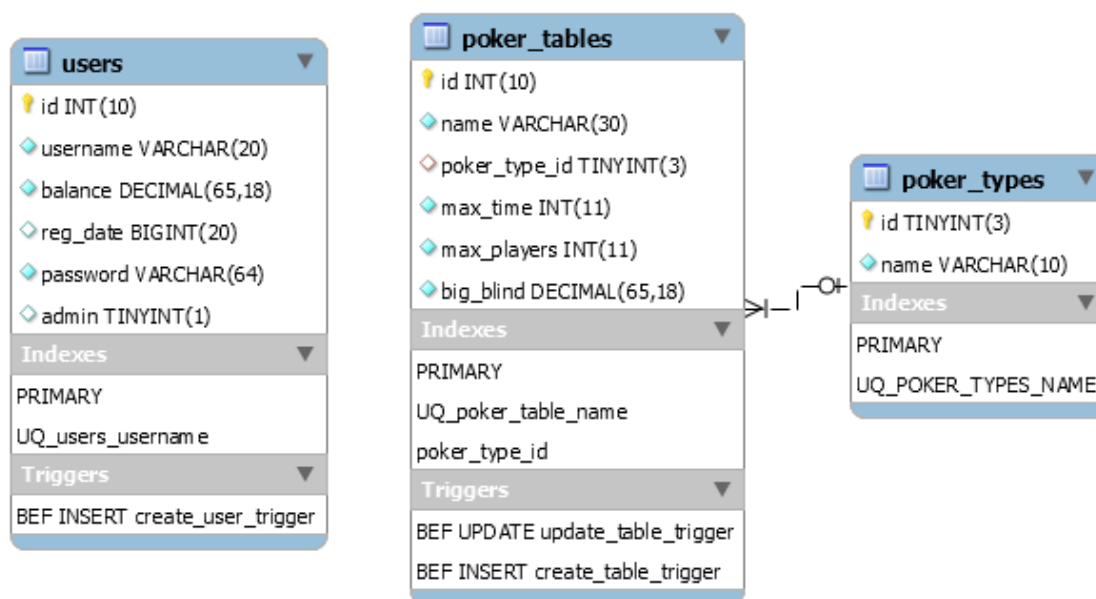
2.4. Felhasznált technológiák

A szakdolgozatomat eclipse fejlesztőkörnyezetben írtam, amelyet végül mavenel fordítottam ki és csomagoltam be. A szakdolgozat felhasznál egy külső könyvtárat [3], amely a nyertes kiértékelési feladatát látja el. A programcsomagot meg kellett támogatni egy adatbázissal is - MySQL -, amely az adatok perzisztens tárolásáért felel. A programcsomag szerver-kliens architektúrában került implementálásra, amely kliens oldalon tovább bomlik MVC (Model-View-Controller) tervezési stílusra. A szerver és a kliens közötti kommunikáció RMI Java API felhasználásával történik. A maven, mint build system nagyban hozzájárult a programcsomag modernkori elvárások megfeleléséhez, mint például a kód dokumentálásához és a programcsomag disztribúciójához. Igyekeztem áttolni a fejlesztési nehézségeket a felhasznált technológiák vállára. Így például, ha valamilyen oknál fogva friss adatbázisra volt szükségem, akkor azt a

```
mvn clean install -Pbootstrap
```

parancs kiadásával érhettem el.

5. ábra. Adatbázis séma



2.5. Adatbázis séma

Az adatbázis (ld. X. ábra) 3 táblából épül fel

- users
- poker_tables
- poker_types

Minden tábla rendelkezik elsődleges kulccsal, amelynek típusa UNSIGNED INTEGER, kivétel a poker_types táblát, ugyanis ennek a táblának az elsődleges kulcsa UNSIGNED TINYINT típusú, amelyet a MySQL Server 4 byteon tárol. Az adatbázis mértékének a szűkítése ebben az esetben indokolt, ugyanis a 255 különböző értékű UNSIGNED TINYINT típus kielégíti a játéktípusok által támasztott követelményeket. A users tábla tartalmazza a regisztrált felhasználókat. A regisztrált felhasználók felhasználónévvel és jelszó párossal tudnak regisztrálni, és ennek megfelelően ezek az adatok tárolásra is kerülnek. A felhasználónév maximális hossza 20 karakter, ameyelet triggerrel ellenőrzők. A jelszót bcrypt függvénnyel nyírom. Só eltárolása nem szükséges a bcrypt implementációjából adódóan. A felhasználóról el kell még tárolni a regisztráció dátumát, amely a szerver ideje alapján számolódik és UNIX timestampként kerül letárolásra, továbbá a jogosultsági (admin) szintet, amely ugyancsak 4 byteon (TINYINT) kerül ábrázolásra. A 0/1 értékek megfeleltethetők a TRUE/FALSE logikai típusú konstans értékeknek, így tehát, ha az érték 0, vagyis FALSE, akkor az adott felhasználó nem rendelkezik admin jogkörrel, különben igen. Ugyancsak tárolandó érték a felhasználó játékbeli egyenlege, amely

BIGDECIMAL típusként van ábrázolva. A játéktáblákat a `poker_tables` adatbázis tábla tárolja. Szükségünk van eltárolni a játéktábla nevét, melynek felső korlátja 30 karakter és egyedinek kell lennie. Ezeket a megszorításokat szintén triggerekkel ellenőrzöm. A játék asztal játéktípusát is eltárolom, amely egyben idegenkulcs is a `poker_types` táblára nézve. Továbbá minden asztal tulajdonsága, hogy maximum hányan játszhatnak rajta, és hány másodpercig gondolkodhatnak a játékosok. Ezen két érték típusaként ugyancsak UNSIGNED TINYINT van meghatározva, ugyanis az egyes asztaloknál a játékosok száma legfeljebb 5 lehet, míg az egyes játékosok gondolkodási ideje maximum 40, de legalább 5 másodperc.

2.6. Modulok

A programcsomag 6 fő modult tartalmaz

poker-server

A póker játék szervere, amely magát a játékot szolgáltatja.

poker-client

A póker játék kliense, amely segítségével a szerverhez lehet csatlakozni.

poker-shared

A póker játék azon modulja, amelytől a szerver és a kliens egyaránt függ.

poker-persist

Az adatok letárolásáért felelős modul.

poker-model

A póker játék modellezéséért felelős csomag.

javapokertexasholdem

Külső könyvtár, amely a nyertes játékos kiértékelési feladatot végzi.

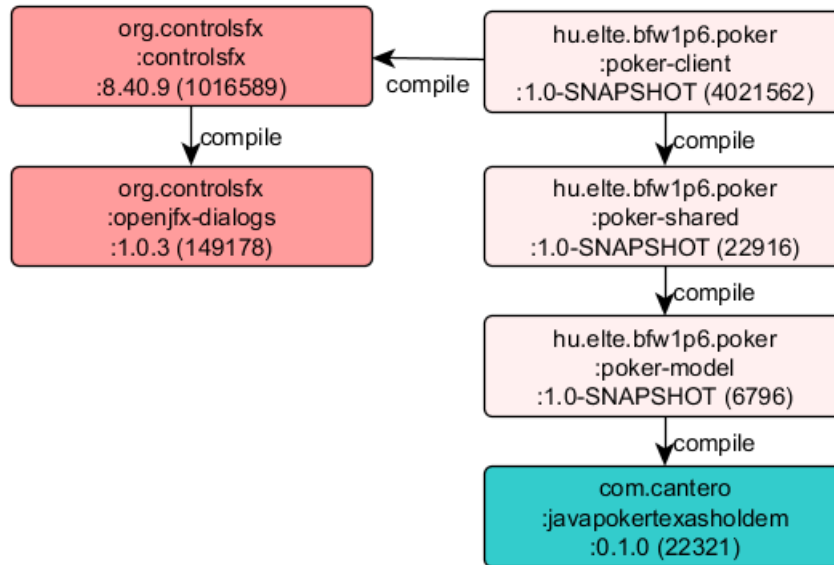
A modulok közötti függőséget a 6. és a 7. ábra szemlélteti.

A programcsomag két fő modulra bontható

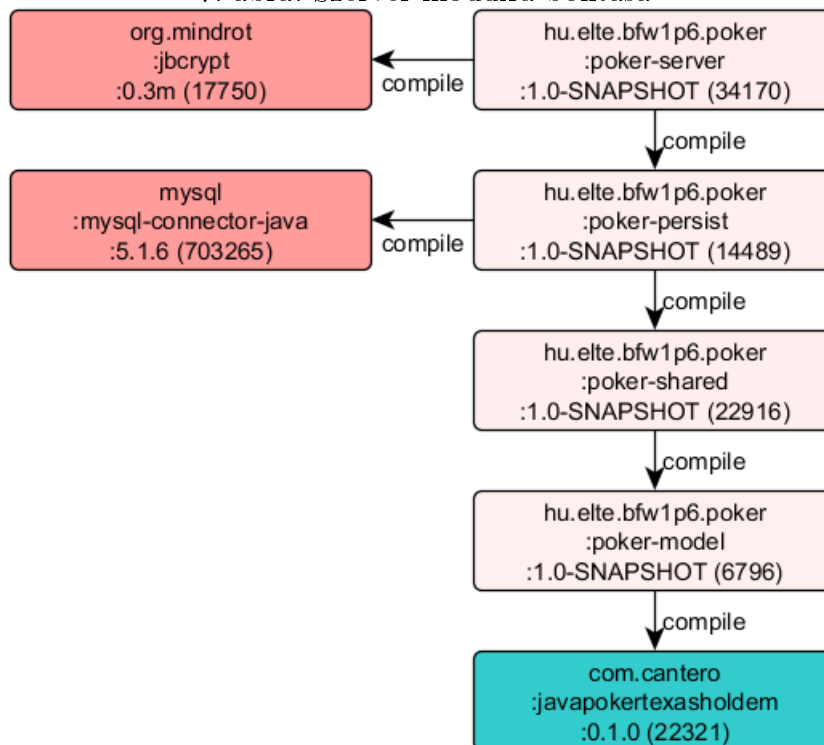
- poker-server
- poker-client

poker-server és poker-client. A szerver a jelszavak titkosítására bcrypt eljárást alkalmaz, amelynek a biztonságát szózással növeli. Továbbá a szerver felhasznál még egy külső csomagot - `mysql-connector-java` -, amely az adatbázis kapcsolatért felel. A poker-shared modul felel a szerver és a kliens jól definiált kommunikációjáért. A shared modul többek között tartalmazza a közös interfészeket, kivételeket és a póker utasítások megvalósítását. A kliens közvetlenül függ ettől a modultól, azonban a

6. ábra. Kliens modulra bontása



7. ábra. Szerver modulra bontása

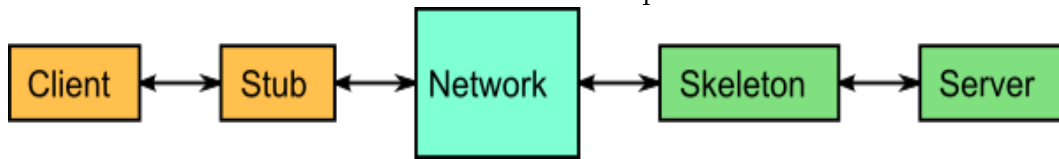


szerver és a shared modul közé beékelődött a poker-persist modul, amely az adatok adatbázisba való írásáért és abból olvasásáért felel.

2.7. Funkciók

Ahogy a témabejelentőben is szerepel..... RMI kép wikiről, majd azt megmagyarázni a shared modullal, kliens hívja, jól definiált interfész etc.... A szerver vázáért a PokerRemote interface felel, amely a játékon végrehajtható műveletek összefogásáért

8. ábra. RMI koncepció



felel. Itt található az összes funkció, amely megvalósításra került, mint például játék asztal létrehozása, új felhasználó létrehozás, admin jog kiosztása stb. A kliens ezt a vázat tudja elkérni az RMI registryből, mint kliens-oldali servercsonk, amelyeken a műveletek meg tudja hívni. Az összes megvalósított funkciót le kell írni? Felsorolás szintjén, vagy hogy? Rövid magyarázattal? És amelyik egyértelmű? Pl. felhasználó módosítása... login...

3. Tovább fejlesztési lehetőségek

- Az adatbázis viszonylag alacsony absztrakciós szinten került implementására, azonban mivel néhány tábláról beszélhetünk csak, ezért igyekeztem elkerülni a keretrendszerek általi overheadet. Ugyanakkor ezen a ponton sokat fejlődhet a programcsomag, ha a későbbiek során esetlegesen bonyolultabban kellene modellezni a játékot adatbázis szempontjából. Például dialektusok - akár Liquibase (hivatkozás) - használata elfedheti a tényleges adatbázis-kezelő rendszer általánosságait, így eggyel magasabb szintre helyezhető a megvalósítás.
- A felhasználói élményen sokat javíthat az animációk használata. A megjelenítés sokkal lágyabb, folyékonyabb lehetne Transition/Animation (bibliográfiába hivatkozás...) objektumok használatával.
- Akár a komplett RMI architektúrát le lehetne váltani, és helyette REST szoftverarchitektúrát alkalmazni, amely modernebb megjelenést (AngularJS, responszív design) és modernebb fejlesztői eszközöket, API-kat vonna maga után.
- A játék nem képes kezelni olyan eseteket, amikor egynél több játékos nyer az adott körben.
- A játék nincs felkészítve arra a szélsőséges esetre, ha valakinek elfogy a zsetonja, akkor pontosan minek (és hogyan) kell történnie.

4. Tesztelés

4.1. Funkcionális tesztelés

Funkció	Elvárt eredmény
Regisztráció	A program jelezte a felhasználónak, hogy a regisztráció sikeresen megtörtént, és
Bejelentkezés	A formot helyesen kitöltve a program sikeresen autentikálta
Tábla módosítás	-
Tábla törlés	-

5. Hivatkozások

- [1] Java SE Runtime Environment 8 letöltése
<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155>
- [2] MySQL Community Server 5.6
<https://dev.mysql.com/downloads/mysql/5.6.html>
- [3] Java Poker Texas Holdem Hand Evaluator
<https://github.com/phstc/javapokertexasholdem>
- [4] bcrypt
<https://en.wikipedia.org/wiki/Bcrypt>