



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. Felhasználói dokumentáció	3
1.1. Telepítés	3
1.1.1. Java SE Runtime Environment 8	3
1.1.2. MySQL Community Server 5.6	3
1.2. Futtatás	4
1.3. Felhasznált technológiák	4
1.4. Adatbázis séma	4
1.5. Modulok	5
2. Funkciók	6
3. Tovább fejlesztési lehetőségek	7
4. Tesztelés	7
4.1. Funkcionális tesztelés	7
5. Hivatkozások	9

1. Felhasználói dokumentáció

1.1. Telepítés

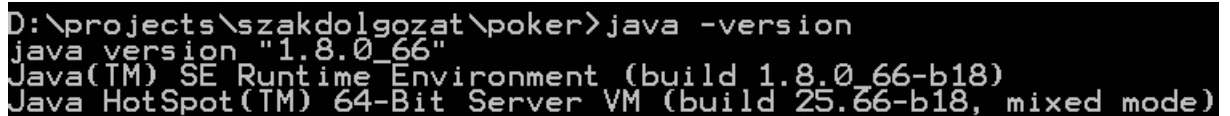
1.1.1. Java SE Runtime Environment 8

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.

1.1.2. 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 könyvtárába, ott pedig adjuk ki a

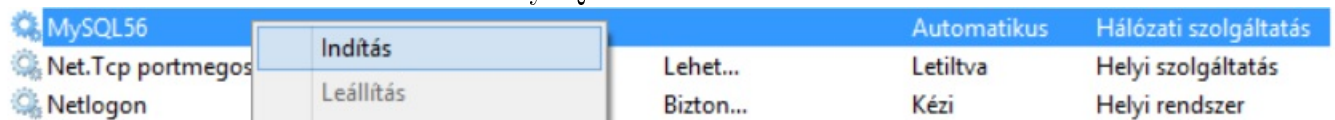
```
mysqld --install
```

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

```
services.msc
```

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

2. ábra. MySQL Service



```
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

```
mysql> SELECT VERSION();
+-----+
| VERSION() |
+-----+
| 5.6.27 log |
+-----+
1 row in set (0.00 sec)
```

lepítettük az adatbázis-kezelő rendszert.

1.2. Futtatás

A program java programozási nyelvben lett megírva, így a kifordított állomány egy jar file, melyet parancssorból az alábbi utasítással tudunk futtatni

```
java -jar <filenev>
```

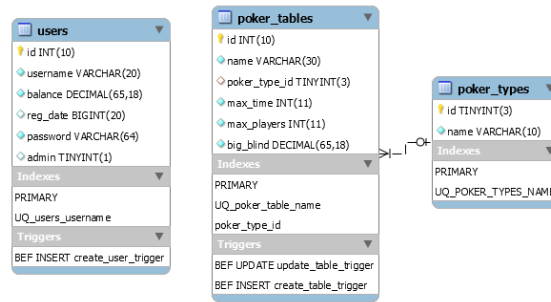
1.3. Felhasznált technológiák

A szakdolgozatomat eclipse fejlesztőkörnyezetben írtam, amelyet végül maven - build rendszer - fordítottam ki és csomagoltam be. A szakdolgozat felhasznál egy külső könyvtárat, 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 tovább bomlik kliens oldalon MVC (Model-View-Controller) tervezési stílusra. A modulok közötti kommunikáció RMI Java API felhasználásával történik.

1.4. Adatbázis séma

Az

4. ábra. Adatbázis séma

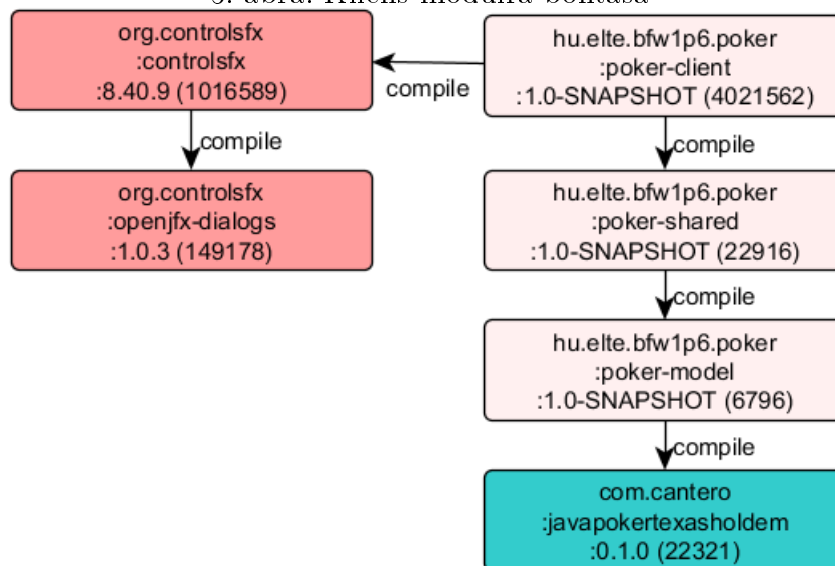


1.5. Modulok

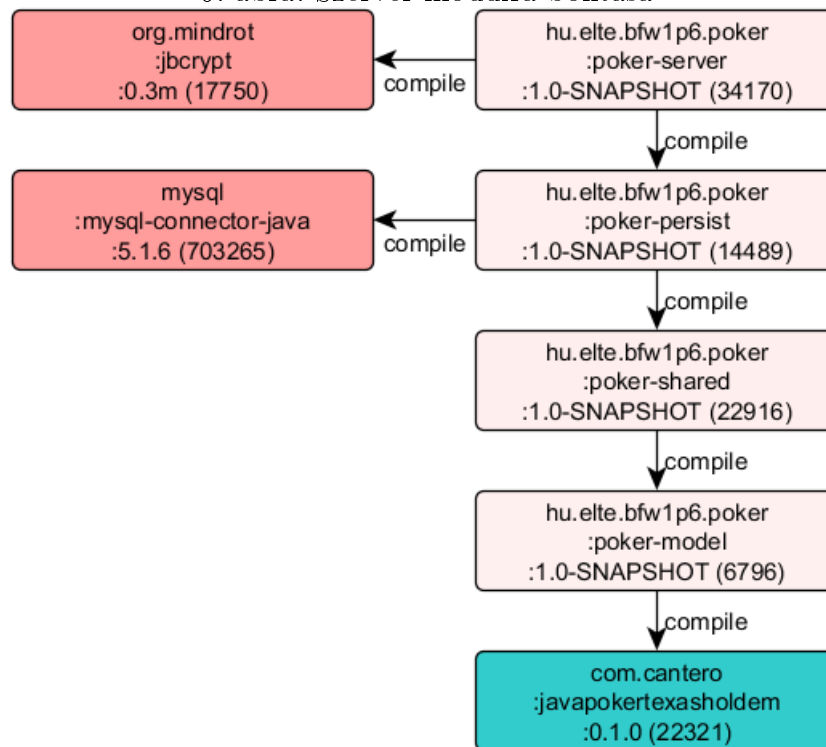
A programcsomag 6 fő modult tartalmaz

- 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.

5. ábra. Kliens modulra bontása



6. ábra. Szerver modulra bontása

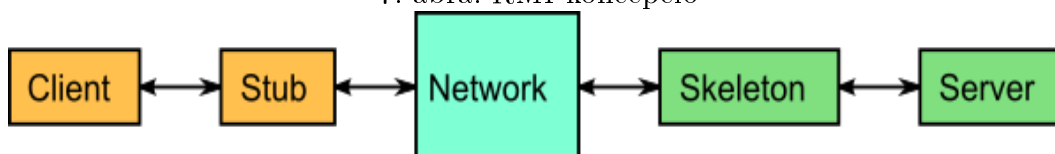


A modulok közötti függőséget a ... ábra mutatja. A programcsomag két fő modulra bontható: poker-server és poker-client. A szerver a jelszavak titkosítására bcrypt eljárást alkalmaz, amelynek a biztonságát szózással növeli. Továbbá a szerver felhasznál még egy külső csomagot - mysql-connector-java -, 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 szerver és a shared modul közé beékelődött a poker-persist modul.

2. Funkciók

Ahogy a témabejelentőben is szerepel..... RMI kép wikiről, majd azt megmagyarázni a shared modullal, kliens hívja, jól definiált interfész etc.... A szerver vázáért

7. ábra. RMI koncepció



a PokerRemote interface felel, amely a játékon végrehajtható műveletek összefogásáért 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 a registryből, mint kliens-oldali szervercsonk, 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 (JDK 1.1-ben jelent meg 18 éves technológia [a http meg 16...]) le lehetne váltani, és helyette REST szoftverarchitektúrát tenni, amely modernebb megjelenést (AngularJS, rezponzív design) és modernebb eszközöket vonna maga után.

A ?? képen látható az adatbázis séma.

4. Tesztelés

4.1. Funkcionális tesztelés

Funkció	Elvárt eredmény
Regisztráció	A program jelezte a felhasználónak, hogy a regisztráció sikeresen megtört
Bejelentkezés	A formot helyesen kitöltve a program sikeresen autenti
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>