Automatizált adatbázis-frissítés

Szakdolgozat beszámoló

Írta: Tobik János – SEVOCP

Témavezető: Dr. Vassányi István

\*insert pannon egyetem logó\*

Pannon Egyetem

Villamosmérnöki- és Információs Rendszerek Tanszék

Tartalom

1. Bevezetés
2. Specifikáció
3. Rendszerterv
4. Technológiák
   1. PostgreSQL
   2. Java
      1. Java Database Connectivity
      2. Java Fx
   3. Apache Commons IO
   4. Git
5. Meglévő adatbázis
6. Az eddig elkészült rendszer
7. A következő félév ütemezése

2. Specifikáció

A Lavinia jelenleg relációs adatbázist használ a receptek, az ételek és a tápanyagok tárolására, melyet a PostgreSQL relációsadatbázis-kezelő rendszerrel valósít meg.

4. Technológiák

4.1 PostgreSQL

A PostgreSQL, más néven Postgres egy relációsadatbázis-kezelő rendszer. Szabad szoftver, melynek fejlesztését önkéntesek végzik közösségi alapon. A munka elsődleges koordináló oldala a postgresql.org. Kezdetben a Berkeley Egyetemen indult meg a fejlesztése a nyolcvanas években, majd a kilencvenes évek közepére elhagyta az egyetem falait és nyílt forráskódúvá vált.

A relációsadatbázis-kezelő rendszer (RDBMS) egy olyan adatbázis-kezelő rendszer, amelynek logikai adatbázisát szoftverkomponensei kizárólag a relációs adatmodellek elvén épülnek fel, illetve kérdezhetőek le. Kizárólag a relációs adatmodell alapú megközelítést támogatja. A relációsadatbázis-kezelő rendszerek szabványos adat hozzáférési nyelve az SQL (Structured Query Language). Az SQL segítségével könnyen és érthetően leírhatók akár az összetettebb CRUD (Create, Read, Update, Delete) funkciók is.

Az adatbázis adataihoz való hozzáférést, manipulációt, valamint az adatszerkezet tanulmányozásához a pgAdmin programot használtam.

4.2 Java

A letölthető fájlok kezelésére, az adatbázis elérésére és manipulálására olyan szoftver kell, amely vezérli az adatátvitelt és közben erről tájékoztatást nyújt a felhasználó számára a folyamatról. Ezért döntöttem a Java nyelven történő implementálásról a feladat során.

A Java egy általános célú, objektumorientált programozási nyelv, amelyet a Sun Microsystems fejlesztett a kilencvenes évek elejétől. Jelenleg az Oracle gondozásában áll. A Java alkalmazásokat jellemzően bájtkód formátumra alakítják. A bájtkód futtatását a Java virtuális gép (Java Virtual Machine) végzi, ami vagy interpretálja a bájtkódot, vagy natív gépi kódot készít belőle, és azt futtatja.

A program fejlesztéséhez a NetBeans IDE fejlesztőkörnyezetet választottam. A Java 8-as környezet szükséges a program futtatásához.

A feladat implementálásához a Java nyelvet választottam. A specifikáció során említett funkciókat könnyű vele megvalósítani. A fájlkezelést, az adatbázis kapcsolatot, a grafikus megjelenítést és az ezek közötti adatátvitelt egyaránt egyszerű felépíteni. Habár a Java Virtual Machine gyorsasága nem éri el a hardware közeli nyelvekét, ez nem jelent számottevő hátrányt a működésben.

4.2.1 Java Database Connectivity

A Java Database Connectivity, röviden JDBC egy API a Java programozási nyelvhez, amely az adatbázishozzáférést támogatja. A JDBC definiálja az adatbázisok lekérdezéséhez és módosításához szükséges osztályokat és metódusokat, miközben igazodik a relációs adatmodellhez.

Az említett osztályok a „java.sql” csomagban találhatóak. Adatbáziskapcsolatot a csomag „Connection” osztály példányával lehet létrehozni. Egy ilyen objektumot a „DriverManager.getConnection()” metódus segítségével adhatunk meg. A metódus paramétereként megadhatjuk az adatbázis elérési útvonalát (IP cím, port szám), nevét, valamint a hozzá tartozó felhasználó nevet és jelszót.

Az adatbázishoz való hozzáférés után SQL parancsot a „Statement” vagy „PreparedStatement” osztályok valamelyikével alkothatunk. A parancsot saját magunknak kell megadnunk szöveges formátumban úgy, hogy megfeleljen az SQL szintaktikának.

„PreparedStatement pstmt = conn.preparedStatement(„SELECT \* FROM MyTable”);

pstmt.execute();„

A végrehajtott művelet után fontos, hogy bezárjuk az adatbáziskapcsolatot. Fontos, hogy az operációs rendszer ne tartsa fent feleslegesen a kapcsolatot és ne foglalja a memóriát, ha nem szükséges. Ezt a „close()” metódussal tehetjük meg, amit a „Connection” objektumunkkal tudunk meghívni.

Ha bármilyen hiba adódna az adatbázis művelet végrehajtása során, úgy egy „SQLException” kivételt dob a program.

4.2.2 Java Fx

A JavaFX egy szoftver platform az asztali alkalmazások létrehozásához, amely a Swing mellett a Java Standard Edition alapértelmezett GUI könyvtára. A JavaFX applikációkat bármilyen asztali, mobil eszközön vagy böngészőben lehet futtatni. A grafikus felületet egy XML fájl definiálja, amelyet FXML fájlban tárolunk. A grafikus felület a Scene Builder program segítségével egyszerűen összeállítható és nem kell aggódnunk az XML fájl összeállításában, mert az automatikusan legenerálja.

4.3 Apache Commons IO

Az Apache Commons IO egy Java könyvtár fájlműveletekhez, melyet az Apache Foundation felügyelete alatt fejlesztettek. Osztályok sokaságát biztosítja a fejlesztők számára, hogy egyszerűbb, rövidebb és érthetőbb kód íródhasson a fájlok kezelésére. Habár a feladatom során kevés műveletet kell végre hajtanom fájlokkal, mégis fontos része a programnak, hiszen fájlok által beolvasott tartalommal kell frissíteni az adott adatbázist.

4.4 Git

A Git egy nyílt forráskódú, elosztott verziókezelő szoftver. Feladata, hogy a projekt fájljainak különböző verzióit tárolja. Implementálás közben nagyon hasznos, ha egy fájl korábbi verziójához szeretnék hozzáférni és használni.