
SHOPPING APPLICATION

Név: Mészáros Ádám

Neptun kód: OUNAQC

Az alkalmazás leírása

Az alkalmazásban egyszerűen nyomon követhetjük kiadásainkat és bevételeinket. Lehetőségünk van külön kategóriákban megadni, hogy mikor, mire, és mennyit költöttünk. Ezek az adatok adatbázisban tárolódnak, így a későbbiek során bármikor visszaneézhetjük kiadásainkat, amelyekhez megjegyzéseket is fűzhetünk. Ezeket az adatokat az alkalmazáson belül táblázatos formában tekinthetjük meg, amelyeket a táblázaton belül bármelyik oszlop szerint rendezhetünk, pl. dátum szerint csökkenő vagy növekvő sorrendben.

Jelmagyarázat



Háztartás

Leírás: Az ikonra kattintva megjelennek a **Háztartás** kategóriába felvitt kiadások.



Étkezés

Leírás: Az ikonra kattintva megjelennek az **Étkezés** kategóriába felvitt kiadások.



Autó/Motor

Leírás: Az ikonra kattintva megjelennek az **Autó/Motor** kategóriába felvitt kiadások.



Szórakozás

Leírás: Az ikonra kattintva megjelennek a **Szórakozás** kategóriába felvitt kiadások.



Utazás

Leírás: Az ikonra kattintva megjelennek az **Utazás** kategóriába felvitt kiadások.



Bevásárlás

Leírás: Az ikonra kattintva megjelennek a **Bevásárlás** kategóriába felvitt kiadások.



Előfizetések

Leírás: Az ikonra kattintva megjelennek az **Előfizetések** kategóriába felvitt kiadások.



Egyéb

Leírás: Az ikonra kattintva megjelennek az **Egyéb** kategóriába felvitt kiadások.



Bevétel rögzítése

Leírás: Az ikonra kattintva felvihetünk egy összeget, amely hozzáadódik egyenlegünkhöz.



Kiadás rögzítése

Leírás: Az ikonra kattintva kiválaszthatjuk, hogy melyik kategóriába szeretnénk kiadást rögzíteni. Az itt felvitt összeget később megtekinthetjük a választott kategória ikonjára kattintva.



Export készítése

Leírás: Az ikonra kattintva a program készít egy exportot, amelyet automatikusan elment a program gyökérkönyvtárába.

Menüsor

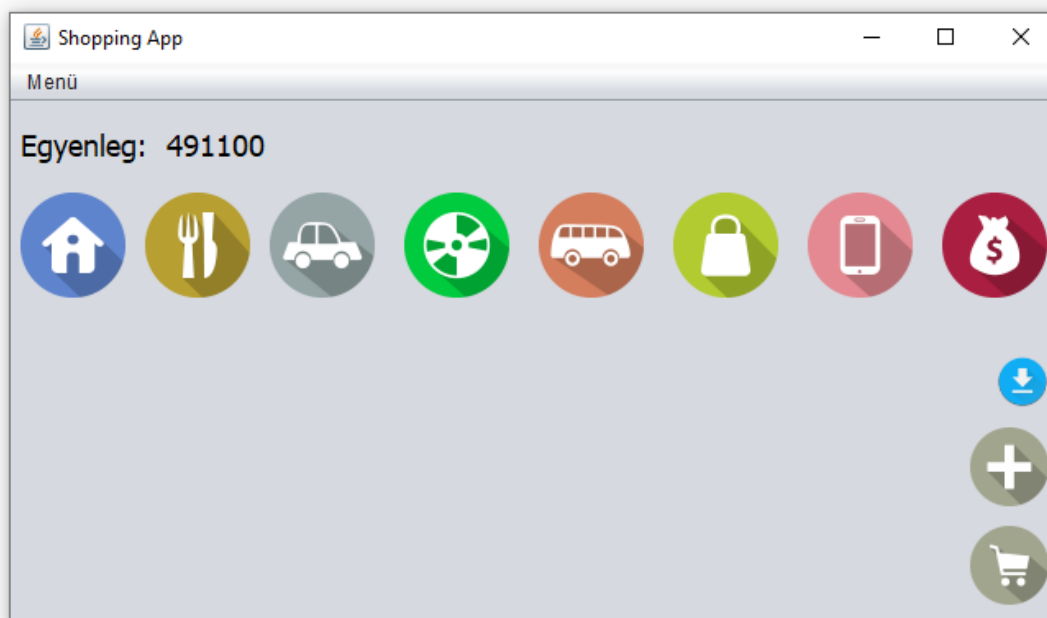
A menüben 2 elem található.

Névjegy: Megjeleníti az alkalmazás névjegyét, ami tartalmazza az alkalmazás nevét, verziószámát, és a fejlesztő adatait.

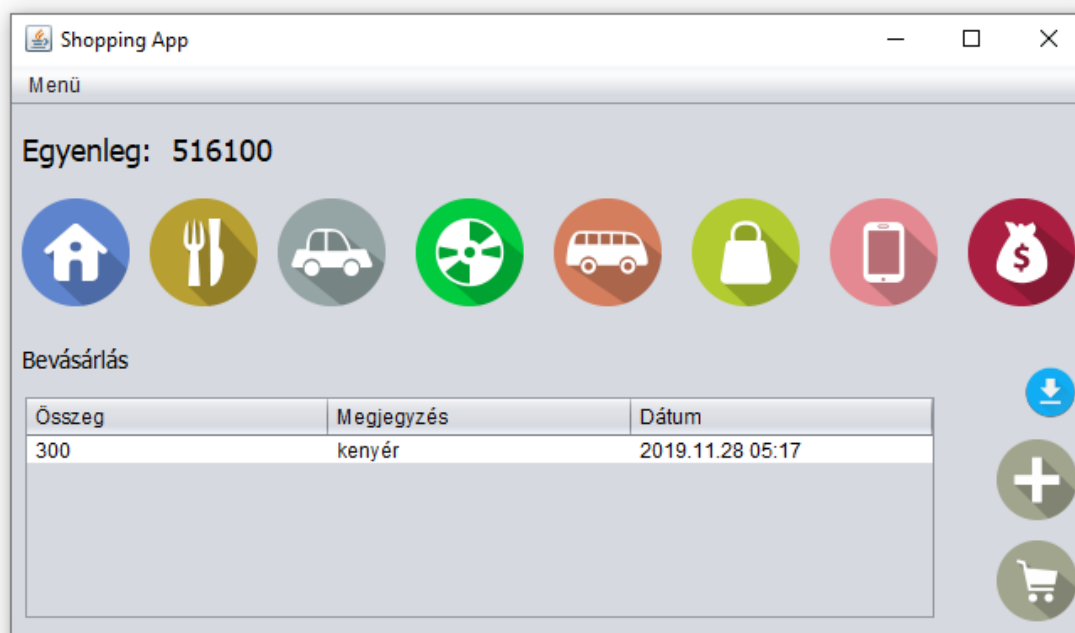
Kilépés: Bezárja az alkalmazást.

Az alkalmazás grafikus felülete

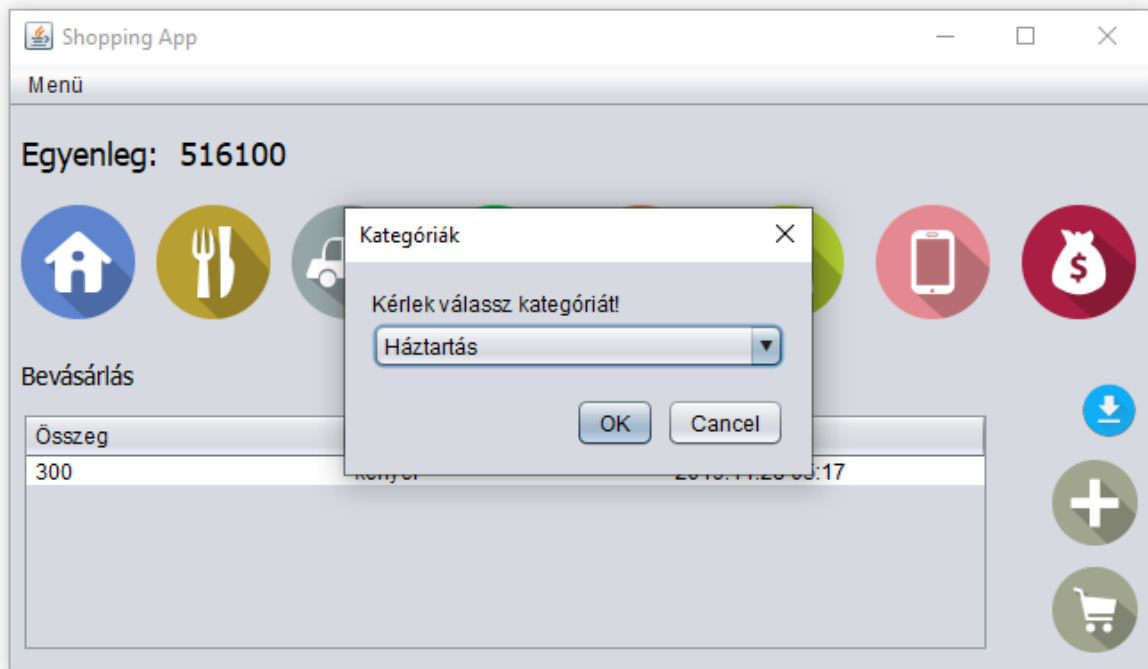
Az alkalmazás elindítása után az alábbi kezdőképernyő fogad:



A felhasználóbarát kialakítás lehetővé teszi, hogy a felhasználó minden egyes funkciót könnyen elérjen a kezdőképernyőről. Az alkalmazásban található ikonok könnyedén érthetővé teszik a felhasználó számára, hogy milyen adatokat fog elérni az adott ikonra való kattintás során. A kiválasztott kategória megnevezése a megjelenő táblázat felett jól látható helyen jelzi az adott kategória nevét.



A kategória kiválasztása és az adatok megadása felugró ablakban történik.



Adatbázis konfiguráció

Az adatbáziskapcsolathoz szükséges adatok megadása, az application.properties fájlban történik:

```
#spring.datasource.url=jdbc:h2:mem:testdb;DB_CLOSE_ON_EXIT=FALSE
```

```
#spring.datasource.driverClassName=org.h2.Driver
```

```
#spring.datasource.username=sa
```

```
#spring.datasource.password=
```

```
#spring.jpa.hibernate.ddl-auto=create-drop
```

```
#spring.jpa.show-sql=true
```

```
#spring.h2.console.enabled=true
```

```
#spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
```

spring.datasource.url=jdbc:mysql://localhost:3306/shopping_data

#spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=root

spring.datasource.password=

spring.jpa.hibernate.ddl-auto=update

Jelenleg XAMPP szerveren futtatott MySQL adatbázishoz van kapcsolva, de tetszőlegesen ez megváltoztatható. Beállítható az alapértelmezett adatbázisra is, ilyenkor vegyük ki a # komment jeleket a használni kívánt sorok elől és kommentezzük ki helyette a MySQL kapcsolatot. A JAVA Spring keretrendszer az indításkor automatikusan létrehozza a táblákat és feltölti működéshez szükséges adatokkal.

A spring.jpa.hibernate.ddl-auto=update parancs azt adja meg, hogy ha létezik az adatbázis, akkor ne csináljon új üres táblákat.

Adatbázis felépítése

Táblák:

balance: Egyenleg értékének tárolására szolgál.

category: Kategórianevek tárolására szolgál.

transactions: Ebbe a táblába kerülnek a felvitt kiadások. Az egyes rekordokhoz tartozik egy kategória azonosító is, ami megadja, hogy mely kategóriába lett rögzítve a kiadás.

hibernate_sequence: Sprint keretrendszer által generált segéd tábla

Alkalmazás beszerzése és indítása

Az alkalmazás publikusan elérhető a GitHub felületén.

<https://github.com/AdamButcher11/Shopping>

Amennyiben korábban már telepítettünk git klienst a számítógépünkre, az alkalmazás letölthető parancssor segítségével:

git clone <https://github.com/AdamButcher11/Shopping.git>

A futtatáshoz importálhatjuk a programot Netbeans projektként, vagy lehetőségünk van a Shopping/target mappában található shopping-1.0.jar fájl futtatására.

Felhasznált JAVA keretrendszer

Spring keretrendszer

A Spring egy nyílt forráskódú, inversion of controlt megvalósító Java alkalmazás keretrendszer.

A Spring keretrendszer magját képező szolgáltatásokat főként Java alkalmazás fejlesztésére használják a programozók. Ugyanakkor a Java EE platformra is elérhetők a bővítményei, amelyek web-alkalmazás fejlesztését segítik elő. Nem rendelkezik külön specifikált fejlesztési modellel, hanem az Enterprise JavaBean (EJB) modell kiegészítése-, helyettesítője-, vagy alternatívájaként vált népszerűvé a Java fejlesztők között.

A Spring keretrendszer több önálló modulból épül fel, amelyek az alábbi szolgáltatásokat nyújtják a fejlesztők számára:

- Inversion of control konténer: a Java objektumok életciklusának kezelése és az alkalmazás-komponensek testreszabása.
- Aspektus orientált programozási paradigma követésének lehetősége.
- Adatelérés: lehetőség van relációs adatbázis-kezelő rendszerek JDBC segítségével történő elérésre, és objektum-relációs lekérdezések, NoSQL integrálására.
- Tranzakciókezelés: többféle tranzakció kezelő API-t tartalmaz.
- Modell-nézet-vezérlő szabvány: egy HTTP- és servlet alapú keretrendszer segítségével valósítható meg, amelyet arra fejlesztettek ki, hogy bővíthetők és személyre szabhatóak legyenek a webszolgáltatások
- Távoli eljárashívás kezelő keretrendszer: biztosítja a RPC alapú, hálózaton keresztül történő Java objektum importokat és exportokat. További támogatást nyújt a RMI, a CORBA és HTTP alapú protokollok használatára, beleértve a webszolgáltatásokat (SOAP) is.
- Kötegelési eljárás támogatása.
- Azonosítás és azonosságkezelés: biztonsági folyamatok konfigurálása, melyet a Spring projekthez tartozó, Spring Security alprojekt tesz lehetővé a különféle protokollok és módszerek biztosításával.
- Üzenetkezelés: a JMS API-n keresztül történő általános üzenetkezelés továbbfejlesztése érhető el.
- Tesztelés: segítséget nyújt a unit- és az integrációs teszt írására.