

## Receptkönyv Szakdolgozat

Receptek szöveges állományainak feldolgozása és  
felhasználása asztali alkalmazással

Készítette: Tóth Krisztián

2/14 ED

Konzulens: Tóth József

|   |    |
|---|----|
| A felhasználói dokumentáció .....   | 3  |
| Célja:.....   | 3  |
| A program általános specifikáció: .....                                       | 3  |
| Rendszerkövetelmények .....   | 3  |
| Hardver követelmények: .....  | 3  |
| Szoftver követelmények: .....   | 3  |
| A program telepítése .....  | 5  |
| A program használatának részletes leírása .....                               | 6  |
| Fejlesztői dokumentáció .....   | 13 |
| Célja:.....   | 13 |
| Témaválasztás indoklása .....   | 13 |
| A program készítéséhez használt fejlesztői eszközök .....                     | 13 |
| Programozási nyelv .....  | 13 |
| Fejlesztői környezet.....   | 13 |
| Az adatbázis .....  | 14 |
| Az adatbázis felépítése:.....   | 14 |
| Osztályok: .....  | 16 |
| FXMLDocumentController.java osztály .....                                     | 16 |
| DB osztály .....  | 27 |
| Panel osztály .....   | 30 |
| Felhasznalo osztály.....  | 31 |
| Konstruktorok: .....  | 31 |
| Kedvenc osztály.....  | 31 |
| Recept osztály .....  | 32 |
| Receptkönyv.java főosztály .....  | 32 |
| Tesztelés .....   | 33 |
| Felugró hiba és információs ablakok a program felületei alapján tagolva:..... | 34 |
| Továbbfejlesztési lehetőségek.....  | 36 |
| Irodalomjegyzék, forrásmegjelölés .....                                       | 36 |
| A CD melléklet tartalma .....   | 36 |

# A felhasználói dokumentáció

## Célja:

A felhasználói dokumentáció célja, hogy segítséget adjon a leendő felhasználónak a programmal, telepítésével és használatával kapcsolatos minden probléma megoldásában, segítse a program kezelésének elsajátítását.

## A program általános specifikáció:

A szakdolgozatom célja egy olyan felület létrehozása, amely adatokat küld be, olvas ki és jelenít meg adatbázisból. Ez a program saját tapasztalatból származó ötlet alapján készül. A program segítségével ahelyett, hogy a papír alapú receptek helyett, amelyek könnyen elvesznek és sérülnek, modernebb egy felületen meg lehet jeleníteni bármelyik eltárolt receptet, szükség esetén módosítani és törölni. A programnak tudnia kell egy adatbázissal kapcsolatteremtés által az adatokat a programba betölteni. Ha a felhasználó olvasni, törölni vagy módosítani akarja a receptet, akkor ezt az programnak elérhetővé kell tennie. Több felhasználós a program, így egyszerre tudja több személy használni anélkül, hogy bárki véletlenül kárt okozna másnak valamely adatában. A program személyi, helyi számítógépre készül, így elegendő egy helyi adatbáziskezelő program. Egy olyan felületnek kell lennie, mely könnyen értelmezhető szöveget, mondatokat és felhasználást tesz lehetővé.

## Rendszerkövetelmények

### Hardver követelmények:

Java SE 8u241 szükséges

Xamp server 7.4.3

PHPMyAdmin 7.4.3

### Szoftver követelmények:

Windows:

Windows 10

Windows 8

Windows 7

Windows Vista

Windows Server 2008

Memória: 128 MB szükséges.

Processzor: legalább egy Pentium 2 266 MHz processzort.

Böngésző: Firefox.

OS X:

Intel-based Mac running Mac OS x 10.8.3+,10.9+

Böngésző: Safari.

Linux:

Ubuntu Linux 15.10

Ubuntu Linux 15.04

Ubuntu Linux 14.x

Ubuntu Linux 12.04 LTS, 13.x

Suse Linux Enterprise Server 12.x (64-bit)2 (8u20)

Suse Linux Enterprise Server 10 SP2+, 11.x

Red Hat Enterprise Linux 7.x (64-bit)2 (8u20)

Red Hat Enterprise Linux 5.5+1, 6.x(32-bit), 6.x(64-bit)2

Oracle Linux 7.x (64-bit)2 (8u20)

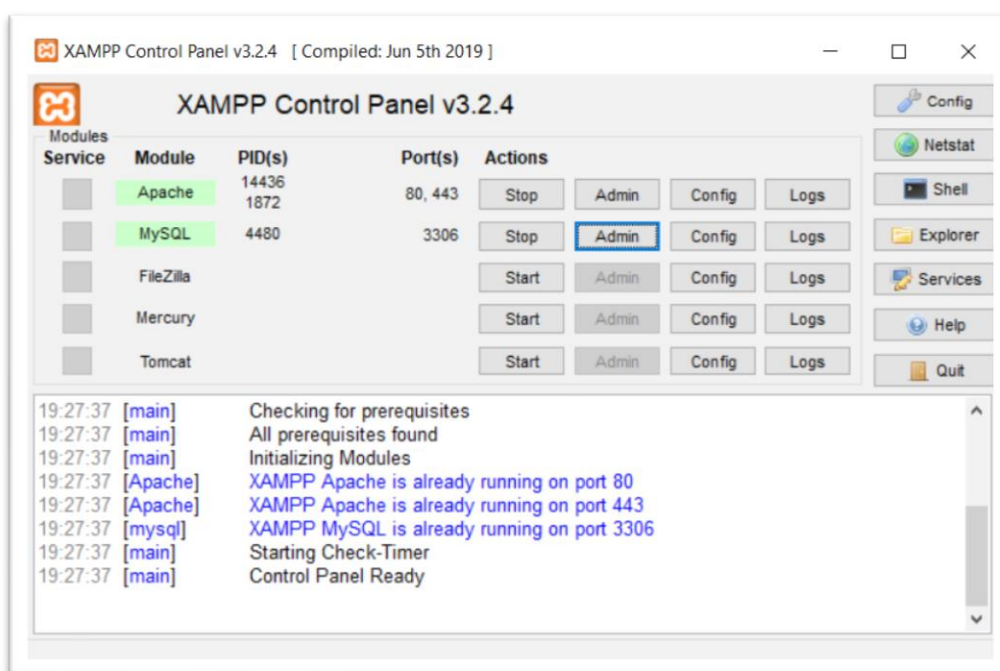
Oracle Linux 6.x(32-bit), 6.x(64-bit)2

Oracle Linux 5.5+1

## A program telepítése

A program nem rendelkezik külön telepítő programmal, ezért manuálisak kell a számítógépre másolni a szükséges fájlokat. A program használatához szükséges fájlokat, a „Telepítő” mappa tartalmát helyezzük el a számítógépen tetszőleges helyre. A mappából az XAMP adatbáziskezelő szoftver telepítőjével vagy internetről beszerezve hajtsuk végre az adatbáziskezelő telepítését. Kattintsunk az „Admin” gombra, a felugró weboldalon lépünk be a „root” felhasználónévvel. Majd belépés után kattintsunk az importálásra és válasszuk ki a „telepítő mappából a receptkonyv.sql fájlt és indítsuk el az importálást. Majd a „sugo” mappát fel kell másolni a C: meghajtóra az xampp mappán belül a htdocs mappába.

Majd a mappából az XAMP adatbáziskezelő szoftver Control Panel felületén az Apache és a MySQL motult szükséges elindítani a felső két Start gomb lenyomásával a számítógép minden elindításakor (1. ábra). Ezután a „receptkönyv” fájlra kattintva indul a program.



1. ábra Az Xamp Control Panel

## A program használatának részletes leírása

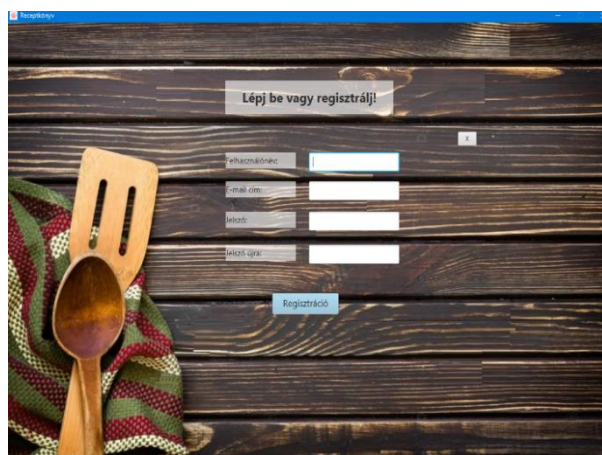
A program leírása funkciók szerint sorban haladva valósul meg.

A program indításakor a "Lépj be vagy regisztrálj" feliratú főoldal jelenik meg (2. ábra). Itt lehetőség van regisztrálni vagy bejelentkezni a "Regisztráció" és a "Belépés" gombokra kattintva.



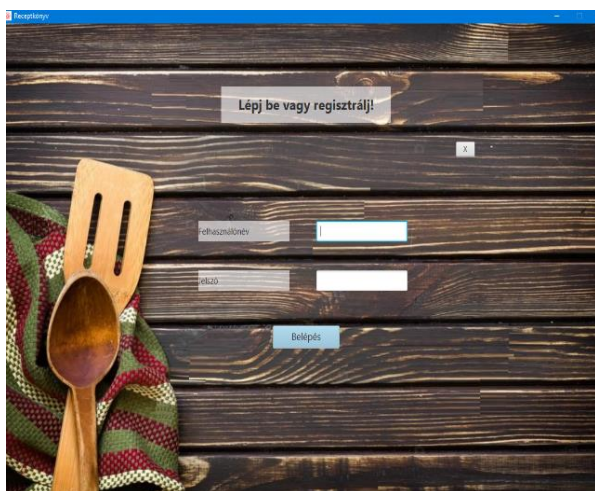
2. ábra "Lépj be vagy regisztrálj"

Ha a "Regisztráció" gombra kattintunk a program láthatóvá teszi a regisztrációs adatokat bekérő üres szövegmezőket (3. ábra). Itt meg kell adni egy új felhasználónevet, e-mail címet, jelszót és a jelszót még egyszer. Ha a jelszó vagy a felhasználónév karaktereinek száma kevesebb mint 5, akkor a program felugró ablakkal jelzi a hibát "A felhasználónévnek minimum 5 karakter hosszúnak kell lennie!", "A felhasználónévnek maximum 40 karakter hosszúnak kell lennie!" és "A jelszónak minimum 5 karakter hosszúnak kell lennie!" szöveggel. Ha a jelszavak szövege nem egyezik akkor is felugró ablakkal jelzi a hibát "Az egyik jelszót rosszul adtad meg!" szöveggel. Ha már van felhasználói fiókunk és mégsem szeretnénk regisztrálni, akkor az "X" gombra kattintva újra a "Regisztráció" és a "Belépés" lehetőség lép fel. De ha mégis regisztrálni akarna a felhasználó és rákattint a "Regisztráció" gombra a program elmenti a megadott adatokat, és létrehozza a felhasználó új fiókját és betölti a program kezelőfelületét.

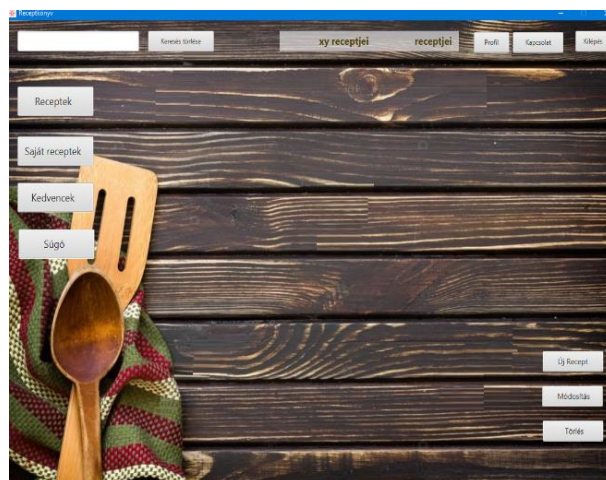


3. ábra Regisztrációs felület

Ha a "Belépés" gombra kattintunk a program láthatóvá teszi a belépéshez szükséges adatokat bekérő üres szövegmezőket (4. ábra). Itt meg kell adni egy a már létező felhasználónevet és jelszót. Ha a felhasználónév hibásan lett megadva, akkor a program felugró ablakkal jelzi a hibát "Nincs ilyen nevű felhasználó!", ha a jelszó lett rosszul megadva akkor azt is felugró ablakkal jelzi a program "Rossz jelszó!" szöveggel. Ha még nincsen felhasználói fiókunk és szeretnénk regisztrálni, akkor az "X" gombra kattintva újra a "Regisztráció" és a "Belépés" lehetőség lép fel. De ha mégis bejelentkezni akarna a felhasználó és rákattint a "Belépés" gombra a program leellenőrzi az adatokat, és ha minden megfelelő, akkor betölti a program kezelőfelületét (5. ábra).



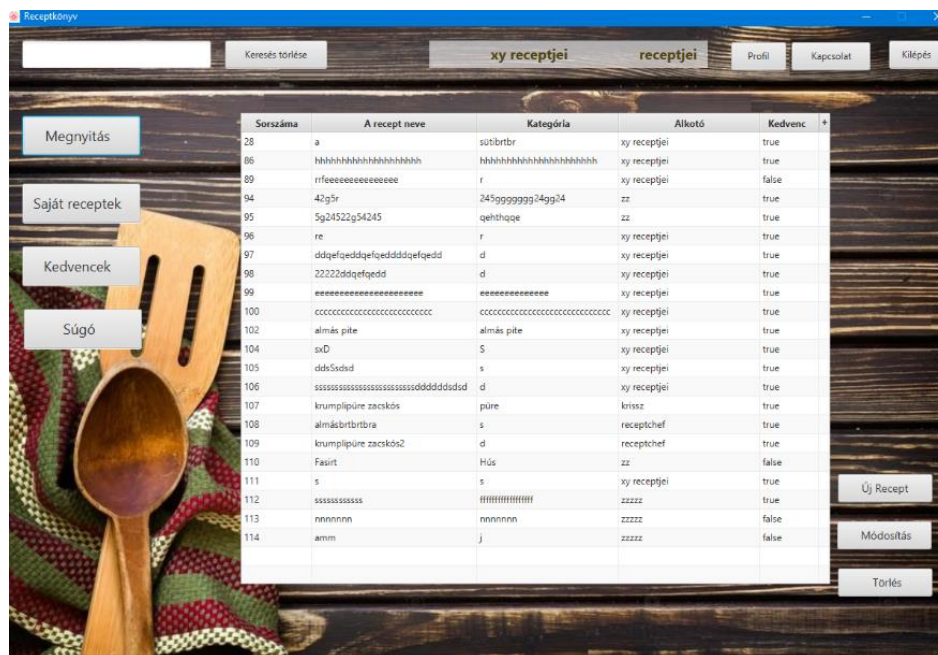
4. ábra Belépés felülete



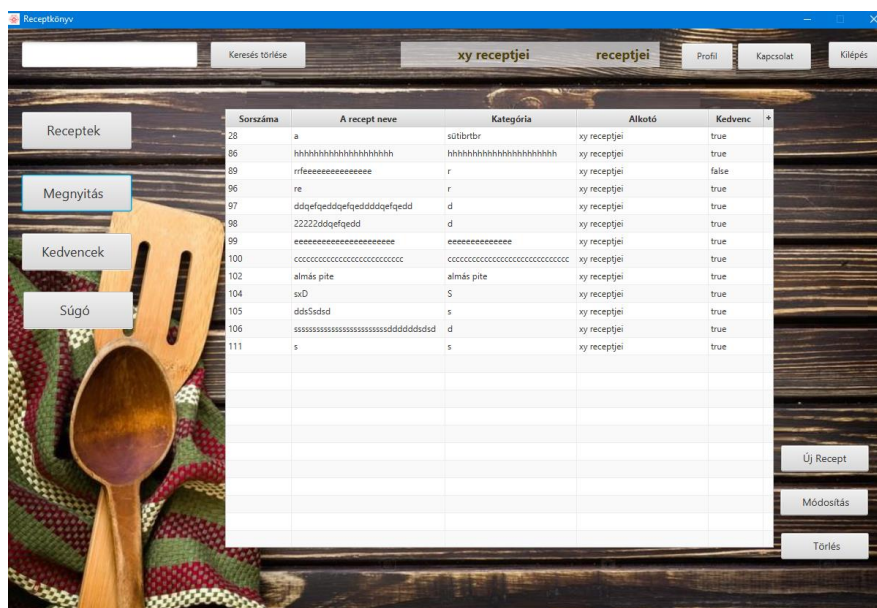
5. ábra A program kezelő felülete

A "Receptek" gombra kattintva megjelenik az összes recept táblázatba rendezve (6. ábra). A "Recept" gomb felirata "Megnyitás"-ra módosul. Ha kilyelölünk egy sort a táblázatban egy kattintással és kattintunk a "Megnyitás" gombra a táblázat eltűnik és egy teljes receptet megjelenítő lap a recept nevével, kategóriájával, hozzávalóival és elkészítésének szövegével jelenik meg. Itt lehetőségünk van a receptet elolvasni, a "Recept másolás" gombra kattintva a recept teljes szövegét a vágólapra másolja, így tetszőlegesen felhasználható. Ha a "Kiírás txt-be" gombra kattintunk felugrik egy rendszerablak melyben a file nevét, kiterjesztését(például:txt.) és új helyét megadva új file-ba menthetjük a megnyitott receptet. A "kedvenc" jelölő négyzetben kedvencé tehetjük az adott receptet, vagy éppen levehetjük róla a kedvenc jelölést. A "Receptek" gombra kattintva újra az összes receptet jeleníti meg táblázatba rendezve, és a "Megnyitás" gomb felirata "Receptek"-re módosul.





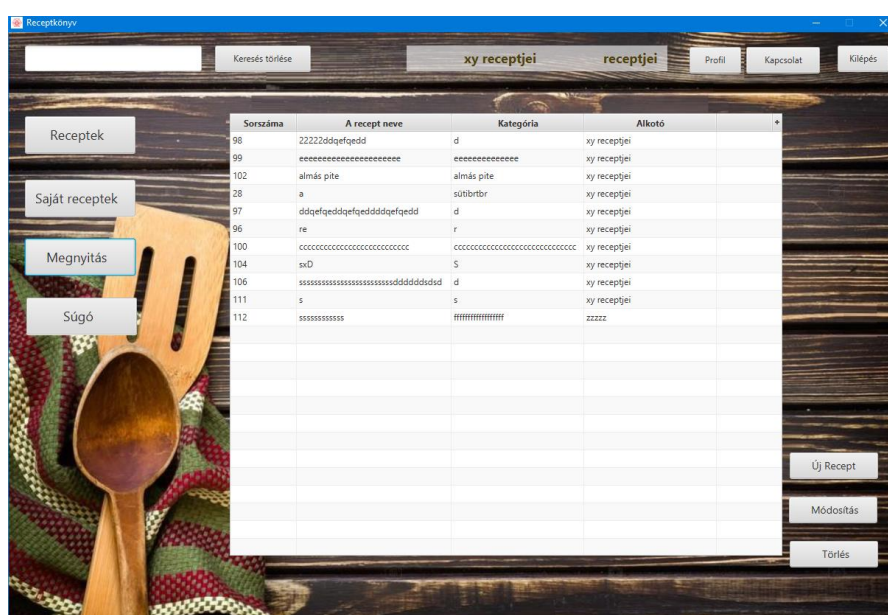
A "Saját receptek" gombra kattintva megjelenik az összes saját recept amit a felhasználó ebben a felhasználói fiókban hozott létre táblázatba rendezve (7. ábra). A "Saját receptek" gomb felirata "Megnyitás"-ra módosul. Ha kilyelölünk egy sort a táblázatban egy kattintással és kattintunk a "Megnyitás" gombra a táblázat eltűnik és egy teljes receptet meglyelénítő lap a recept nevével, kategóriályával, hozzávaóival és elkészítésének szövegével jelenik meg. Itt lehetőségünk van a receptet elolvasni, a "Recept másolás" gombra kattintva a recept teljes szövegét a vágólapra másolja, így tetszőlegesen felhasználható. Ha a "Kiírás txt-be" gombra kattintunk felugrik egy rendszerablak melyben a file nevét, kiterjesztését(például:txt.) és új helyét megadva új file-ba menthetjük a megnyitott receptet. A "kedvenc" jelölő négyzetben kedvencé tehetjük az adott receptet, vagy éppen levehetjük róla a kedvenc jelölést. A "Saját receptek" gombra kattintva újra az összes saját receptet jeleníti meg táblázatba rendezve, és a "Megnyitás" gomb felirata "Saját receptek"-re módosul.





### 7. ábra Saját receptek tábla

A "Kedvencek" gombra kattintva megjelenik az összes kedvencnek jelölt recept táblázatba rendezve (8. ábra). A "Kedvencek" gomb felirata "Megnyitás"-ra módosul. Ha kilyelölünk egy sort a táblázatban egy kattintással és kattintunk a "Megnyitás" gombra a táblázat eltűnik és egy teljes receptet meglyelénítő lap a recept nevével, kategóriályával, hozzááoival és elkészítésének szövegével jelenik meg. Itt lehetőségünk van a receptet elolvasni, a "Recept másolás" gombra kattintva a recept teljes szövegét a vágólapra másolja, így tetszőlegesen felhasználható. Ha a "Kiírás txt-be" gombra kattintunk felugrik egy rendszerablak melyben a file nevét, kiterjesztését(például:txt.) és új helyét megadva új file-ba menthetjük a megnyitott receptet. A "kedvenc" jelölő négyzetben kedvencé tehetjük az adott receptet, vagy éppen levehetjük róla a kedvenc jelölést. A "Kedvencek" gombra kattintva újra az összes kedvenc receptet jeleníti meg táblázatba rendezve, és a "Megnyitás" gomb felirata "Kedvencek"-re módosul.



8. ábra A kedvencek tábla

A “Súgó gombra kattintva a program egy internetes böngésző ablakot nyit meg, és betölti a program “sugo” oldalát. Itt két opció közül választhatunk. Az első a program használatát részletezi, a második a telepítés részleteit írja le és illusztrálja.

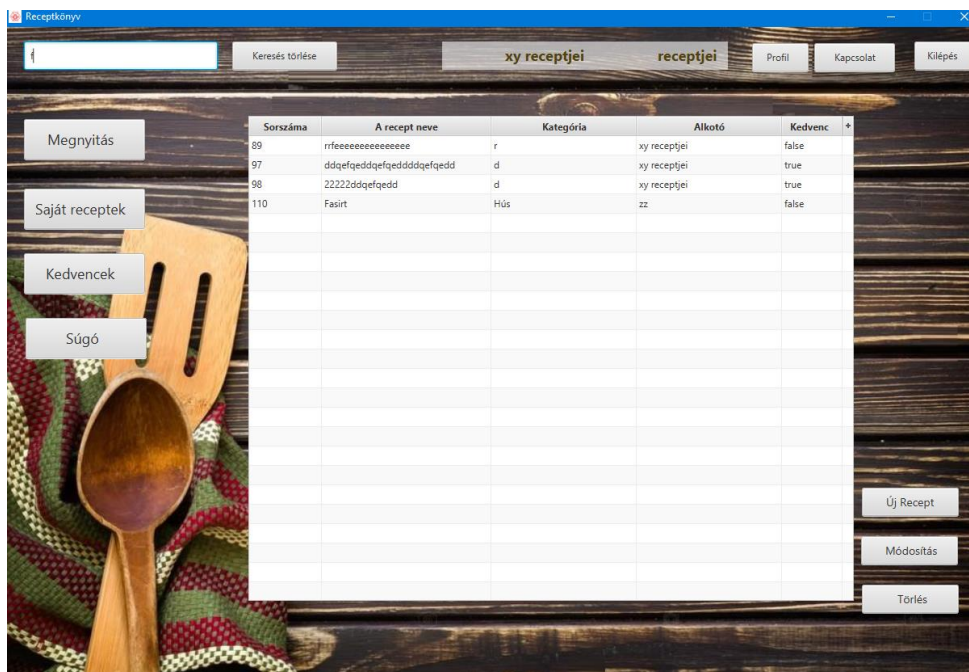
A "Új recept" gombra kattintva megjelenik a recept megjelenítő lap (9. ábra), melyen üres szövegmezők a recept nevét, karakterét, hozzávalóit és elkészítésének a szövegét kéri be. Minden mezőt kötelező kitölteni. A "kedvenc" jelölő négyzetben kedvencé tehetjük az adott receptet, vagy éppen levehetjük róla a kedvenc jelölést. Ha egy már létező recept nevét írjuk az új recept nevéhez, akkor felugró ablak "Már van ilyen recept!", ha legalább egy mező üres maradt a program felugró ablakban hibaüzenetet küld "Kihagytál egy mezőt!" szöveggel. Ha mégsem akarsz új receptet hozzáadni, akkor a "Vissza" gombra kattintva visszalép a receptek táblázathoz. Ha a "Hozzáad" gombra kattintunk a gomb feliratát visszaváltoztatja "Új recept"-re. És kilép a receptek táblázathoz.



10. ábra Törlés gombra kattintva

A kereső mezőben a "Receptek" a "Saját receptek" vagy a "Kedvencek" gomb megnyomása után a receptek nevével, de a nevük részletével(például: almás pite=almás ,vagy pite,stb) is lehet keresni. Betűlenyomásra automatikusan keres (11. ábra).

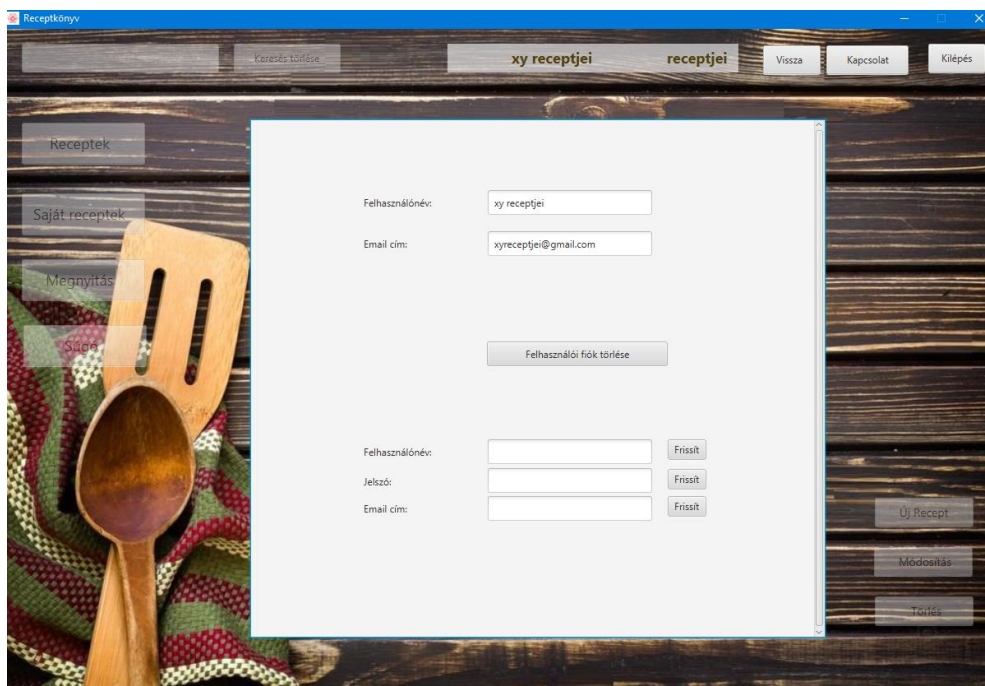
A "Keresés törlése" gombon kattintva azonnal törlődik a keresést és a keresés mező tartalmát.



### 11. ábra Keresés

A "Profil" gombra kattintva a gomb felirata "Vissza"-ra változik, egy adatkiíró lapon a lap tetején a felhasználó adatait a felhasználónevet és az e-mail címet írja ki (12. ábra). A "Fiók törlése" gombra kattintva felugró ablakban megkérdezi, hogy "Biztosan törölni szeretnéd a fiókodat?", a "nem" gombra kattintva vagy a felugró ablak felső sarkában az X re kattintva nem történik semmi, az "igen" gombra kattintva törli a felhasználó fiókját és kiléptet a "Lépj be vagy regisztrálj" feliratú főoldalra. Ezután már nem lesz meg a fiókod! A lap alján a felhasználónevet a jelszót és az e-mail címet lehet megváltoztatni a szövegbeviteli mezőkbe írva az új értékeket. Külön módosíthatók a "Frissít gombra kattintva. Ha 5 karakternél rövidebb a jelszó, vagy a felhasználónév 5 karakternél rövidebb vagy 40 karakternél hosszabb, akkor felugró ablakban "A felhasználónévnek/jelszónak legalább 5 karakter hosszúnak kell lennie!", ha sikerül a változtatás felugró ablakban értesít "A felhasználónév/jelszó/e-mail cím sikeresen megváltozott". A "Vissza" gombra kattintva újra használhatóvá válik a kezelőfelület és a gomb szövege visszaváltozik "Profil"-ra.

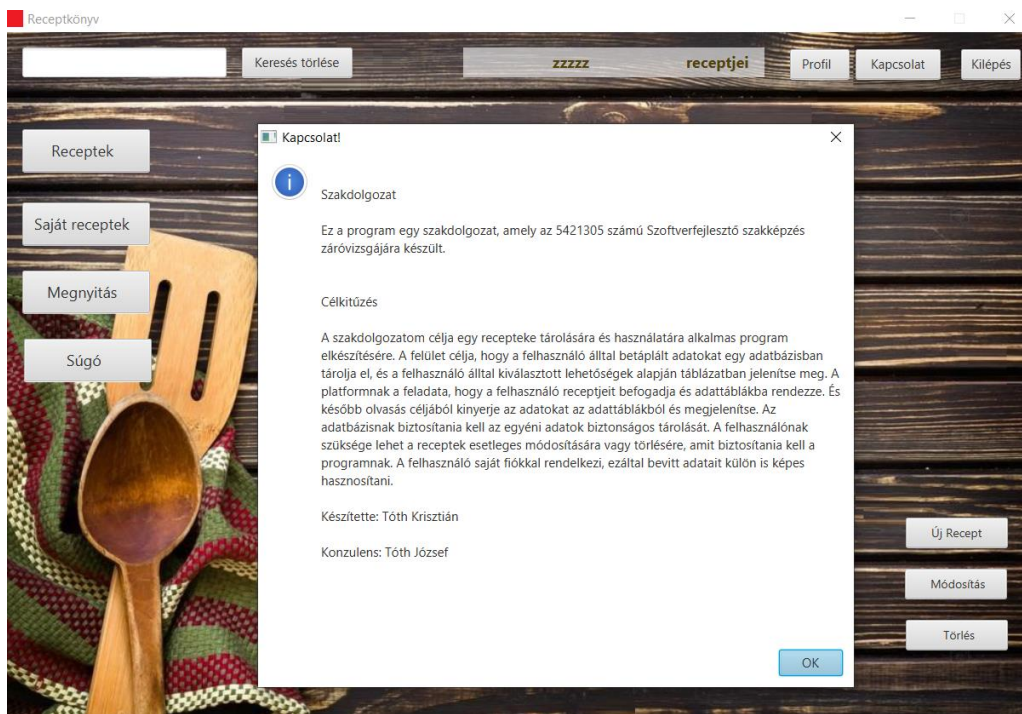




12. ábra Profil

A felhasználó neve a "Profil" gomb melletti szürke sávban olvasható.

A "Kapcsolat" gombra kattintva a kapcsolati adatok találhatóak egy felugró ablakban (13. ábra). Az "ok" vagy a felugró ablak felső sarkában az "x"-re kattintva léphetünk ki az ablakból.



13. ábra Kapcsolat ablak

A "Kilépés" gombra kattintva a program kilép a receptek kezelőfelületéről "Lép be vagy regisztrálj" feliratú főoldalra, ahonnan újra be lehet lépni vagy lehet új felhasználói fiókot létrehozni.

## Fejlesztői dokumentáció

### Célja:

A fejlesztői dokumentáció célja, hogy a segítse a program logikájának, illetve a program kódjának a megértését, illetve a program továbbfejlesztését. Szakemberek számára készül, elvárás tehát a pontosság és a szakmai jellegű stílus.

### Témaválasztás indoklása

A témája saját ötletből származik. Azért választottam a receptekkel foglalkozó program elkészítését, mert az a tapasztalatom, hogy sok ember nem tapasztalt a konyhai alkalmazkodás terén. Aki pedig meg akar tanulni főzni, vagy már valamennyire tud annak is szüksége van receptekre. Ezeket általában receptkönyvekben vagy weboldalakról lehet könnyen beszerezni. Ezek jók, de az a baj velük, hogy egy idő után túlságosan felhalmozódnak, könnyen sérülnek papír formában, elvész a többi között, nehéz megtalálni. Ezt teszi könnyebbé ez a program úgy, hogy a receptek között lehet a receptek egész- vagy részleges nevével keresni. Saját és kedvenc recepteket külön használni.

## A program készítéséhez használt fejlesztői eszközök

### Programozási nyelv

A program írásához a Java programozási nyelvet használtam, ez teljesen megfelelt a program minden részleténél. A kódok a program projekt mappájában találhatók csomagokba rendezve. Az adattároláshoz SQL nyelven használható adatbázis lett felhasználva.

Alkalmazott fejlesztői eszközök:

NetBeans IDE 8.2

Xamp server 7.4.3

PHPMyAdmin 7.4.3

Java SE 8u241 szükséges

### Fejlesztői környezet

A program java kódolása a Netbeans IDE 8.2-es verziójában lett elkészítve. A program grafikus felületének kódja a JavaFX Scene Builder 8.5.2-es verziójával készült. Az adatok kezeléséhez és tárolásához az XAMP szoftver 7.4.3-0-ás verzióját alkalmazza. A dokumentációban lévő képeket a Windows rendszer képszerkesztőjével készült.

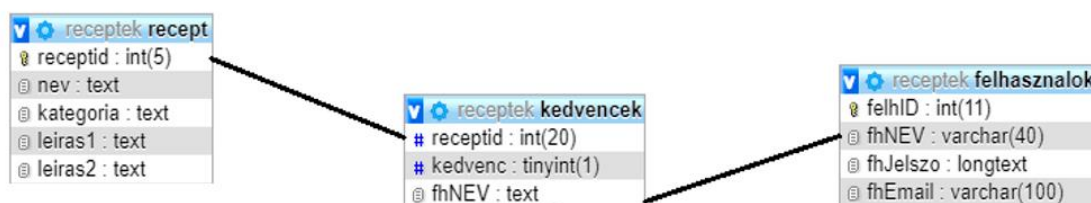
## Az adatbázis

A program az adatok tárolását adatbázisban teszi lehetővé (14. ábra). A program által használt adatbáziskezelő szoftver az xamp.

Egy szabad és nyílt forrású platformfüggetlen webservert-szoftvercsomag, amelynek legfőbb alkotóelemei az Apache webservert, korábban a MySQLadatbázis-kezelő, valamint a PHP és a Perl programozási nyelvek értelmezői (végrehajtó rendszerei). Az XAMPP tervezői az eszközt egy fejlesztőrendszernek szánták, amellyel a web-tervezők és programozók internetes kapcsolat nélkül fejleszthetik és tesztelhetik alkalmazásaikat.

Az adatok egymással összekapcsolt táblákban vannak. Kezelésüket adatbázis-kezelő rendszerek segítségével végezhetjük el. Szerver-kliens alapú adatbázis-kezelő rendszer a MySQL, míg a számítógép fájlrendszerét éri el a Microsoft Access és az SQLite.

Az adatbázis felépítése:



14. ábra receptkonyv adatbázisban a kapcsolati felépítések

Az adatbázis neve: receptkonyv

Adattáblák:

A receptkonyv adatbázis három adattáblából épül fel.

1.: recept tábla

A recept adattábla hat oszlopból épül fel (16. ábra). Ez az adattábla tárolja a recepteket. A receptid, amely a receptek sorszámát adja meg. Az adattáblában ez az elsődleges kulcs (PRIMARY KEY). Ez automatikusan hozza létre az adatait, így minden új sorban növekszik az adat értéke. A nev, a recept nevét, a kategória a recept minősítését például:(húsétel, püré, szósz), a leiras1 a recept hozzávalóit, a leiras2 a recept elkészítését, az alkotó a receptet elkészítő felhasználó nevét tartalmazza.



| #                          | Név         | Típus  | Illesztés            | Tulajdonságok | Nulla | Alapértelmezett | Megjegyzések | Extra          |
|----------------------------|-------------|--------|----------------------|---------------|-------|-----------------|--------------|----------------|
| <input type="checkbox"/> 1 | receptid 🗝️ | int(5) |                      |               | Nem   | Nincs           |              | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | nev         | text   | utf8mb4_hungarian_ci |               | Nem   | Nincs           |              |                |
| <input type="checkbox"/> 3 | kategoria   | text   | utf8mb4_hungarian_ci |               | Nem   | Nincs           |              |                |
| <input type="checkbox"/> 4 | leiras1     | text   | utf8mb4_hungarian_ci |               | Nem   | Nincs           |              |                |
| <input type="checkbox"/> 5 | leiras2     | text   | utf8mb4_hungarian_ci |               | Igen  | NULL            |              |                |
| <input type="checkbox"/> 6 | alkoto      | text   | utf8mb4_hungarian_ci |               | Nem   | Nincs           |              |                |

15. ábra A recept tábla

## 2.: felhasználók tábla

A felhasználók adattábla a felhasználók főbb adatait tartalmazza négy oszlop (16. ábra). Ebben a táblában az elsődleges kulcs (PRIMARY KEY) felhID, amely a felhasználó azonosítóját, az fhNEV a felhasználó nevét, az fhJelszo a felhasználó jelszavát, az fhEmail a felhasználó e-mail címét tárolja.

| #                          | Név       | Típus        | Illesztés            | Tulajdonságok | Nulla | Alapértelmezett | Megjegyzések | Extra          |
|----------------------------|-----------|--------------|----------------------|---------------|-------|-----------------|--------------|----------------|
| <input type="checkbox"/> 1 | felhID 🗝️ | int(11)      |                      |               | Nem   | Nincs           |              | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | fhNEV     | varchar(40)  | utf8mb4_hungarian_ci |               | Nem   | Nincs           |              |                |
| <input type="checkbox"/> 3 | fhJelszo  | longtext     | utf8mb4_hungarian_ci |               | Nem   | Nincs           |              |                |
| <input type="checkbox"/> 4 | fhEmail   | varchar(100) | utf8mb4_hungarian_ci |               | Nem   | Nincs           |              |                |

16. ábra felhasználók tábla

## 3.: kedvencek tábla

A kedvencek adattábla három oszlopban tárolja el a kedvelt receptek azonosításához szükséges adatait (17. ábra). A receptid a recept azonosítóját, a kedvenc a „true” (1) értéket, az fhNEV a felhasználó nevét.

| #                          | Név      | Típus      | Illesztés            | Tulajdonságok | Nulla | Alapértelmezett | Megjegyzések | Extra |
|----------------------------|----------|------------|----------------------|---------------|-------|-----------------|--------------|-------|
| <input type="checkbox"/> 1 | receptid | int(20)    |                      |               | Nem   | Nincs           |              |       |
| <input type="checkbox"/> 2 | kedvenc  | tinyint(1) |                      |               | Nem   | Nincs           |              |       |
| <input type="checkbox"/> 3 | fhNEV    | text       | utf8mb4_hungarian_ci |               | Nem   | Nincs           |              |       |

17. ábra kedvencek tábla

Osztályok:

*FXMLDocumentController.java osztály*

Importálások:

```
import java.awt.Desktop;
import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import java.net.URI;
import java.net.URISyntaxException;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Button;
import javafx.scene.control.CheckBox;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.ScrollPane;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextArea;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.image.ImageView;
import javafx.scene.input.Clipboard;
import javafx.scene.input.ClipboardContent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.HBox;
import javafx.scene.layout.Pane;
import javafx.stage.FileChooser;
```

Osztályváltozók:

```
DB ab = new DB();  
private boolean profil = true;  
private boolean modosit = true;  
private boolean hozzaad;  
private boolean leüt = true;  
private boolean koppint = true;  
private boolean megnyom = true;  
char[] betuk = { };
```

FXML elemek:

```
@FXML private Pane fopane;  
@FXML private AnchorPane bellsoancorpaneegy;  
@FXML private HBox lbevagyrhbox;  
@FXML private PasswordField bejelentkezésbe;  
@FXML private TextArea ahozzavalokszovegearea;  
@FXML private ScrollPane scrollpane1recept;  
@FXML private TextArea areceptszovegearea;  
@FXML private TextField txtfelhasznalonevre;  
@FXML private AnchorPane bellsoancorpaneketto;  
@FXML private AnchorPane regisztracioancorpane;  
@FXML private AnchorPane belepesancorpane;  
@FXML private TextField txtfelhasznalobe;  
@FXML private TextField txtkeres;  
@FXML private TextField ujreceptnev;  
@FXML private AnchorPane scrollpaneancorpane2;  
@FXML private AnchorPane scrollpaneancorpane1;  
@FXML private PasswordField jelszore;  
@FXML private PasswordField jelszoujrare;  
@FXML private ImageView receptkepeimg;  
@FXML private TextField txtemailcimre;  
@FXML private TextField ujkategoria;
```

```

@FXML private TableView<Recept> tbllista;
@FXML private TableColumn<Recept, Integer> tablecrceptid;
@FXML private TableColumn<Recept, String> tablecreceptnev;
@FXML private TableColumn<Recept, String> tableckategoria;
@FXML private TableColumn<Recept, String> tablecalkotoja;
@FXML private TableColumn<Recept, Boolean> tableckedvenc;
@FXML private Button xgombid;
@FXML private Label lblfhnev;
@FXML private Button receptekgombid;
@FXML private Button sajatreceptekgombid;
@FXML private Button receptmodositasagombid;
@FXML private Label receptnevfelirat;
@FXML private Label kategoriafelirat;
@FXML private Label hozzavalokfelirat;
@FXML private Label receptszovegfelirat;
@FXML private Button ujreceptgombid;
@FXML private TableView<Recept> tbllista1;
@FXML private TableColumn<Recept, Integer> tablecrceptid1;
@FXML private TableColumn<Recept, String> tablecreceptnev1;
@FXML private TableColumn<Recept, String> tableckategoria1;
@FXML private TableColumn<Recept, String> tablecalkotoja1;
@FXML private TableColumn<Recept, Boolean> tableckedvenc1;
@FXML private TableView<Kedvenc> tbllista2;
@FXML private TableColumn<Kedvenc, Integer> tablecrceptid2;
@FXML private TableColumn<Kedvenc, String> tablecreceptnev2;
@FXML private TableColumn<Kedvenc, String> tableckategoria2;
@FXML private TableColumn<Kedvenc, String> tablecalkotoja2;
@FXML private Button kimasolgomb;
@FXML private CheckBox kedvid;
@FXML private Button kedvencekgombid;
@FXML private Button visszagombid;
@FXML private Button kimasolgombid;

```

```

@FXML private ScrollPane scrollpane2felhasznaloiadatok;
@FXML private TextField felhkapcsolat;
@FXML private TextField emailkapcsolat;
@FXML private Button profilgombid;
@FXML private Button recepttorlesegombid;
@FXML private Button sugogombid;
@FXML private Button keresestorlesegombid;
@FXML private Button kapcsolatgombid;
@FXML private TextField felhkapcsolat1;
@FXML private TextField jelszokapcsolat1;
@FXML private TextField emailkapcsolat1;
@FXML private Button exporid;

```

---

#### Metódusok:

@FXML public void felhasznaloifioktorlese(): Az lblfhnev FXML elem szövegét egy felhasználó(String) változóba tölti. Ha a meghívott Panel osztály igennem metódusa igazat ad vissza, akkor meghívja a DB osztály fioktorles metódust és a felhasználó változót küldi paraméterként. A profil osztályváltozót igazra állítja és meghívja a kilepesgomb metódust. Egyébként kilép a metódusból.

@FXML public void kapcsolatgomb(): Meghívja kattintáskor a Panel osztály kapcsolat metódusát és paraméterként küldi a kapcsolat lapra kiírandó szöveget.

private void kattinthatosag(boolean f): Inaktivitását(Disable) értékét állítja be a receptekgombid, sajátreceptekgombid, kedvencekgombid, receptmodositasagombid, recepttorlesegombid, ujreceptgombid, sugogombid, keresestorlesegombid, txtkeres, kilepesgombid FXML elemeknek a paraméterként kapott f(boolean) változó szerint.

@FXML public void profilgomb(): Ha a profil osztályváltozó értéke igaz, akkor a tbllista, tbllista1, tbllista2, scrollpane1recept FXML elemek láthatóságát(Visible) hamisra, a scrollpane2felhasznaloiadatok FXML elem láthatóságát igazra állítja. Meghívja a kattinthatosag metódust és a profil osztályváltozó értékét küldi paraméterként. Egy tomb(String) tömbben meghívja a DB osztály feltolt metódusát és paraméterként küldi az lblfhnev FXML elem értékét. A felhkapcsolat és az emailkapcsolat FXML elemek értékeit a tomb változó megfelelő értékére és szerkeszthetőségüket hamisra, a profilgombid szövegét ("Vissza") -ra, a profil osztály változót hamisra állítja. Egyébként meghívja a kattinthatosag metódust és a profil osztályváltozó értékét küldi paraméterként, a profilgombid szövegét ("Profil") -ra, a scrollpane2felhasznaloiadatok FXML elem láthatóságát hamisra, a profil osztály változót igazra állítja és meghívja a beolvas metódust.

@FXML public void regisztraciosoldalGomb(): Az xgombid, regisztracioancorpane FXML elemek láthatóságát igazra, az lbevagyrhbox, belepesancorpane, bellsoancorpaneketto FXML elemek láthatóságát hamisra állítja.

@FXML public void belepesoldalGomb(): Az xgombid, belepesancorpane láthatóságát igazra, az lbevagyrhbox, regisztracioancorpane, bellsoancorpaneketto láthatóságát hamisra állítja.

@FXML public void regisztracioGomb(): A txtfelhasznalonevre, txtemailcimre, txtjelszore, txtjelszujrare Stirng változóba a txtfelhasznalonevre, txtemailcimre, jelszore, jelszoujrare FXML elemek értékeit feltölti. Ha valamelyik változó szövege üres, ha a két jelszó nem egyenlő, ha a DB osztályból meghívott felhasznalonevvizsgalat melynek paraméterként küldött txtfelhasznalonevre változó igazat ad vissza, ha a felhasználónév 5 karakternél rövidebb vagy 40 karakternél hosszabb, ha a jelszó 5 karakternél rövidebb, akkor a Panel osztály hiba metódusával üzenetet küld a megfelelő hibaüzenettel. Egyébként a DB osztály felhasználohozzaad metódusát meghívja és paraméterként küldi a változókat. A bellsoancorpaneketto FXML elem láthatóságát igazra, a scrollpane2felhasznaloiadatok, bellsoancorpaneegy, scrollpane1recept, tbllista, tbllista1, tbllista2, kedvid láthatóságát hamisra, az lblfhnev szövegét a txtfelhasznalonevre változó értékére, a kurzor fókuszát a txtkeres FXML elemre állítja. A txtfelhasznalonevre, txtemailcimre, jelszore, jelszoujrare FXML elemek értékét törli.

@FXML public void belepesGomb(): A txtfelhasznalobe és txtbelepesbe Stirng változóba a txtfelhasznalobe és bejelentkezésbe FXML elemek értékeit feltölti. Ha valamelyik változó szövege üres, ha a DB osztályból meghívott felhasznalonevvizsgalat melynek paraméterként küldött txtfelhasznalobe változó hamisat ad vissza, ha a DB osztályból meghívott jelszovizsgalat mely paraméterként a txtfelhasznalobe hamisat ad vissza, akkor a Panel osztály hiba metódusával üzenetet küld a megfelelő hibaüzenettel. A bellsoancorpaneketto FXML elem láthatóságát igazra, a scrollpane2felhasznaloiadatok, bellsoancorpaneegy, scrollpane1recept, tbllista, tbllista1, tbllista2 láthatóságát hamisra, az lblfhnev szövegét a txtfelhasznalobe változó értékére, a kurzor fókuszát a txtkeres FXML elemre állítja. A txtfelhasznalobe és bejelentkezésbe FXML elemek értékét törli.

@FXML public void xgomb(): Az lbevagyrhbox FXML elem láthatóságát igazra, a regisztracioancorpane, belepesancorpane, xgombid FXML elemek láthatóságát hamisra állítja. A txtfelhasznalonevre, txtemailcimre, jelszore, jelszoujrare, txtfelhasznalobe, bejelentkezésbe FXML elemek tartalmát törli.

@FXML public void visszagomb(): A hozzaad osztályváltozót hamisra, az ujreceptgombid szövegét ("Új recept")-re, a sajátreceptekgombid, receptekgombid, kedvencekgombid, receptmodositasagombid, recepttorlesegombid, profilgombid, kilepesgombid inaktív(Disable) értéket hamisra, a visszagombid, kedvid és scrollpane1recept FXML elemek láthatóságát hamisra, a tbllista FXML elemek láthatóságát igazra állítja. Meghívja a beolvas, sajátbeolvas, kedvencekbeolvas és a receptekgomb metódusokat.

@FXML public void keresestorlesegomb(): A txtkeres FXML elem értékét törli.

private void szerkeszthetoseg(boolean igaze): Az ujreceptnev, ujkategoria, ahozzavalokszovegearea, areceptszovegearea FXML elemek szerkeszthetőségét(Editable) a paraméterként kért igaze(boolean) szerint állítja be.

private int figyelő(): Az index(int) változó értékét adja vissza a tbllista kijelölt elemének indexével.



private int figyelol(): Az index1(int) változó értékét adja vissza a tbllista1 kijelölt elemének indexével.

private int figyelol2(): Az index2(int) változó értékét adja vissza a tbllista2 kijelölt elemének indexével.

@FXML public void receptekgomb(): Ha a megnyom osztályváltozó igaz, akkor ha a figyelol metódus értéke nagyobb mint -1 a receptid, lblfhnev és a kedvid változókat feltölti. Ha a kedvid osztályváltozó igaz és a DB osztály kmevszamolvane metódusa hamisat ad vissza, a DB osztály kltrehoz metódusával feltölti az adatbázist. Ha a kedvid osztályváltozó hamis és a DB osztály kmevszamolvane metódusa igazat ad vissza, a DB osztály ktorol metódusával töröl az adatbázisban. A sajátreceptekgombid feliratát ("Saját receptek") -re, a kedvencekgombid feliratát ("Kedvencek") -re, a receptekgombid feliratát ("Megnyitás") -ra, a koppint és leüt osztályváltozót igazra, a megnyom osztályváltozót hamisra, a scrollpanelrecept, tbllista1, kimasolgombid, tbllista2, kedvid FXML elemek láthatóságát hamisra, a tbllista láthatóságát igazra állítja és meghívja a beolvas metódust. A szerkeszthetoseg metódust meghívja és paraméterének igazat küld. Egyébként, ha a figyelol osztályváltozó értéke nagyobb mint -1, akkor a receptid, lblfhnen és a kedvid változókat feltölti. Ha a kedvid osztályváltozó igaz és a DB osztály kmevszamolvane metódusa hamisat ad vissza, a kedvid FXML elem értékét hamisra, ha a kedvid osztályváltozó hamis és a DB osztály kmevszamolvane metódusa igazat ad vissza, a kedvid FXML elem értékét igazra állítja. A tbllista, tbllista1, tbllista2 FXML elemek láthatóságát hamisra, a kimasolgombid, kedvid és scrollpanelrecept láthatóságát igazra, a receptekgombid feliratát ("Receptek") -re, a megnyom osztályváltozót igazra állítja.

@FXML public void sajátreceptekgomb():Ha a koppint osztályváltozó igaz, akkor ha a figyelol metódus értéke nagyobb mint -1 a receptid, lblfhnen és a kedvid változókat feltölti. Ha a kedvid osztályváltozó igaz és a DB osztály kmevszamolvane metódusa hamisat ad vissza, a DB osztály kltrehoz metódusával feltölti az adatbázist. Ha a kedvid osztályváltozó hamis és a DB osztály kmevszamolvane metódusa igazat ad vissza, a DB osztály ktorol metódusával töröl az adatbázisban. A sajátreceptekgombid feliratát ("Saját receptek") -re, a kedvencekgombid feliratát ("Kedvencek") -re, a receptekgombid feliratát ("Megnyitás") -ra, a megnyom és leüt osztályváltozót igazra, a koppint osztályváltozót hamisra, a scrollpanelrecept, tbllista, kimasolgombid, tbllista2, kedvid FXML elemek láthatóságát hamisra, a tbllista1 láthatóságát igazra állítja és meghívja a sajátbeolvas metódust. A szerkeszthetoseg metódust meghívja és paraméterének igazat küld. Egyébként, ha a figyelol osztályváltozó értéke nagyobb mint -1, akkor a receptid, lblfhnen és a kedvid változókat feltölti. Ha a kedvid osztályváltozó igaz és a DB osztály kmevszamolvane metódusa hamisat ad vissza, a kedvid FXML elem értékét hamisra, ha a kedvid osztályváltozó hamis és a DB osztály kmevszamolvane metódusa igazat ad vissza, a kedvid FXML elem értékét igazra állítja. A tbllista, tbllista1, tbllista2 FXML elemek láthatóságát hamisra, a kimasolgombid, scrollpanelrecept, kedvid láthatóságát igazra, a sajátreceptekgombid feliratát ("Saját receptek") -re, a koppint osztályváltozót igaz-ra állítja.

@FXML public void kedvencekgombid(): Ha a leüt osztályváltozó igaz, akkor ha a figyelol2 metódus értéke nagyobb mint -1 a receptid, lblfhnen és a kedvid változókat feltölti. Ha a kedvid osztályváltozó igaz és a DB osztály kmevszamolvane metódusa hamisat ad vissza, a DB osztály kltrehoz metódusával feltölti az adatbázist. Ha a kedvid osztályváltozó hamis és a DB osztály kmevszamolvane metódusa igazat ad vissza, a DB osztály ktorol metódusával töröl az adatbázisban. A sajátreceptekgombid feliratát ("Saját receptek") -re, a kedvencekgombid feliratát ("Kedvencek") -re, a receptekgombid feliratát ("Megnyitás") -ra, a megnyom és koppint osztályváltozót igazra, a leüt osztályváltozót hamisra, a scrollpanelrecept, tbllista1, kimasolgombid, tbllista, kedvid FXML elemek láthatóságát hamisra, a tbllista2 láthatóságát igazra állítja és meghívja a kedvencekbeolvas metódust. A szerkeszthetoseg metódust

meghívja és paraméterének igazat küld. Egyébként, ha a `figyelo2` osztályváltozó értéke nagyobb mint -1, akkor a `receptid`, `lblfhnen` és a `kedvid` változókat feltölti. Ha a `kedvid` osztályváltozó igaz és a `DB` osztály `kmegszamolwane` metódusa hamisat ad vissza, a `kedvid` FXML elem értékét hamisra, ha a `kedvid` osztályváltozó hamis és a `DB` osztály `kmegszamolwane` metódusa igazat ad vissza, a `kedvid` FXML elem értékét igazra állítja. A `tbllista`, `tbllista1`, `tbllista2` FXML elemek láthatóságát hamisra, a `kimasolgombid` és `scrollpane1recept`, `kedvid` láthatóságát igazra, a `kedvencekgombid` feliratát ("Kedvencek") -re, a `koppint` osztályváltozót igazra állítja.

`@FXML public void sugogomb()`: A „Súgó” gombra kattintva a program használatát bemutató oldalt tölti be egy böngészőben.

`@FXML public void kilepesgomb()`: A `bellsoancorpaneegy`, `lbevagyrhbox` FXML elemek láthatóságát igazra, a `regisztracioancorpane`, `belepesancorpane`, `bellsoancorpaneketto`, `xgombid` FXML elemek láthatóságát hamisra állítja. A `txtkeres` FXML elem értékét törli.

`private void felirat(boolean igaze)`: A `receptnevfelirat`, `kategoriafelirat`, `hozzavalokfelirat`, `receptszovegfelirat` FXML elemek láthatóságát a paraméterként kapott `igaze(boolean)` szerint állítja.

`@FXML public void ujreceptgomb()`: Ha a `hozzaad` osztályváltozó hamis, akkor az `ujreceptnev`, `ahozzavalokszovegearea`, `areceptszovegearea`, `ujkategoria` FXML elemek szövegét törli, a `tbllista` kijelölését leveszi és az `ujreceptnev` mezőre állítja. Az `ujreceptgombid` feliratát ("Hozzáad") -ra, a `hozzaad` osztályváltozót igazra, a `sajatreceptekgombid`, `receptekgombid`, `kedvencekgombid`, `receptmodositasagombid`, `recepttorlesegombid`, `exporid`, `profilgombid`, `kilepesgombid` inaktív (Disable) értékét igazra, a `tbllista`, `tbllista1`, `tbllista2` láthatóságát hamisra, a `scrollpane1recept`, `kedvid` és `visszagombid` láthatóságát igazra állítja. Meghívja a felirat metódust és igazat ad át paraméterként. Egyébként `receptid`, `ujreceptnev`, `ahozzavalokszovegearea`, `areceptszovegearea`, `lblfhnev`, `ujkategoria`, `kedvid` változóiban eltárolja a megfelelő FXML elemek értékeit. Ha a szövegbeviteli mezők valamelyike üres "", ha a `DB` osztály `nevvizsgalat` metódusa melybe az `ujreceptnev` változód küldi paraméterként igazat ad vissza, a `Panel` osztály `hiba` metódusa üzen. Meghívja a `DB` `hozzaad` metódusát paraméterként küldve a változókat. Ha a `kedvid` osztályváltozó igaz és a `DB` osztály `kmegszamolwane` metódusa hamisat ad vissza, a `DB` osztály `kletrehoz` metódusával feltölti az adatbázist. Ha a `kedvid` osztályváltozó hamis és a `DB` osztály `kmegszamolwane` metódusa igazat ad vissza, a `DB` osztály `kto`rol metódusával töröl az adatbázisban. A `hozzaad` osztályváltozót hamisra, a `sajatreceptekgombid`, `receptekgombid`, `kedvencekgombid`, `receptmodositasagombid`, `recepttorlesegombid`, `exporid`, `profilgombid`, `kilepesgombid` inaktív (Disable) értékét hamisra, a `tbllista1`, `tbllista2`, `scrollpane1recept`, `kedvid` és `visszagombid` láthatóságát hamisra, a `tbllista` láthatóságát igazra állítja. Meghívja a `beolvas` metódust.

`private void modosit()`: Egy `nev(String)` változóban eltárolja a `tbllista` figyelő osztálymetódusának értéke szerinti alkotót. Egy `felhasznalo(String)` változóban a felhasználó nevét az `lblfhnev` FXML elem szövegét tárolja. Ha a `figyelo` metódus értéke egyenlő -1-gyel, a program kilép a metódusból. Ha a két változó szövege egyenlő és ha a `figyelo` metódus értéke nagyobb mint -1, akkor meghívja a felirat metódust és paraméterként hamisat ad át. Egyébként `receptid`, `ujreceptnev`, `ahozzavalokszovegearea`, `areceptszovegearea`, `lblfhnev`, `ujkategoria`, `kedvid` változóiban eltárolja a megfelelő FXML elemek értékeit. Ha a `modosit` osztályváltozó igaz, akkor `receptekgombid`, `sajatreceptekgombid`, `kedvencekgombid`, `exporid` inaktív (Disable) értékét igazra, a `tbllista`, `tbllista1`, `tbllista2` láthatóságát hamisra, a `scrollpane1recept` láthatóságát igazra, a `modosit` osztályváltozót hamisra, a `receptmodositasagombid` feliratát ("Szerkesztés") -re állítja. Egyébként, ha a szövegbeviteli mezők valamelyike üres "", a `Panel` osztály `hiba`

metódusa üzen. Meghívja egy sor(int) változóba a DB modosit metódusát paraméterként küldve a változókat. Ha a kedvid osztályváltozó igaz és a DB osztály kmetrhez metódusával feltölti az adatbázist. Ha a kedvid osztályváltozó hamis és a DB osztály kmetrhez metódusa igazat ad vissza, a DB osztály ktorol metódusával töröl az adatbázisban. Meghívja a beolvas, sajátbeolvas, kedvencekbeolvas metódust. Ha a sor változó nagyobb mint 0, akkor meghívja a receptekgomb metódust és a receptmodositasagombid feliratát ("Módosít") -ra változtatja. Egy for ciklussal végighalad a tbllista elemein, és amelyik elem indexe megegyezik a receptid változóval, akkor a program kijelöli a listában. A modosit osztályváltozót igazra, a sajátreceptekgombid, receptekgombid, kedvencekgombid, scrollpanelrecept, exporid, profilgombid, kilepesgombid inaktív (Disable) értékét hamisra, a tbllista láthatóságát igazra, a tbllista1, tbllista2, kedvid láthatóságát hamisra állítja. Ha a felhasználó nem módosíthatja a receptet, akkor a Panel osztály hiba metódusa üzen a felhasználónak.

private void modosit1(): Egy nev1(String) változóban eltárolja a tbllista1 figyelő osztálymetódusának értéke szerinti alkotót. Egy felhasznalo(String) változóban a felhasználó nevét az lblfhnev FXML elem szövegét tárolja. Ha a figyelo1 metódus értéke egyenlő -1-gyel, a program kilép a metódusból. Ha a két változó szövege egyenlő és ha a figyelo1 metódus értéke nagyobb mint -1, akkor meghívja a felirat metódust és paraméterként hamis-at ad át. Egyébként receptid, ujreceptnev, ahozzavalokszovegearea, areceptszovegearea, lblfhnev, ujkategoria, kedvid változóban eltárolja a megfelelő FXML elemek értékeit. Ha a módosít osztályváltozó igaz, akkor receptekgombid, sajátreceptekgombid, kedvencekgombid, exporid, kilepesgombid, profilgombid inaktív (Disable) értékét igazra, a tbllista, tbllista1, tbllista2 láthatóságát hamisra, a scrollpanelrecept és a kedvid láthatóságát igazra, a modosit osztályváltozót hamisra, a receptmodositasagombid feliratát ("Szerkesztés") -re állítja. Egyébként, ha a szövegbeviteli mezők valamelyike üres "", a Panel osztály hiba metódusa üzen. Meghívja egy sor(int) változóba a DB modosit metódusát paraméterként küldve a változókat. Ha a kedvid osztályváltozó igaz és a DB osztály kmetrhez metódusával feltölti az adatbázist. Ha a kedvid osztályváltozó hamis és a DB osztály kmetrhez metódusa igazat ad vissza, a DB osztály ktorol metódusával töröl az adatbázisban. Meghívja a beolvas, sajátbeolvas, kedvencekbeolvas metódust. Ha a sor változó nagyobb mint 0, akkor meghívja a receptekgomb metódust és a receptmodositasagombid feliratát ("Módosít") -ra változtatja. Egy for ciklussal végighalad a tbllista1 elemein, és amelyik elem indexe megegyezik a receptid változóval, akkor a program kijelöli a listában. A modosit osztályváltozót igazra, a sajátreceptekgombid, receptekgombid, kedvencekgombid, scrollpanelrecept, exporid, kilepesgombid, profilgombid inaktív (Disable) értékét hamisra, a tbllista1 láthatóságát igazra, a kedvid láthatóságát hamisra állítja. Ha a felhasználó nem módosíthatja a receptet, akkor a Panel osztály hiba metódusa üzen a felhasználónak.

private void modosit2(): Egy nev2(String) változóban eltárolja a tbllista2 figyelő osztálymetódusának értéke szerinti alkotót. Egy felhasznalo(String) változóban a felhasználó nevét az lblfhnev FXML elem szövegét tárolja. Ha a figyelo2 metódus értéke egyenlő -1-gyel, a program kilép a metódusból. Ha a két változó szövege egyenlő és ha a figyelo2 metódus értéke nagyobb mint -1, akkor meghívja a felirat metódust és paraméterként hamisat ad át. Egyébként receptid, ujreceptnev, ahozzavalokszovegearea, areceptszovegearea, lblfhnev, ujkategoria, kedvid változóban eltárolja a megfelelő FXML elemek értékeit. Ha a módosít osztályváltozó igaz, akkor receptekgombid, sajátreceptekgombid, kedvencekgombid, exporid, kilepesid, profilgombid inaktív (Disable) értékét igazra, a tbllista, tbllista1, tbllista2 láthatóságát hamisra, a scrollpanelrecept és a kedvid láthatóságát igazra, a modosit osztályváltozót hamisra, a receptmodositasagombid feliratát ("Szerkesztés") -re állítja. Egyébként, ha a szövegbeviteli mezők

valamelyike üres "", a Panel osztály hiba metódusa üzen. Meghívja egy sor(int) változóba a DB modosit metódusát paraméterként küldve a változókat. Ha a kedvid osztályváltozó igaz és a DB osztály kmevszamolvane metódusa hamisat ad vissza, a DB osztály kltrehoz metódusával feltölti az adatbázist. Ha a kedvid osztályváltozó hamis és a DB osztály kmevszamolvane metódusa igazat ad vissza, a DB osztály ktorol metódusával töröl az adatbázisban. Meghívja a beolvas, sajátbeolvas, kedvencekbeolvas metódust. Ha a sor változó nagyobb mint 0, akkor meghívja a receptekgomb metódust és a receptmodositasagombid feliratát ("Módosít") -ra változtatja. Egy for ciklussal végighalad a tbllista2 elemein, és amelyik elem indexe megegyezik a receptid változóval, akkor a program kijelöli a listában. A modosit osztályváltozót igazra, a sajátreceptekgombid, receptekgombid, kedvencekgombid, scrollpane1recept, exportid, kilepesgombid, profilgombid natív (Disable) értékét hamisra, a tbllista2 láthatóságát igazra, a kedvid láthatóságát hamisra állítja. Ha a felhasználó nem módosíthatja a receptet, akkor a Panel osztály hiba metódusa üzen a felhasználónak.

@FXML public void receptmodositasagomb(): A gombra kattintáskor ha a megnyom osztályváltozó hamis, akkor meghívja az osztály modosit metódusát. Ha a koppint osztályváltozó hamis, akkor meghívja az osztály modosit1 metódusát. Ha a leüt osztályváltozó hamis, akkor meghívja az osztály modosit2 metódusát.

private boolean recepttorlese(): Egy nev1(String) változóban eltárolja a tbllista figyelő osztálymetódusának értéke szerinti alkotót. Egy felhasznalo(String) változóban a felhasználó nevét az lblfhnev FXML elem szövegét tárolja. Ha a figyeló metódus értéke egyenlő -1-gyel, a metódusnak hamis értéket ad. Ha a két változó szövege egyenlő, ha a Panel osztály iggennem metódusa hamis értéket ad vissza, a metódus értékét hamisra állítja. Egy id(int) változóban tárolja a tbllista figyeló osztálymetódusának értéke szerinti receptid, egy sor(int) változóba meghívja a DB osztály torol metódusát és paraméterként küldi az id és nev1 változót. Meghívja a DB osztály ktorlo metódusát és az id változót küldi paraméterként. Ha a sor változó értéke nagyobb mint 0, akkor a metódus értékének igazat állít be és meghívja a beolvas metódust. Egyébként a Panel osztály hiba metódusával üzenetet küld a megfelelő szöveggel.

private boolean recepttorlese1(): Egy nev2(String) változóban eltárolja a tbllista1 figyeló1 metódusának értéke szerinti alkotót. Egy felhasznalo(String) változóban a felhasználó nevét az lblfhnev FXML elem szövegét tárolja. Ha a figyeló1 metódus értéke egyenlő -1-gyel, a metódusnak hamis értéket ad. Ha a két változó szövege egyenlő, ha a Panel osztály iggennem metódusa hamis értéket ad vissza, a metódus értékét hamisra állítja. Egy id(int) változóban tárolja a tbllista1 figyeló1 osztálymetódusának értéke szerinti receptid, egy sor(int) változóba meghívja a DB osztály torol metódusát és paraméterként küldi az id és nev2 változót. Meghívja a DB osztály ktorlo metódusát és az id változót küldi paraméterként. Ha a sor változó értéke nagyobb mint 0, akkor a metódus értékének igazat állít be és meghívja a sajarbeolvas metódust. Egyébként a Panel osztály hiba metódusával üzenetet küld a megfelelő szöveggel.

private boolean recepttorlese2(): Egy nev3(String) változóban eltárolja a tbllista2 figyeló2 metódusának értéke szerinti alkotót. Egy felhasznalo(String) változóban a felhasználó nevét az lblfhnev FXML elem szövegét tárolja. Ha a figyeló metódus értéke egyenlő -1-gyel, a metódusnak hamis értéket ad. Ha a két változó szövege egyenlő, ha a Panel osztály iggennem metódusa hamis értéket ad vissza, a metódus értékét hamis-ra állítja. Egy id(int) változóban tárolja a tbllista2 figyeló2 osztálymetódusának értéke szerinti receptid-t, egy sor(int) változóba meghívja a DB osztály torol metódusát és paraméterként küldi az id és nev3 változót. Meghívja a DB osztály ktorlo metódusát és az id változót küldi paraméterként. Ha a sor változó értéke nagyobb mint 0, akkor a metódus értékének igazat állít be és meghívja a

kedvencekbeolvas metódust. Egyébként a Panel osztály hiba metódusával üzenetet küld a megfelelő szöveggel.

@FXML public void recepttorlesegomb(): A gombra kattintáskor ha a megnyom osztályváltozó hamis, akkor meghívja az osztály recepttorlese metódusát. Ha a koppint osztályváltozó hamis, akkor meghívja az osztály recepttorlese1 metódusát. Ha a leüt osztályváltozó hamis, akkor meghívja az osztály recepttorlese2 metódusát.

@FXML public void kimasolgomb(): Egy cb(Clipboard) változóba bekéri a Clipboard stílusát. Létrehoz egy tartalom (ClipboardContent) változót és ebbe feltölti az ujreceptnev, ahozzavalokszovegearea, areceptszovegearea FXML elemek szövegét. Majd a cb változó értékét a tartalom értékére állítja.

private void beolvas(): Meghívja az AB osztály beolvas metódusát és paraméterként küldi a tbllista elemeit.

private void sajátbeolvas(): Meghívja az AB osztály sajátok metódusát és paraméterként küldi a tbllista1 elemeit és a felhasználó nevét az lblfhnev FXML elemből.

private void kedvencekbeolvas(): Meghívja az AB osztály kedvencekbeolvas metódusát és paraméterként küldi a tbllista2 elemeit és a felhasználó nevét az lblfhnev FXML elemből.

private void keresem(): Létrehoz egy keresett(String) változót, lekéri a változóba a txtkeres FXML elem értékét és létrehoz egy sql(String) változót. Ha a megnyom osztályváltozó hamis, akkor az sql változó szövegét "select \* from recept where nev like '%" + keresett + "%' order by receptid;" -re állítja. Meghívja a DB osztály keresem metódusát és paraméterként küldi a tbllista FXML elem és az sql változók értékét. Ha a koppint osztályváltozó hamis, akkor az sql változó szövegét "SELECT \* from recept WHERE alkoto='" + lblfhnev.getText() + "' AND nev LIKE '%" + keresett + "%' ;"-re állítja. Meghívja a DB osztály keresem metódusát és paraméterként küldi a tbllista1 FXML elem és az sql változók értékét. Ha a leüt osztályváltozó hamis, akkor az sql változó szövegét "SELECT \* from kedvencek JOIN recept on recept.receptid=kedvencek.receptid WHERE recept.alkoto='" + lblfhnev.getText() + "' and kedvencek.kedvenc=1 and recept.nev LIKE '%" + keresett + "%' ;"-re állítja. Meghívja a DB osztály keresem metódusát és paraméterként küldi a tbllista1 FXML elem és az sql változók értékét.

private void tablabol(int i): Paraméterként vár egy i(int) változót. Ha az i változó egyenlő -1-el a program kilép a metódusból. Ha az osztály figyelő metódusának értéke nagyobb mint -1, akkor egy e(Recept) változóba tölti a tbllista elemének i elemeit. És beállítja az ujreceptnev., ujkategori, ahozzavalokszovegearea, areceptszovegearea, kedvid FXML elemek értékét. Ha az osztály figyelő2 metódusának értéke nagyobb mint -1, akkor egy e1(Recept) változóba tölti a tbllista1 elemének i elemeit. És beállítja az ujreceptnev., ujkategori, ahozzavalokszovegearea, areceptszovegearea, kedvid FXML elemek értékét. Ha az osztály figyelő2 metódusának értéke nagyobb mint -1, akkor egy e2(Kedvenc) változóba tölti a tbllista2 elemének i elemeit. És beállítja az ujreceptnev., ujkategori, ahozzavalokszovegearea, areceptszovegearea, kedvid FXML elemek értékét.

@FXML public void felhasznalonevmodosit(): Ha a felhkapcsolat1 FXML elem bekért szövegének hossza 5 karakternél kisebb, akkor a Panel osztály hiba metódusával üzenetet küld a megfelelő szöveggel. Egyébként meghívja a DB osztály receptbenalkotoatiras metódusát és paraméterként küldi a felhasználó új nevét és a régi nevét. Meghívja a DB osztály f1 metódusát és paraméterként küldi a felhkapcsolat1 és az lblfhnev FXLM értékét. Beállítja az lblfhnev

FXML elem értékét a felhkapcsolat1 FXML elem szövegére. Egy tomb (String) tömbben meghívja a DB osztály feltolt metódusát és paraméterként küldi a felhasználó nevét. A felhkapcsolat és az emailkapcsolat FXML elemek értékét a tomb változó megfelelő értékére állítja. A Panel osztály informacio metódusával üzenetet küld a küld a megfelelő szöveggel.

@FXML public void jelszomodosit(): Ha a jelszokapcsolat1 FXML elem bekért szövegének hossza 5 karakternél kisebb, akkor a Panel osztály hiba metódusával üzenetet küld a megfelelő szöveggel. Egyébként meghívja a DB osztály f2 metódusát és paraméterként küldi a jelszokapcsolat1 és az lblfhnev FXLM értékét. Majd a Panel osztály informacio metódusával üzenetet küld a küld a megfelelő szöveggel.

@FXML public void emailmodosit(): Meghívja a DB osztály f3 metódusát, paraméterének az emailkapcsolat1 és az lblfhnev szövegét. Egy tomb(String) tömbben meghívja a DB osztály feltolt metódusát és paraméterként küldi a felhasználó nevét. A felhkapcsolat és az emailkapcsolat FXML elemek értékét a tomb változó megfelelő értékére állítja. Majd a Panel osztály informacio metódusával üzenetet küld a küld a megfelelő szöveggel.

private String txt1(int index): Paraméterként kér egy index értéket. Egy kikuldes(String) változóba a tbllista indexedik elemének értékeit tárolja. És a metódusnak a kikuldes értékét állítja be.

private String txt2(int index): Paraméterként kér egy index értéket. Egy kikuldes(String) változóba a tbllista1 indexedik elemének értékeit tárolja. És a metódusnak a kikuldes értékét állítja be.

private String txt3(int index): Paraméterként kér egy index értéket. Egy kikuldes(String) változóba a tbllista2 indexedik elemének értékeit tárolja. És a metódusnak a kikuldes értékét állítja be.

@FXML public void export(): Egy fc(FileChooser) változónak fejlécének szövegét ("Exportálás")-ra állítja. Létrehoz három változóban új txt, doc file-t és hozzáadja az fc változóhoz. Létrehoz egy d(File) változót, majd dialógus ablakot tölt be melynek értéke null. Ha a d változó értéke nem null, akkor egy ir(PrintWriter) változóban „UTF8” szöveg típussal létrehozza a dokumentum szövegét ami ha az osztály figyeló metódusa nagyobb -1- nél. akkor az ir változóba meg hívja a txt1 metódus értékét és paraméterként a figyeló metódus értékét küldi. Ha az osztály figyeló1 metódusa nagyobb -1- nél. akkor az ir változóban meg hívja a txt2 metódus értékét és paraméterként a figyeló1 metódus értékét küldi. Ha az osztály figyeló2 metódusa nagyobb -1- nél. akkor az ir változóba meg hívja a txt3 metódus értékét és paraméterként a figyeló2 metódus értékét küldi. Egyébként hibaüzenetet küld a Panel osztály hiba metódusával és a megfelelő szöveggel.

@Override public void initialize(URL url, ResourceBundle rb): A tbllista, tbllista1.tbllista2 FXML elemek oszlopainak értékét állítja be, meghívja a beolvas, sajátbeolvas, kedvencekbeolvas metódusokat, figyel a tbllista, tbllista1.tbllista2 FXML elemeket és meghívja a tablabol metódust. És a txtkeres FXML elem értékét is figyel a meghívja rá a keresem metódust.



Importálások:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javafx.collections.ObservableList;
```

Osztályváltozók:

```
final String db = "jdbc:mysql://localhost:3306/receptkonyv"+ "?useUnicode=true&characterEncoding=UTF-8";
final String user = "chef";
final String pass = "chef";
```

---

Metódusok:

public void beolvas(ObservableList<Recept> tabla):c Új kapcsolatot hoz létre az osztályváltozók és az s(String) változóban eltárolt "select \* from recept order by receptid; " SQL paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és lekérdezi. Létrehoz egy tabla(Recept) ObservableList adatszerkezetet, kitisztítja, majd eltárolja benne a lekérdezés eredményét. Ha a kapcsolat nem sikerült a Panel osztály hiba metódusa üzen. Ezzel feltölti a receptek táblát a receptekkel.

public void keresem(ObservableList<Recept> tabla, String sql): Új kapcsolatot hoz létre az osztályváltozók és a paraméterként kapott sql(String) SQL paranccsal. A program futtassa és lekérdezi. Létrehoz egy tabla(Recept) ObservableList adatszerkezetet, kitisztítja, majd eltárolja benne a lekérdezés eredményét. Ha a kapcsolat nem sikerült a Panel osztály hiba metódusa üzen. Ezzel feltölti a receptek és a saját receptek táblát a receptekkel.

public void kedvencekkeresem(ObservableList<Kedvenc> tabla, String sql): Új kapcsolatot hoz létre az osztályváltozók és a paraméterként kapott sql(String) SQL paranccsal. A program futtassa és lekérdezi. Létrehoz egy tabla(Kedvenc) ObservableList adatszerkezetet, kitisztítja, majd eltárolja benne a lekérdezés eredményét. Ha a kapcsolat nem sikerült a Panel osztály hiba metódusa üzen. Ezzel feltölti kedvencek táblát a receptekkel.

public int hozzaad(String nev, String categoria, String leiras1, String leiras2, String alkoto, boolean kedvenc): Új kapcsolatot hoz létre az osztályváltozók és az s(String) változóban eltárolt "insert into recept (nev, categoria,leiras1,leiras2, alkoto, kedvenc) values (?, ?, ?, ?, ?, ?); " SQL paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és létrehozza. Ha a kapcsolat nem sikerült a Panel osztály hiba metódusa üzen és a metódus értékének 0-at ad vissza. Így létrehozza az új receptet.

public int receptidlekerdezo(): Új kapcsolatot hoz létre az osztályváltozók és az s(String) változóban eltárolt "select max(receptid) as maxid from recept ; " SQL paranccsal. Egy

maxid(int) változóban eltárolja a lekérdezés eredményét. A program futtassa és lekérdezi és a módszernek átadja a maxid értékét. Ha a kapcsolat nem sikerült a Panel osztály hiba módszere üzen és -1-et ad vissza. Ezzel megadja a legutolsó recept azonosítóját.

public boolean nevvizsgalat(String nev): Új kapcsolatot hoz létre az osztályváltozókkal és az s(String) változóban eltárolt "SELECT COUNT(nev) as Darab FROM recept WHERE nev=?;" SQL paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és lekérdezi. Egy vane(int) változóban eltárolja a lekérdezés eredményét. Visszaadja a vane változó nagyobb-e mint 0 választ (igaz, hamis) Ha a kapcsolat nem sikerült a Panel osztály hiba módszere üzen és a módszer értékének igazat ad vissza. Így megszámolja, hogy hány darab olyan név van, amit ad a felhasználó.

public int modosit(String nev, String kategoria, String leiras1, String leiras2, boolean kedvenc, int receptid): Új kapcsolatot hoz létre az osztályváltozókkal és az s(String) változóban eltárolt "update recept set nev=?, kategoria=?, leiras1=?, leiras2=?, kedvenc=? where receptid=?;" SQL paranccsal. A program futtassa és módosítja. Ha a kapcsolat nem sikerült a Panel osztály hiba módszere üzen. A módszernek 0 értéket ad. Így módosítja a recepteket

public int torol(int receptid, String alkoto): Új kapcsolatot hoz létre az osztályváltozókkal és az s(String) változóban eltárolt "delete from recept where receptid=? and alkoto=?;" SQL paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és törli. Ha a kapcsolat nem sikerült a Panel osztály hiba módszere üzen és a módszer értékének 0-at ad vissza. Így törli a receptet.

public int felhasznalohozzaad(String fhNEV, String fhEmail, String fhJelszo): Új kapcsolatot hoz létre az osztályváltozókkal és az s(String) változóban eltárolt "insert into felhasznalok (fhNEV,fhJelszo,fhEmail) values (?,?);" SQL paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és létrehozza. Ha a kapcsolat nem sikerült a Panel osztály hiba módszere üzen és a módszer értékének 0-at ad vissza. Így létrehozza az új felhasználót.

public boolean felhasznalonevvizsgalat(String fhNEV): Új kapcsolatot hoz létre az osztályváltozókkal és az s(String) változóban eltárolt "SELECT COUNT(fhNEV) as Darab FROM felhasznalok WHERE fhNEV=?;" SQL paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és lekérdezi. Egy vane(int) változóban eltárolja a lekérdezés eredményét. Visszaadja a vane változó nagyobb-e mint 0 választ (igaz, hamis) Ha a kapcsolat nem sikerült a Panel osztály hiba módszere üzen és a módszer értékének igazat ad vissza. Így megszámolja, hogy hány darab olyan név van, amit ad a felhasználó.

public String jelszovizsgalat(String fhNEV): Új kapcsolatot hoz létre az osztályváltozókkal és az s(String) változóban eltárolt "SELECT fhJelszo from felhasznalok where fhNEV=?;" SQL paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és lekérdezi. Egy vane(int) változóban eltárolja a lekérdezés eredményét. Visszaadja a vane változó nagyobb-e mint 0 választ (igaz, hamis) Ha a kapcsolat nem sikerült a Panel osztály hiba módszere üzen és a módszer értékének ""-at ad vissza. Így le ellenőrzi, hogy jó-e a jelszó, amit belépéskor ad a felhasználó.

public void sajátok(ObservableList<Recept> tabla, String alkoto): Új kapcsolatot hoz létre az osztályváltozókkal és az s(String) változóban eltárolt "select \* from recept where alkoto=?;" SQL paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és lekérdezi. Létrehoz egy tabla(Recept) ObservableList adatszerkezetet, kitisztítja, majd eltárolja benne a lekérdezés eredményét. Ha a kapcsolat nem sikerült a Panel osztály hiba módszere üzen. Ezzel feltölti a saját receptek táblát a receptekkel.

public void kedvencekbeolvas(ObservableList<Kedvenc> tabla, String alkoto): Új kapcsolatot hoz létre az osztályváltozókkal és az s(String) változóban eltárolt "select recept.nev, recept.receptid, recept.kategoria, recept.leiras1, recept.leiras2, recept.alkoto, kedvencek.receptid, kedvencek.kedvenc, kedvencek.fhNev "+"from kedvencek join recept on recept.receptid=kedvencek.receptid where kedvencek.kedvenc=true AND kedvencek.fhNev=? ; " SQL paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és lekérdezi. Létrehoz egy tabla(Kedvenc) ObservableList adatszerkezetet, kitisztítja, majd eltárolja benne a lekérdezés eredményét. Ha a kapcsolat nem sikerült a Panel osztály hiba metódusa üzen. Ezzel feltölti a kedvencek táblát a receptekkel.

public boolean kmegszamolwane(int receptid, String fhNEV): Új kapcsolatot hoz létre az osztályváltozókkal és az s(String) változóban eltárolt "DELETE FROM kedvencek where receptid=?; "SQL paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és lekérdezi. Egy vane(int)változóban eltárolja a lekérdezés eredményét. Visszaadja a vane változó nagyobb-e mint0 választ (igaz, hamis) Ha a kapcsolat nem sikerült a Panel osztály hiba metódusa üzen és a metódus értékének igazat ad vissza. Így megszámlolja, hogy hány darab sor van ezzel a felhasználónévvel és receptnévvel.

public int ktorlo(int receptid): Új kapcsolatot hoz létre az osztályváltozókkal és az s(String) változóban eltárolt "DELETE FROM kedvencek where receptid=?; "SQL paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és törli. Ha a kapcsolat nem sikerült a Panel osztály hiba metódusa üzen és a metódus értékének 0-at ad vissza. Így törli a felhasználó kedvenc recept sorát.

public int kletrehozoz(int receptid, boolean kedvenc, String fhNEV): Új kapcsolatot hoz létre az osztályváltozókkal és az s(String) változóban eltárolt "insert into kedvencek (receptid,kedvenc,fhNev) values (?,?,?); " SQL paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és létrehozza. Ha a kapcsolat nem sikerült a Panel osztály hiba metódusa üzen és a metódus értékének 0-at ad vissza. Így létrehozza a felhasználó kedvenc recept lehetőségét.

public int fioktorles(String fhNEV): Új kapcsolatot hoz létre az osztályváltozókkal és az s(String) változóban eltárolt SQL "delete from felhasznalok where fhNEV=?; " paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és törli. Ha a kapcsolat nem sikerült a Panel osztály hiba metódusa üzen. Így törli a felhasználó fiókját.

public void f1(String fhNEVm, String fhNEV): Új kapcsolatot hoz létre az osztályváltozókkal és az s(String) változóban eltárolt SQL "UPDATE felhasznalok SET fhNEV=? WHERE fhNEV=?" paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és frissíti. Ha a kapcsolat nem sikerült a Panel osztály hiba metódusa üzen. Így módosítja a felhasználó nevét.

public void f2(String fhJelszom, String fhJelszo): Új kapcsolatot hoz létre az osztályváltozókkal és az s(String) változóban eltárolt "UPDATE felhasznalok SET fhJelszo=? WHERE fhNEV=?" SQL paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és frissíti. Ha a kapcsolat nem sikerült a Panel osztály hiba metódusa üzen. Így módosítja a felhasználó jelszavát.

public void f3(String fhemailm, String fhEmail): Új kapcsolatot hoz létre az osztályváltozókkal és az s(String) változóban eltárolt "UPDATE felhasznalok SET fhEmail=? WHERE fhNEV=?" SQL paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és frissíti. Ha a kapcsolat nem sikerült a Panel osztály hiba metódusa üzen. Így módosítja a felhasználó e-mail címét.

`public String[] feltolt(String fhNEV):` Új kapcsolatot hoz létre az osztályváltozókkal és az `s(String)` változóban eltárolt `"SELECT fhNEV,fhEmail FROM felhasznalok WHERE fhNEV=?; "` SQL paranccsal. Egy `tomb(String)` tömbben eltárolja a lekérdezés eredményét. A program futtassa és lekérdezi. Ha a kapcsolat nem sikerült a Panel osztály hiba metódusa üzen. A metódusnak átadja a `tomb` értékét. Így vissza adja a felhasználó nevét és e-mail címét.

`public String[] feltolt(String fhNEV):` Új kapcsolatot hoz létre az osztályváltozókkal és az `s(String)` változóban eltárolt `"UPDATE recept SET alkoto=? WHERE alkoto=?"` SQL paranccsal. A paraméterként kért adatokkal feltölti az SQL kódot. A program futtassa és frissíti. Ha a kapcsolat nem sikerült a Panel osztály hiba metódusa üzen. Így a receptek alkotóinak a neve megváltozik ott, ahol szükséges.

#### *Panel osztály*

Importálások:

```
import javafx.scene.control.Alert;
import javafx.scene.control.ButtonBar.ButtonData;
import javafx.scene.control.ButtonType;
import javafx.scene.layout.Background;
```

---

Metódusok:

`public static void hiba(String cim, String uzenet):` Egy új felugró(alert) ablak stílusát hibára(error) állítja. Két paraméterből vár be értéket, cím(String) és uzenet(String). Az ablak címét a cím paraméterre, a szövegtörzsének az uzenet változó értékét állítja be. Majd betölti és várakoztatja.

`public static boolean igennem(String cim, String uzenet):` Egy új felugró(alert) ablak stílusát eldöntendőre(confirmation) állítja. Két paraméterből vár be értéket, cím(String) és uzenet(String). Az ablak címét a cím paraméterre, a szövegtörzsének az uzenet változó értékét, a fejlécének pedig üres(null) értéket állítja be. Egy `btIgen(ButtonType)` gombot „Igen” felirattal és `igen(yes)` értékkel, és egy `btNem(ButtonType)` gombot „Nem” felirattal és `nem(no)` értékkel hoz létre. Majd betölti és várakoztatja.

`public static void kapcsolat(String cim, String uzenet):` Egy új felugró(alert) ablak stílusát információ(information) állítja. Két paraméterből vár be értéket, cím (String) és uzenet(String). Az ablak címét a cím paraméterre, a szövegtörzsének az uzenet változó értékét állítja be. Az ablak transzformációs értékét bekéri. Az ablak koordinátáit x (650) és y (200) -ra, preferált méretét (700, 630) -ra, háttérét empty-re és a fejlécét üresre(null) állítja. Majd betölti és várakoztatja.

`public static void informacio(String cim, String uzenet):` Egy új felugró(alert) ablak stílusát információ(information) állítja. Két paraméterből vár be értéket, cím (String) és uzenet(String). Az ablak címét a cím paraméterre, a szövegtörzsének az uzenet változó értékét állítja be. Az ablak transzformációs értékét bekéri. Az ablak koordinátáit x(700) és y(220) -ra,

preferált méretét(400, 250)-ra, háttérét empty-re és a fejlécét üresre(null) állítja. Majd betölti és várakoztatja.

#### *Felhasználó osztály*

Osztálymetódusok:

private int felhID;

private String fhNEV, fhJelszo, fhEmail;

---

#### *Konstruktorok:*

public Felhasználó(int felhID, String fhNEV, String fhJelszo, String fhEmail): A felhID, fhNEV, fhJelszo, fhEmail paramétereket teszi egyenlővé az osztályváltozókkal.

Getters and Setters: Az osztály változóit teszi szerkeszthetővé és lekérdezhetővé más osztályok számára.

#### *Kedvenc osztály*

Osztályváltozók:

private int receptid;

private int kedvencekreceptid;

private String kedvencekfhNEV;

private String nev, categoria, leiras1, leiras2, alkoto;

private boolean kedvencekkedvenc;

private boolean kedvenc;

---

Metódusok:

public Kedvenc(int receptid, String fhNEV, boolean kedvenc): A receptid, fhNEV, kedvenc osztályváltozókat teszi egyenlővé a paraméterekkel.

public Kedvenc(int recept\_receptid, String recept\_nev, String recept\_categoria, String recept\_leiras1, String recept\_leiras2, String recept\_alkoto, int kedvencekreceptid, String kedvencekfhNEV, boolean kedvencekkedvenc): A receptid, nev, categoria, leiras1, leiras2, alkoto, kedvencekreceptid, kedvencekfhNEV, kedvencekkedvenc osztályváltozókat teszi egyenlővé a paraméterekkel.

Getters and Setters: Az osztály változóit teszi szerkeszthetővé és lekérdezhetővé más osztályok számára.

### *Recept osztály*

Osztályváltozók:

```
private int receptid;  
private String nev, categoria, leiras1, leiras2, alkoto;  
private boolean kedvenc;
```

---

Metódusok:

`public Recept(int receptid, String nev, String categoria, String leiras1, String leiras2, String alkoto, boolean kedvenc)`: A `receptid`, `nev`, `categoria`, `leiras1`, `leiras2`, `alkoto`, `kedvenc` osztályváltozókat teszi egyenlővé a paraméterekkel.

Getters and Setters: Az osztály változóit teszi szerkeszthetővé és lekérdezhetővé más osztályok számára.

### *Receptkönyv.java főosztály*

Importálások:

```
import javafx.application.Application;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.image.Image;  
import javafx.stage.Stage;
```

---

Metódusok:

A program osztálya az `Application` osztály leszármazottja. A `main` metódusból elindított `launch` metódus lefuttatja a `start` metódust. Ezt írja felül az ablak megjelenítéséhez szükséges utasításokkal. A `start` metódus először betölti az `FXML` fájl tartalmát egy `Parent` típusú root elembe. Ezután készít egy új jelenetet belőle (`Scene`), a jelenetet hozzárendeli egy ablakhoz (`Stage`), módosítja az `Icons` képet, leveszi az átméretezhetőségét (`Resizable`), majd megjeleníti az ablakot a `show` metódus segítségével.

`public static void main(String[] args)`: Elindítja a `Launch` metódust.



## Tesztelés

A programot fejlesztés közben teszteltem és javítottam.

Tesztelés: A program grafikus felülete részben elkészült. A Receptek, Saját receptek, Keresés törlése gomb és a keresőmező működéséhez szükséges metódusok készültek el.

Hiba: A recepteket betölti a két táblába, de csak az első gombnyomáskor fut le a Receptek, Saját receptek gombok metódusa.

Tesztelés: A Receptek, Saját receptek Kedvencek, Keresés törlése gomb és a keresőmező, Módosít, Új recept, Törlés működéséhez szükséges metódusok készültek el.

Hiba: Bármelyik receptet, más felhasználó receptjeit is módosíthatja vagy törölheti bármelyik felhasználó.

Tesztelés: A Receptek, Saját receptek, Kedvencek, Keresés törlése gomb és a keresőmező, Módosít, Új recept, Törlés, Profil, Kilépés, Kapcsolat működéséhez szükséges metódusok készültek el.

Hiba: A program Profil ablakában nem ellenőrzi a jelszavak, a felhasználónevek és az e-mailcímek módosítását. Bármilyen adathosszúságot meg lehet adni. Nincsenek felugró ablakok hibaüzenetekkel és visszajelzési információkkal.

Tesztelés: A recepteket megfelelően menti név alkotó, hozzávalók, elkészítés és kedvencek jelölés szempontjából.

Hiba: A program lapjai (táblázatok, schrollpane, ancorpane, gombok, inputmezők) láthatóságai, szerkeszthetőségein inaktivitása nem minden metódus futása után módosulnak megfelelően.

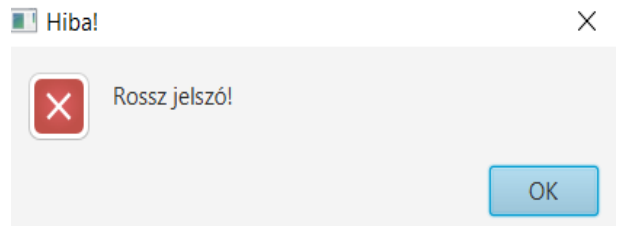
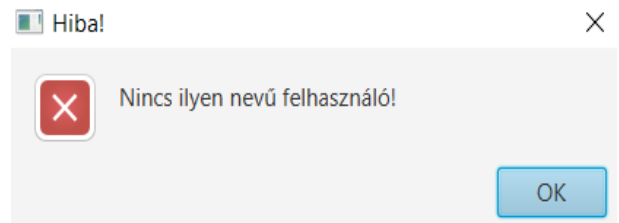
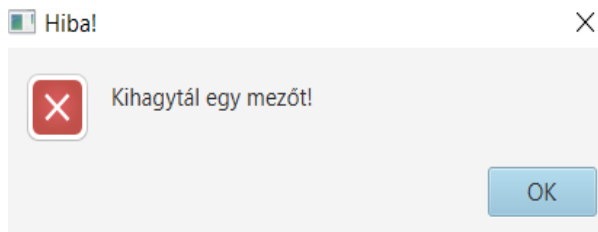
Tesztelés: A kedvenceknek jelölő gomb kattintható, a kedvencek táblát feltölti és módosítja a megfelelő helyeken.

Hiba: Minden recept megnyitáskor módosítja a kedvencek táblát akkor is, ha a felhasználó nem használta előtte a jelölőgombot.

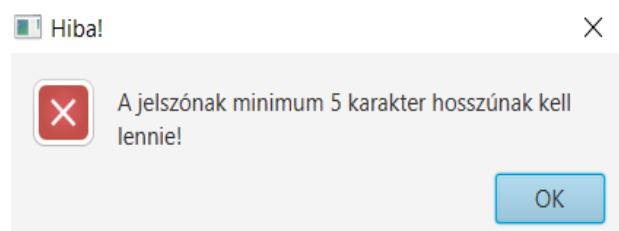
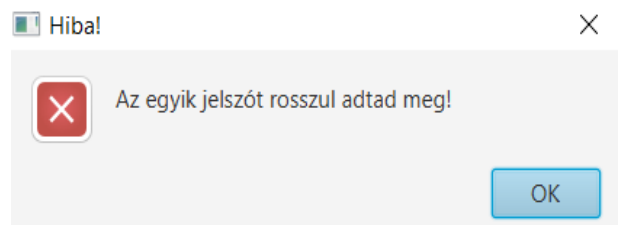
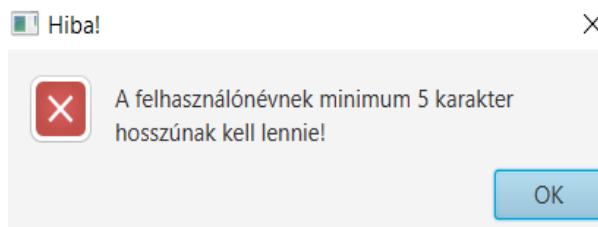
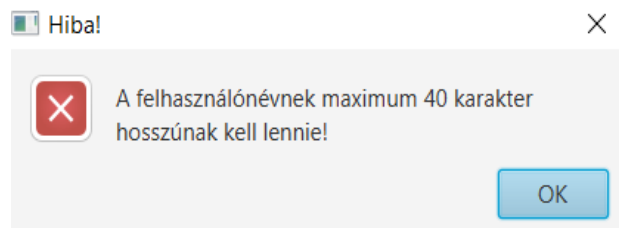
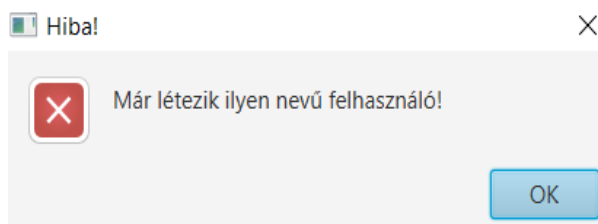
Tesztelés: A program egésze elkészült. Elvárásnak megfelelően működik. Hiba: Helyesírási és ablakok méretezési hibák.

Felugró hiba és információs ablakok a program felületei alapján tagolva:

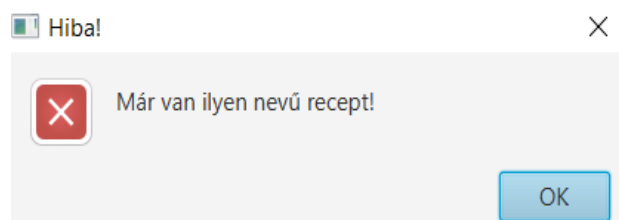
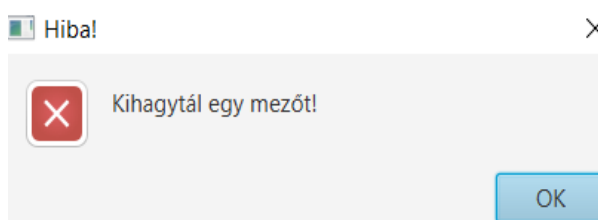
#### Belépés:



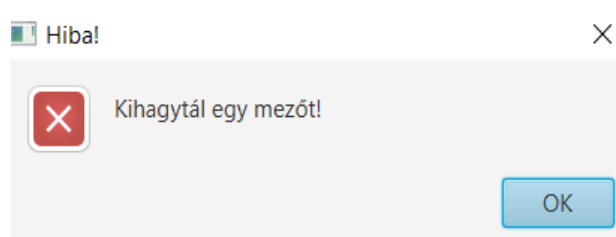
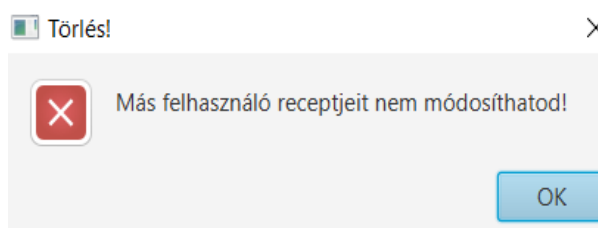
#### Regisztrálás:



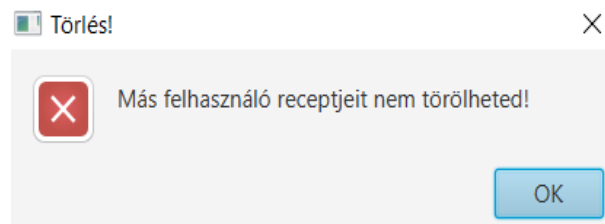
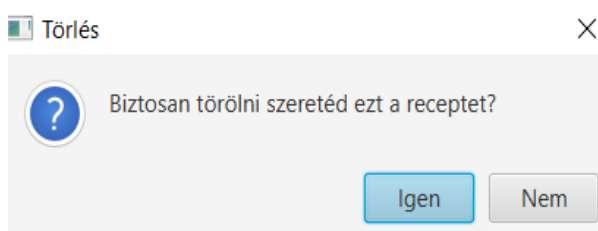
#### Új recept:



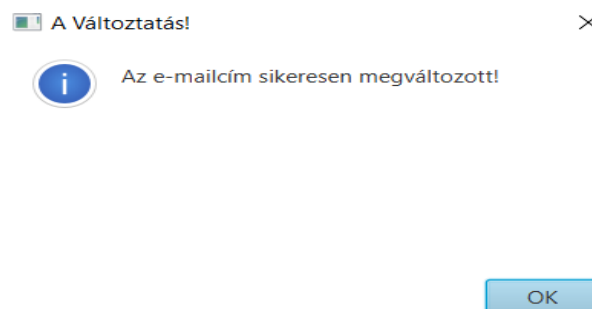
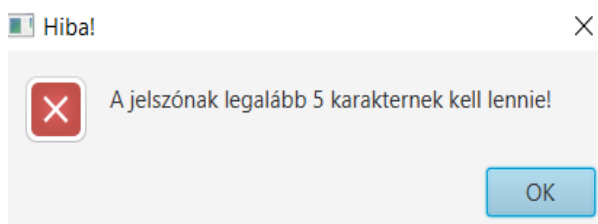
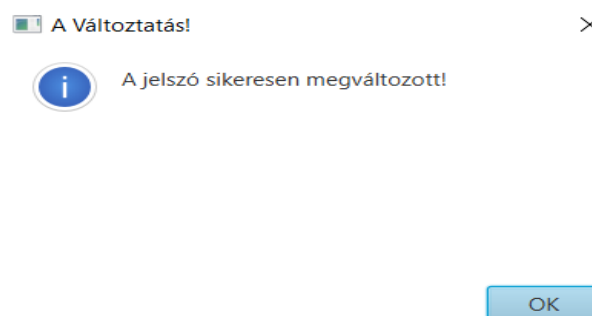
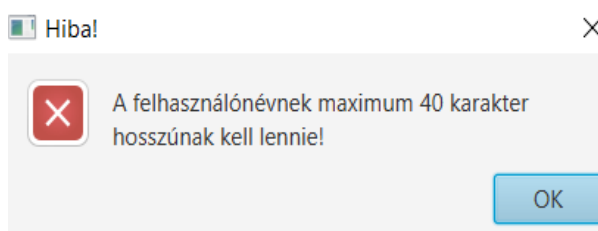
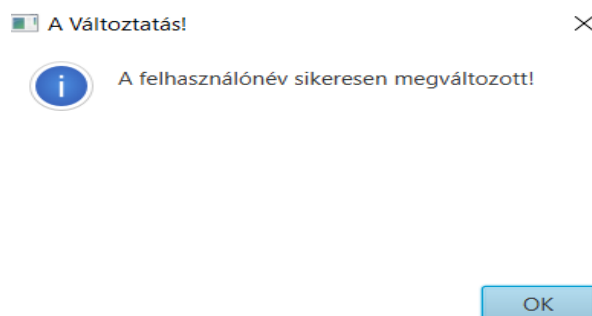
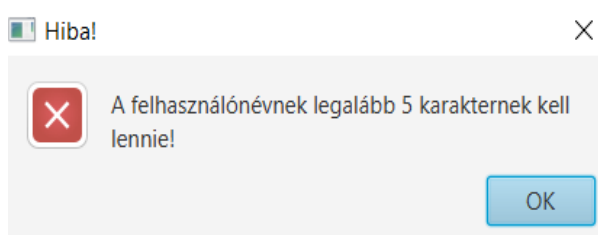
## Módosítás:



## Törlés



## Profil



## Továbbfejlesztési lehetőségek

A program célja egy könnyen használható programmal receptek egyszerű és gyors kezelése, ezáltal lehetne a recepteket még egy Csoportok táblázatban kategóriánként, vagy személyesebb elnevezésű csoportosítással megjeleníteni a recepteket, az összes felhasználó készíthetne saját magának. De egymás között meg is oszthatnák ezeket a csoportokat.

A recepteket még lehetne a felhasználó által bekért képekkel is illusztrálni. Ezeket a program ablakában megadott névvel, vagy rendszerablakkal beimportálni. Ezt pedig adatbázisban elmentve a recepthez kötni. Ezeket lehetne a dokumentumokhoz is elmenteni dokumentumok kiírásakor.

A recepteket hozzávalók alapján is lehetne keresni, így a háztartásban már meglévő alapanyagok szerint is ki lehetne keresni az éppen elkészíthető recepteket. Így nem kellene minden kiválasztott recept után boltba járni, ha nincsen elegendő alapanyag.

Elrejthető megjegyzéseket is lehetne a receptekhez adni jó ötletekkel a receptekhez, amelyhez minden felhasználó írhatna hozzászólást attól függetlenül, hogy a receptet ki hozta létre. Ezt a receptet létrehozó személy elolvasva eldöntheti, hogy módosítja a saját receptjeit vagy nem. De ettől függetlenül mindenki láthatná a jó ötleteket.

Létre lehetne hozni egy kategóriák gombot is, amely beállítaná, hogy melyik kategória alapján keresünk receptet. Vagy a táblázatok oszlopai szerint, nem csak név alapján lehetne keresni, hanem más oszlopban megjelenített szöveg által.

A receptekhez létre lehetne hozni egy dátum kezelőt, mely alapján a receptek létrehozása után a létrehozás dátumát is lehetne menteni. Ezáltal a recept létrejöttének időpontjára is lehetne keresni.

## Irodalomjegyzék, forrásmegjelölés

A program háttérét alkotó kép: A képet ingyenesen töltöttem le a <https://www.pexels.com/> weboldalról. Ezt a képet használtam fel a program háttérének.

Súgóhoz: A sugo weblap megnyitásához felhasznált kódhoz a <https://stackoverflow.com/> weboldalról származó kód segítségével készítettem el.

## A CD melléklet tartalma

A teljes forráskód

A Teleptő mappa

A Szakdolgozat dokumentáció

A Powerpoint prezentáció