|  |
| --- |
| Easyway Fitness |
| Dokumentáció-Asztali alkalmazás |
| Adminisztrációs felület |

|  |
| --- |
| Várszegi Ádám  2023.04.15. |

**Felhasználói dokumentáció**

**Környezet:**

Backend szerver indítása után a target\Vizsgaremek\_JavaFx mappában található exe fájl futtatása.

**Használat:**

A fejlesztők által kapott admin felhasználónév és jelszó helyes megadása után teljes körben használható a program.

A képen szöveg, képernyőkép, tervezés, lámpa látható

Automatikusan generált leírásHa elfelejtette a jelszót vagy felhasználónevet lépjen kapcsolatba a fejlesztőkkel

*Felhasználók listázása*

A képen képernyőkép, szöveg, szoftver, Multimédiás szoftver látható

Automatikusan generált leírás

Ebben a táblázatban látják az összes regisztrált felhasználó adatát. Szerkeszteni vagy bármilyen módon törölni nem lehetséges.

*Ételek*

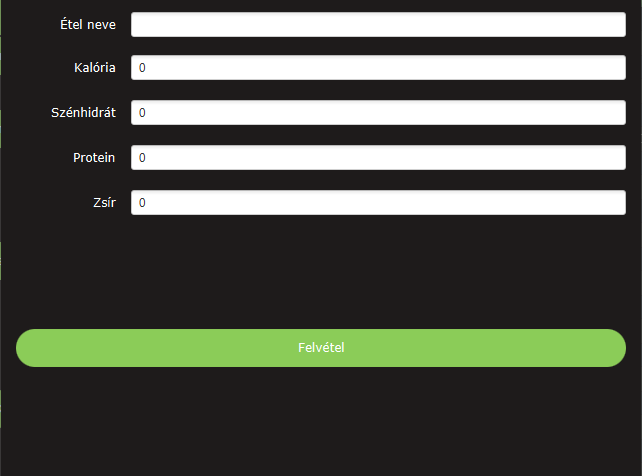
A képen képernyőkép, szöveg, képernyő, szoftver látható

Automatikusan generált leírásEzen a részen láthatja azokat az ételeket, amik bekerültek az adatbázisba. Bármikor lehet új ételt hozzáadni, vagy a meglévőket szerkeszteni.

Ezek az alablakok a kívánt gomb kattintása után nyílnak meg.

*Étel felvétele*

Az étel felvétele gomb kattintása után egy új ablak nyílik meg a meglévőre. A beviteli mezőkbe a megfelelő adatok beírása[[1]](#footnote-1) után a **Felvétel** gombra kattintva bekerül az adatbázisba. Ezután felvehet egy következő ételt vagy be is zárhatja az ablakot. Az imént felvett étel megjelenik a listázott ételek között. Esetleges duplikálás esetén a szerkesztés segítségével meg lehet változtatni az egyik étel adatait.



*Étel frissítése*

A frissíteni kívánt étel kiválasztásához kattintson rá az adattáblában az egyik ételre[[2]](#footnote-2), sikeres kiválasztás után az étel sorában levő szövegek zöld színűek lesznek. Ezután kattintson az étel frissítése gombra.



Az új ablak egy az egyben ugyanúgy néz ki, mint a felvétel, viszont a beviteli mezőkben a kijelölt étel adatai találhatóak[[3]](#footnote-3). Az átírt adatok után a **Frissít** gombra kattintás után az adatbázisban frissültek az étel adatai, a frissítés ablak pedig bezáródik.

*Statisztika*

Ezen a részen a felhasználók kérdőiv eredményeinek egy része van grafikonba rendezve. A statisztika csak arra szolgál, hogy a weboldal felhasználói milyen oknál fogva látogattak oda. Az oszlopdiagram alapján látható milyen arányban vannak a férfiak és nők, illetve felnőttek és fiatalabbak.

Statisztika


*Program bezárása*

Minden főablakon található egy Kilépés gomb, ezt megnyomva a program teljesen leáll. Az ablak tetején lévő X gomb is ugyan ezt a funkciót látja el.

**Fejlesztői dokumentáció**

**Forráskód:**

Az összes fájl és forráskód az src\main helyen található.

**A program felépítése:**

Ha ezek közül bármelyik fájl is hiányzik a

program nem megfelelően vagy nem fog futni.

**A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás**

*Controller.class*

Ez az osztály kezeli a hiba- és fő ablakok megjelenítését. Ha ez az osztály hiányzik, az alkalmazás nem tud elindulni.

A *SceneOpen* függvény hívódik meg mindegyik gomb megnyomása után, amely fő ablakot nyit meg. Várt inputok: gomb, fxml fájl neve, cím.

Három hiba függvény hívja meg az *alert* függvényt. Alapértelmezetten nincs hiba típusa, ezt a meghívó függvényeknél lehet megadni.

Várt inputok:

* alert: cím, fejléc szövege, tartalom szövege, hiba típusa
* error: fejléc szövege, tartalom szövege- meghíjva az *alert* függvényt
* error: fejléc szöveg- meghívja az előző *error* függvényt
* warning: fejléc szövege- meghívja az *alert* függvényt

*Admin.class*

Tartalmazza az admin bejelentkezéshez szükséges adatokat.

*AdminController.class*

A felhasználók megjelenítését kezeli. Tartalmazza a *loadUsers* függvényt, ami az adatbázisról lekéri az összes felhasználó adatát. Az *initalize* tölti bele a táblázatba az adatokat. A *click-*el kezdődő függvények a *Controller SceneOpen* függvényét hívják meg.

*Age.class*

A felhasználók korát tartalmazó osztály a kérdőív alapján, konkrét kort nem vesz fel, csak darabszámot.

*App.class*

Az alkalmazás indításánál ez az első osztály, ami lefut, ez nyitja meg a bejelentkező ablakot.

*Food.class*

Tartalmazza az ételek adatípusait, amelyek meghívódnak a *FoodController*-ben.

*FoodController.class*

Az ételek megjelenítését kezeli. Tartalmazza a *loadFoods* függvényt, ami betölti az ételeket az adatbázisból. Az osztály két új gombot is tartalmaz. Ezek a *Frissítés és Hozzáadás* gombok nem használják a SceneOpen függvényt.

*Gender.class*

A felhasználók nemét tartalmazó osztály a kérdőív alapján, konkrét nemet nem vesz fel, csak darabszámot.

*InsertFoodController.class*

Új étel hozzáadását teszi lehetővé az adatbázishoz. A *submit* függvény lefutása után az ablak nem záródik be, további ételekez is fel lehet venni anélkül, hogy újra meg kellene nyitni az ablakot. Bezárás után töltődik újra a táblázat.

*LoginController.class*

Tartalmazza a *loadAdmin* függvényt, ami betölti az adminok adatait a bejelentkezéshez. Helyes adatok megadása után a *loginClick* függvény meghívja a *SceneOpen* függvényt.

*Look.class*

A felhasználok testalkatát tartalmazó osztály a kérdőív alapján, konkrét testalkatot nem vesz fel, csak darabszámot.

*RequestHandler*

A szerver kommunikációhoz szükséges függvényeket tartalmazza. Az *addRequestBody* dolgozza fel a kapott adatokat. Várt inputok: http kapcsolat, adat.

A *getResponse* a *Response* osztály segítségévelkezeli és ellenőrzi a http kapcsolatot, majd visszaküldi a kapott választ. Várt input: http kapcsolat.

A *setupConnection* függvény állítja fel a kapcsolatot a szerver és az alkalmazás között. Várt input: URL.

A *get* függvény kéri le az adott táblát az adatbázisból a *setupConnection* és a *getResponse* függvény segítségével.Várt input: URL.

A *post* függvény feltölt adatot az adott táblába az *addRequestBody, setupConnection* és *getResponse* függvény segítségével.

A *put* függvény egy meglévő rekordot frissít az adott táblában az *addRequestBody, setupConnection* és *getResponse* függvény segítségével.

*Response.class*

A szerver által visszaadott http kódot és választ tartalmazza.

*StatController.class*

A felhasználok válaszai alapján statisztikát készít a nem, kor és testalkatról. Csak darabszámokat tartalmaznak a diagramok. Az oszlop-diagramban a kor és nem, a kör-diagramban pedig a testalkat statisztikája található.

*UpdatefoodController.class*

Meglévő étel adatainak megváltoztatását teszi lehetővé. Az *updateClick* függvény lefutása után a kiválaszott étel adatai frissülnek és az ablak bezáródik is újra meghívódik a *loadFoods* függvény.

*User.class*

A felhasználók adattípusait tartalmazza, amelyek meghívódnak az *AdminController-*ben.

*admin.fxml*

A felhasználók kilistázása ablak. Tartalmaz egy táblázatot és három gombot.

*food.fxml*

Az ételek kilistázása ablak. Tartalmaz egy táblázatot és öt gombot.

*insertfood.fxml*

Új étel hozzáadása ablak. Tartalmaz öt inputot és egy gombot.

*login.fxml*

Admin bejelentkezés ablak. Tartalmaz két inputot és egy gombot.

*stat.fxml*

Statisztika ablak. Tartalmaz egy oszlop-diagramot, egy kör-diagramot és három gombot.

*style.css*

Az alkalmazás stílus beállításait tartalmazza.

*updatefood.fxml*

Étel frissítése ablak. Tartalmaz öt input mezőt és egy gombot.

**Tesztelés dokumentáció**

|  |  |  |
| --- | --- | --- |
| **Elvégzett művelet** | **Kimenet** | **Várt kimenet** |
| Admin bejelentkezés hibás adatokkal | Hibaüzenet, nem enged tovább | Hibaüzenet, nem enged tovább |
| Admin bejelentkezés üres adatokkal | Hibaüzenet, nem enged tovább | Hibaüzenet, nem enged tovább |
| Admin bejelentkezés helyes adatokkal | Megjelenik a felhasználók kilistázása | Megjelenik a felhasználók kilistázása |
| Alkalmazás indítása backend nélkül | Hiba, az alkalmazás leáll | Hiba az alkalmazás leáll |
| Alkalmazás indítása backend mellett | Bejelentkezés ablak | Bejelentkezés ablak |
| Étel felvétele hibás értékekkel | Hibaüzenet, az étel nem töltődik fel | Hibaüzenet, az étel nem töltődik fel |
| Étel felvétele üres értékekkel | Hibaüzenet, az étel nem töltődik fel | Hibaüzenet, az étel nem töltődik fel |
| Étel felvétele megfelelő értékekkel | Az étel feltöltődik | Az étel feltöltődik |
| Étel frissítése hibás értékekkel | Hibaüzenet, az étel nem töltődik fel | Hibaüzenet, az étel nem töltődik fel |
| Étel frissítése üres értékekkel | Hibaüzenet, az étel nem töltődik fel | Hibaüzenet, az étel nem töltődik fel |
| Étel frissítése megfelelő értékekkel | Az ablak bezáródik és a frissült étel jelenik meg a táblázatban | Az ablak bezáródik és a frissült étel jelenik meg a táblázatban |
| Új ablak megnyitása valamelyik gomb segítségével backend nélkül | Hiba, majd az alkalmazás leáll | Hiba, majd az alkalmazás leáll |
| Új ablak megnyitása valamelyik gomb segítségével backend mellett | Az ablak megnyíli a meglévő pedig bezárul | Az ablak megnyílik a meglévő pedig bezárul |
| Kilépés gomb megnyomása | Az alkalmazás leáll | Az alkalmazás leáll |

1. A mezők nem lehetnek üresek, a szám mezők pedig csak nem negativ valós számokat fogadnak el. [↑](#footnote-ref-1)
2. Ha nincs kiválasztva étel a program nem engedélyezi a frissítés ablakot [↑](#footnote-ref-2)
3. A beviteli mezőkre ugyan azok a kitételek vonatkoznak, mint a felvételre. [↑](#footnote-ref-3)