

# PIZZMAN – felhasználói dokumentáció

Adminisztrációs szoftver pizzasütödék számára

## Program:

A szoftver alapvető feladata a pizzarendelések adatbázisának nyilvántartása, használata. Lehetőséget kínál adminisztrátori, szállítói és felhasználói kezelésre, azaz háromféle profilkör létezik: Admin, Deliverer és User. Az Adminoknak van csak teljeskörű hozzáférése a funkciókhoz. A program elején lehet Customerként regisztrálni, avagy bejelentkezni más minőségben.

## Profilkör: User (A következők menüpontok is az alkalmazáson belül)

- Felhasználónév és jelszó segítségével bejelentkezik/regisztrál.
- Lekérheti a rendeléseinek állapotát.
- Elindíthat rendelt.
- Itt lekérheti a pizzákat.
- Hozzáadhat pizzát. / Eltávolíthat felvett pizzát.
- Hozzáadhat egy pizzához plusz feltétet.
- Saját címére, vagy máshova kéri. (Default: saját)
- Választhat fizetési módot (Default: készpénz)
- (Leadás dátumát az alkalmazás generálja)
- Megjegyzés hozzáadása.
- Elküldheti a rendelt.

## Profilkör: Deliverer (Az ő adatait kézzel visszük be az adatbázisba)

- Felhasználónév és jelszó segítségével bejelentkezik.
- Magához vehet leadott rendeléseket.
- Itt lekérheti a „szállítható” rendeléseket.
- Rendelésazonosítóval lefoglalhatja a rendelt.
- Visszajelezhet kiszállított rendelésekről.
- Itt lekérheti a saját kiszállítás alatt lévő rendeléseit.
- Rendelésazonosítóval visszajelezheti, hogy sikeres volt-e a kiszállítás.

## Profilkör: Admin (Az ő adatai be vannak égetve adatbázisba)

- Felhasználónév és jelszó segítségével bejelentkezik.
- Lekérheti az összes létező pizzát.
- Lekérheti az összes kiszállítatlan rendelt.
- Átállíthat rendelt „frissen felvett”-ről „szállítható”-ra.
- Elindíthat rendelt. (Ugyanúgy, mint User)
- Lekérheti egy nap bevételét.
- Létrehozhat pizzát.
- Elnevezheti a pizzát.
- Létrehozhat itt új feltétet.

# PIZZMAN – programozói dokumentáció

Adminisztrációs szoftver pizzasütődék számára

## UML diagram:

*Lásd következő oldal, bele lehet nagyítani...*

## Megvalósítási ötletek:

- Teszteléshez a memtrace környezet bevetésre kerül.
- A profilok tárolásához és kezeléséhez – mivel azok heterogén kollekcióban tárolódnak – külső forrásból meríték ötletet ([https://prog2.cppftw.org/extra\\_heterogen/](https://prog2.cppftw.org/extra_heterogen/)). Csupán az én programom statikusan tartalmazza a prototípusokat.
- Bejelentkezés alkalmával a bejelentkezési adatokkal generálunk egy dummy Profile-t, a Profilok listájának find() metódusa pedig a lineáris keresés algoritmusával megszerzi a megfelelő Profile-t, amire ráeresztve az ő (virtuálisan megszerzett) verifyLogin(username, pw) metódusa fog visszatérni azzal (Rights), ami eldönti, milyen hozzáférési jogokat érhet el a bejelentkezett felhasználó.
- A főprogramban globálisan lesz lementve a bejelentkezett felhasználó jogköre (Rights), és listákban a beolvasott profilok (Profile), rendelések (Order), pizzák (Pizza), valamint feltétek (Topping).
- Minden fontos osztálynak van komparátor operátora, főleg, hogy a List find\_p() metódusa jól működhessen.
- List néven implementált egy láncolt lista saját bejáró osztállyal, amelyet indexelni is lehet (az indexelés a megfelelő pizza kiválasztásához szükséges új feltét hozzáadásakor).

## Fájlkezelés:

Külön fájlokba (táblákba) vannak rendezve: profilok, rendelések, pizzák, feltétek. Minden osztálynak van save() és load() metódusa pontosan ezen fájlokból való kiolvasásra és beolvasásra. A beolvasások az alkalmazás indításakor futnak le, a kiírások pedig a bezáráskor.