HAIRDRESS AIR

– FODRÁSZATI FOGLALÓRENDSZER –

FEJLESZTŐI ÉS FELHASZNÁLÓI DOKUMENTÁCIÓ

2025.

Tartalom

[BEVEZEZETÉS 3](#_Toc195651386)

[FEJLESZTŐI DOKUMENTÁCIÓ 4](#_Toc195651387)

[FELHASZNÁLT TECHNOLÓGIÁK 4](#_Toc195651388)

[A FEJLESZTŐI KÖRNYEZET 4](#_Toc195651389)

[ADATBÁZISTERV 5](#_Toc195651390)

[ADATMODELLEK 6](#_Toc195651391)

[OSZTÁLYOK 7](#_Toc195651392)

[REST API VÉGPONTOK 12](#_Toc195651393)

[FEJLESZTÉSI LEHETŐSÉGEK 14](#_Toc195651394)

[RENDSZERSPECIFIKUS ALGORITMUSOK 14](#_Toc195651395)

[FELHASZNÁLÓI DOKUMENTÁCIÓ 15](#_Toc195651396)

[RENDSZERKÖVETELMÉNYEK 15](#_Toc195651397)

[JOGOSULTSÁGI SZINTEK 15](#_Toc195651398)

[VENDÉG FELHASZNÁLÓI KÉZIKÖNYV 16](#_Toc195651399)

[FODRÁSZ FELHASZNÁLÓ KÉZIKÖNYV 17](#_Toc195651400)

[SUPER ADMIN FELHASZNÁLÓ KÉZIKÖNYV 17](#_Toc195651401)

# BEVEZEZETÉS

**A projekt célja**

Moduláris felépítésű, összetett fodrászati alkalmazás első, teljes értékű funkciójának, a fodrászatba érkező foglalásoknak a kezelése. Célunk, hogy a felhasználók webes felületen leadhassák foglalásukat a kiválasztott fodrász irányába, melyet a front office asszisztens vagy recepciós *superadminként* tud menedzselni.

A honlapot használni nem tudó felhasználók számára lehetőséget biztosítunk telefonos foglalásra, ebben az esetben a super admin tudja a regisztrációt és/vagy a foglalást rendezni.

Cél, hogy a rendszer törzsadatai adminként kezelhetőek, a vendégek számára lekérhetőek legyenek, ezt a szolgáltatások CRUD műveleteivel és azok árlistába történő beolvasásával biztosítjuk. A rendszer mobiltelefonról is kezelhető, minden menüpont és funkció elérhető.

**Fejlesztői csapat**

Kovács Jázmin – tesztelés

Ottó Attila – Backend programozás

Pál-Szabó András – Frontend programozás

# FEJLESZTŐI DOKUMENTÁCIÓ

## FELHASZNÁLT TECHNOLÓGIÁK

Backend:

* Laravel API
* Sanctum autentikáció
* XAMPP (PHPMyAdmin)

Frontend:

* Angular keretrendszer
* Bootstrap 5

Kiszolgáló programok:

* Krita (grafikák)
* Google Chrome, Firefox, Microsoft Edge (böngészők)
* Microsoft Visual Studio Code
* Insomnia
* Git

Operációs rendszerek:

* Windows 11
* Linux

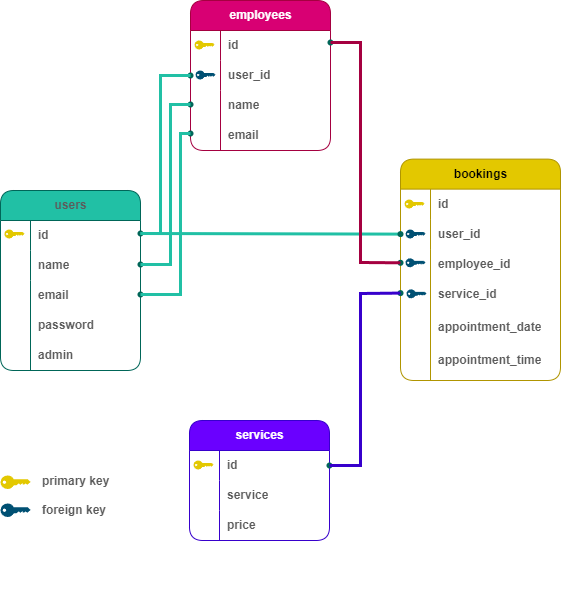
## A FEJLESZTŐI KÖRNYEZET

Az XAMPP és a Laravel a fejlesztés kezdetén felmerülő igényeket tökéletesen támogató kereteknek bizonyultak, ezért a foglaló backend részének építését ezeken a rendszereken kezdtük meg. A frontend oldalon az Angular CSS alapú kiszolgálójára esett a választásunk, mely a letisztult, modern felület felépítését szintúgy lehetővé tette. Ezt a Krita grafikai szoftverrel támogattuk meg, a *home* oldalon fogadó logó tervezését itt végeztük, a betűtípusokat a Google Fonts oldaláról szereztük be.

A tesztelést backend oldalon Insomnia szoftverrel, a frontend esetén többfajta, piacvezető böngészővel végeztük.

A backend fejlesztő Linux, míg a frontend fejlesztő Windows operációs rendszeren dolgozott, mindketten Visual Studio Code alkalmazásban, a Git verziókövető használata mellett.

## ADATBÁZISTERV

****

## ADATMODELLEK

**user**

* id (bigint) auto increment
* name (varchar)
* email (varchar)
* password (varchar)
* admin (int)

**employee**

* id (bigint) auto increment
* user\_id (biginit) foreign key)
* name (varchar)
* email (varchar) index
* phone (varchar, nullable)

**service**

* id (bigint) auto increment
* service (varchar)
* price (decimal)

**booking**

* id (bigint) auto increment
* user\_id (bigint) foreign key
* employee\_id (bigint) foreign key
* service\_id (bigint) foreign key

## OSZTÁLYOK

**AuthController**

Az AuthController osztály a rendszer felhasználóival kapcsolatos adminisztratív műveletekért felelős. Az itt található metódusok csak megfelelő jogosultságokkal rendelkező felhasználók (pl. super admin) számára elérhetők, amelyet a Laravel Gate jogosultságkezelője biztosít.

**Metódusok:**

* getUsers()
  + Lekéri az összes felhasználót az adatbázisból.
  + Csak *super admin* szerepkörrel elérhető.
  + Visszatérési érték: összes regisztrált felhasználó JSON formátumban.
* setAdmin(Request $request)
  + A megadott id alapján kiválasztott felhasználónak admin jogosultságot ad (admin = 1).
  + Csak *super admin* végezheti el.
* demotivate(Request $request)
  + Az id alapján kiválasztott felhasználótól admin jogosultságot von meg (admin = 0).
  + Csak *super admin* jogosult a végrehajtására.
* updateUser(Request $request)
  + A felhasználó nevét és e-mail címét módosítja.
  + Szintén csak *super admin* hajthatja végre.
* newUser(RegisterRequest $request)
  + Új felhasználót hoz létre a rendszerben.
  + Ha ez az első felhasználó, automatikusan *super admin* jogosultságot kap (admin = 2).
  + Egyébként az admin értéke 0 lesz (sima felhasználó).
  + Csak *admin* vagy magasabb jogosultság esetén hívható meg.
* avadaKedavra(Request $request)
  + A megadott id alapján törli a felhasználót az adatbázisból.
  + Csak *super admin* hajthatja végre.

**BookingController**

A BookingController osztály felel az időpontfoglalási rendszer backend-oldali kezeléséért. A kontroller kezeli a foglalások létrehozását, lekérdezését, módosítását és törlését, különböző jogosultsági szintek figyelembevételével. A metódusok igyekeznek megkülönböztetni az admin, dolgozó (admin = 1) és ügyfél (admin = 0) jogosultságú felhasználókat.

**Metódusok:**

* getBookings()
  + Visszaadja a felhasználóhoz kapcsolódó foglalásokat:
    - *admin = 0*: saját foglalásokat kapja meg.
    - *admin = 1*: az adott alkalmazott foglalásait látja.
    - *admin = 2* (super admin): az összes foglalást látja.
  + A lekérdezés során kapcsolt entitásokat (user, employee, service) is visszaad, feldolgozott formátumban.
* getOneBooking(Request $request)
  + Egyetlen foglalást keres ki a booking mező alapján.
  + Ha a foglalás nem található, hibával tér vissza.
* addBooking(BookingRequest $request)
  + Új foglalást hoz létre a rendszerben.
  + Csak *admin vagy super admin* jogosultság esetén hajtható végre.
  + Lekéri a kapcsolódó ügyfél, dolgozó és szolgáltatás azonosítókat más kontrollereken keresztül.
* updateBooking(BookingRequest $request)
  + Egy meglévő foglalást módosít az id alapján.
  + Csak *admin vagy super admin* végezheti el.
  + A kapcsolódó entitásokat szintén újra hozzárendeli az aktualizálás során.
* deleteBooking(Request $request, $id)
  + A foglalást törli az id alapján.
  + Sikeres törlés esetén visszaadja a törölt rekordot.
* store(Request $request)
  + Új foglalást rögzít az appointment\_date és appointment\_time mezők alapján.
  + Előtte ellenőrzi, hogy az adott fodrásznál az időpont már foglalt-e.
  + Ütközés esetén hibaüzenetet ad vissza, egyébként létrehozza a foglalást.
  + Ez a metódus a frontendből érkező foglalási igények kezelésére alkalmas.
* getBookedTimes($stylistId, $date)
  + Visszaadja egy adott fodrász (employee\_id) foglalt időpontjait egy adott napra (appointment\_date).
  + A válasz JSON formátumban, csak az időpontokat tartalmazza (H:i formátumban).

**EmployeeController**

A EmployeeController osztály a rendszerhez tartozó dolgozók (alkalmazottak) kezeléséért felel. Lehetővé teszi a dolgozók listázását, létrehozását, azonosítását, törlését, valamint a felhasználóból dolgozóvá konvertálást. A metódusok a User és Employee modellek közötti kapcsolatot is kezelik.

**Metódusok:**

* getEmployees()
  + Lekéri az összes dolgozót az employees táblából, csak az id és name mezőkkel.
  + Az eredmény JSON formátumban tér vissza, success kulccsal ellátva.
* getEmployeeId($employeeName)
  + Egy dolgozó azonosítóját (id) adja vissza a megadott név alapján.
  + Hibát jelez, ha nincs ilyen nevű dolgozó.
* addEmployeeByUserId(Request $request, $userId)
  + Létrehoz egy új dolgozót a megadott user\_id alapján.
  + Először ellenőrzi, hogy a felhasználó létezik-e, és hogy még nincs-e már rögzítve dolgozóként.
  + A dolgozó adatai a User tábla megfelelő mezőiből származnak, opcionálisan telefonszámmal kiegészítve.
* removeEmployeeByUserId($userId)
  + Törli a dolgozót az employees táblából a megadott user\_id alapján.
  + Hibával tér vissza, ha az adott felhasználó nem szerepel a dolgozók között.

**ServiceController**

A ServiceController osztály a rendszerben elérhető szolgáltatások kezeléséért felel. Lehetővé teszi a szolgáltatások listázását, létrehozását, módosítását, törlését, valamint az azonosítójuk lekérdezését név alapján. A kontrollert elsősorban adminisztratív funkciók használják a szolgáltatási kínálat karbantartására.

**Metódusok:**

* getServices()
  + Lekéri az összes szolgáltatást az adatbázisból.
  + A válasz formázva érkezik: minden szolgáltatás neve, azonosítója (id) és ára (price) szerepel a listában.
* getService(Request $request)
  + Egyetlen szolgáltatást keres meg a szolgáltatás neve alapján (service mező).
  + Ha nincs ilyen szolgáltatás, hibaüzenettel tér vissza.
* addService(Request $request)
  + Új szolgáltatást hoz létre, ha az ár és név mezők validálása sikeres.
  + A szolgáltatás neve (service) és ára (price) kerül mentésre.
* updateService(Request $request, $id)
  + Egy meglévő szolgáltatás módosítását végzi el az id alapján.
  + Ha a szolgáltatás nem található, 404-es hibaüzenetet ad vissza.
  + A módosítás az összes beküldött mezőre kiterjed ($request->all()).
* deleteService($id)
  + Törli az adott szolgáltatást az id alapján.
  + Ha sikeres a törlés, visszatér egy success: true üzenettel, egyébként 404-es hibával.
* getServiceId($serviceName)
  + Egy szolgáltatás azonosítóját (id) adja vissza a megadott név (service) alapján.
  + A metódus hasznos más kontrollerek (pl. BookingController) számára, ahol az id szükséges a kapcsolatok létrehozásához.

**UserController**

A UserController osztály a felhasználók hitelesítéséért (auth) felelős. Lehetővé teszi a regisztrációt, a bejelentkezést és a kijelentkezést, valamint kezeli a bejelentkezési próbálkozások korlátozását is. A controller több másik controllerrel (pl. BannerController, MailController) is együttműködik a bejelentkezési logika finomhangolása érdekében.

**Metódusok:**

* register(RegisterRequest $request)
  + Új felhasználó regisztrálása.
  + Az első regisztrált felhasználó automatikusan *super admin* jogosultságot kap (admin = 2), a többiek alapértelmezés szerint sima felhasználók lesznek (admin = 0).
  + A jelszó bcrypt titkosítással kerül tárolásra.
* login(LoginRequest $request)
  + Hitelesíti a felhasználót email és jelszó alapján.
  + Hibás bejelentkezések számlálása és tiltás időzítése BannerController segítségével történik.
  + Sikeres bejelentkezés esetén egy új API token kerül generálásra (Sanctum token).
  + Visszaadott adatok: felhasználó emailje, jogosultsága (admin), token és a tiltás lejártának időpontja (ha volt korábban tiltás).
* logout()
  + Az aktuálisan bejelentkezett felhasználó kijelentkeztetése.
  + A bejelentkezési token (Sanctum) törlésre kerül.
  + Visszatérési érték: a kijelentkezett felhasználó neve.

## REST API VÉGPONTOK

|  |  |  |
| --- | --- | --- |
| Végpont | Metódus | Leírás |
| /register | POST | Felhasználó regisztrációja |
| /login | POST | Felhasználó bejelentkezése |
| /logout | POST | Felhasználó kijelentkezése |
| /getusers | GET | Összes felhasználó (superadmin számára) |
| /newuser | POST | Új felhasználó létrehozása admin jogosultsággal |
| /admin/{id} | PUT | Admin jog megadása |
| /polymorph/{id} | PUT | Admin jog megvonása |
| /updateuser/{id} | PUT | Felhasználó adatainak frissítése |
| /voldemort/{id} | DELETE | Felhasználó törlése |
| /bookings | GET | Foglalások listázása (jogosultságtól függően szűrt) |
| /onebooking/{id} | GET | Egy foglalás lekérdezése ID alapján |
| /addbooking | POST | Foglalás rögzítése (admin/superadmin) |
| /updatebooking/{id} | PUT | Foglalás frissítése |
| /deletebooking/{id} | DELETE | Foglalás törlése |
| /bookings | POST | Foglalás létrehozása frontend felől |
| /booked-times/{stylistId}/{date} | GET | Foglalt időpontok lekérdezése fodrász szerint |
| /employees | GET | Dolgozók listázása |
| /add-employee/{id} | POST | Felhasználó dolgozóként történő rögzítése |
| /remove-employee/{id} | POST | Dolgozó eltávolítása |
| /services | GET | Szolgáltatások listázása |
| /addservice | POST | Új szolgáltatás rögzítése |
| /updateservice/{id} | PUT | Szolgáltatás frissítése |
| /deleteservice/{id} | DELETE | Szolgáltatás törlése |

## FEJLESZTÉSI LEHETŐSÉGEK

A rendszer jelenleg egyetlen fodrászatra, egy super adminra és mindössze pár fodrászra van optimalizálva. Nincs e-mail küldés, mert a jelenlegi létszámnál a működés admin felületen átlátható.

Amennyiben több telephely vagy több franchise tartana igényt egy közös felületre, arra is készen áll a rendszer. Ebben az esetben az e-mailek küldése minden, UserControllerben és BookingControllerben meghatározott metódus eleme lenne.

Következő lépésként a user tábla nullable mezőkkel való bővítését tervezzük. Itt a regisztrációt követően egy saját adatok menüpont áll majd a foglaló rendelkezésére, amelyen a fodrászok számára releváns adatokat is kitölthet a telefonszámtól a hajtípuson át a számlázási adatokig.

Tervezzük a szolgáltatások időtartamának beállítását, valamint a legördülő választék pipálós szisztémára való leváltását, amellyel egyszerre több szolgáltatás is rendelhetővé válik.

Vizuális, személyre szabott naptárral, valamint külön kijelzőre tehető, egyedi, napi foglalási táblával kívánjuk komfortossá tenni a fodrászok munkáját. Szintén ugyanők egy időpontnyilvántartó rendszerben vezethetik a szabadságaikat, ezeken a napokon az időpontok nem foglalhatók, az esetlegesen előre rögzített foglalásokról a foglalók részére szintén e-mail értesítést küldünk.

Végső cél a raktárnyilvántartás bevezetése, de egyelőre ez nem szerepel a távlati project scope-jában.

## RENDSZERSPECIFIKUS ALGORITMUSOK

A rendszer jelenleg egységesen fél órás időtartamokban teszi lehetővé a foglalásokat. A /booked-times/{stylistId}/{date}végponton az egyelőre frontenden rögzített időpontok frissülnek annak megfelelően, hogy az adott napon az adott fodrász neve mellett milyen időbélyegek szerepelnek. Ugyanitt a magyar ünnepnapokat is vezetjük, amelyek a hétvégékkel egyetemben nem választható dátumok.

A **CRUD** műveletek maradéktalanul a **Szolgáltatások** menüpont alatt valósulnak meg, ahol rögzíthetünk, törölhetünk és módosíthatunk is, kizárólag super adminként. Az itt vezetett szolgáltatások olvasásra minden látogató számára elérhetővé válnak olvasásra az **Árlista** menüpont alatt.

# FELHASZNÁLÓI DOKUMENTÁCIÓ

## RENDSZERKÖVETELMÉNYEK

* aktív internetkapcsolat
* webböngésző (Chrome, Firefox, Edge, Safari, stb)
* minimum gépkövetelmény (PC, laptop, mobileszköz):
  + 2 GHz-es processzor
  + 4 GB RAM

## JOGOSULTSÁGI SZINTEK

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **regisztrált felhasználó** | | |
| Funkció | **látogató** | **vendég** | **fodrász** | **super admin** |
| Regisztráció új névvel | ✓ | ✓ | ✓ | ✓ |
| Kezdőlap megtekintése | ✓ | ✓ | ✓ | ✓ |
| Árlista megtekintése | ✓ | ✓ | ✓ | ✓ |
| Bejelentkezés | x | ✓ | ✓ | ✓ |
| Foglalás saját névvel | x | ✓ | ✓ | ✓ |
| Saját névre szóló foglalások megtekintése | x | ✓ | ✓ | ✓ |
| Két napon túli, saját névre szóló foglalás törlése | x | ✓ | ✓ | ✓ |
| Két napon belüli, saját névre szóló foglalás törlése | x | x | ✓ | ✓ |
| Más nevére szóló foglalás törlése | x | x | x | ✓ |
| Felhasználó jogosultsági szintjének állítása | x | x | x | ✓ |
| Szolgáltatás hozzáadása | x | x | x | ✓ |
| Szolgáltatás módosítása | x | x | x | ✓ |
| Szolgáltatás törlése | x | x | x | ✓ |

## VENDÉG FELHASZNÁLÓI KÉZIKÖNYV

A kezdőlap bal felső sarkában található **HAIRDRESS AIR** feliratra kattintva mindig visszatérünk a kezdőlapra.

Az árlista megtekintéséhez kattintsunk az **Árlista** elemre. Amennyiben a látott szolgáltatások elnyerték tetszésünket, és szeretnénk foglalni, első körben regisztrálnunk kell, amennyiben még nem tettük meg. Ehhez kattintsunk a jobb felső sarokban található **Regisztráció** gombra.

Itt adjuk meg saját nevünket, majd saját e-mail címünket! Válasszunk jelszót. A jelszónak az alábbi kritériumok mindegyikét teljesítenie kell: legalább 8, legfeljebb 20 karakter hosszú; tartalmaz legalább 1 nagybetűt; tartalmaz legalább 1 kisbetűt; tartalmaz legalább 1 számot.

A jelszó megerősítése mezőbe írjuk be újra jelszavunkat, majd kattintsunk a **Regisztrálok** gombra! A sikeres regisztrációt követően beléphetünk a jobb felső sarokban lévő **Bejelentkezés** feliratra kattintva. Itt írjuk be a regisztráció során megadott e-mail címünket és jelszavunkat, majd kattintsunk a **Bejelentkezés** gombra!

Sikeres bejelentkezés után ismét a kezdőlapon találjuk magunkat, és nevünk megjelenik a jobb felső sarokban.

Amennyiben új foglalást szeretnénk leadni, kattintsunk az **Új foglalás** feliratra! Az e-mail címünk automatikusan kitöltésre kerül, első körben válasszunk fodrászt és szolgáltatást a legördülő menüből! Ezek után megadhatjuk a választott napot kézzel beírva (év, hónap, nap sorrendben, például: 2025.04.01.) vagy a kis naptár ikonra kattintva. A választott fodrásznál még elérhető időpontok legördülő listájából válasszuk ki a nekünk tetszőt a Foglalás időpontja mezőből, amennyiben nem találunk megfelelőt, választhatunk másik fodrászt és másik napot, az elérhető időpontok listája ennek megfelelően frissül. A **Lefoglalom** gomb megnyomásával véglegesíthetjük a foglalásunkat. **Figyelem!** A fodrászat hétvégén és ünnepnapokon zárva tart, ezekre a napokra nem tudunk foglalni!

A foglalásunk a **Foglalások** menüpontban jelenik meg. Ide kattintva a fehér kártyákon a jövőbeli, az alsó sávban, listában pedig a múltbéli foglalásainkat tekinthetjük meg. Azon foglalásokat, melyek két napon túl esedékesek, törölhetjük a piros hátterű X jelre kattinva. A két napon belüli foglalásoknál ez az opció nem elérhető, ilyen esetben kérjük, keresse fodrászatunkat a **kezdőlapon** látható telefonszámon!

Amennyiben szükséges, kijelentkezhetünk a jobb felső sarokban látható **Kijelentkezés** feliratra kattintva.

## FODRÁSZ FELHASZNÁLÓI KÉZIKÖNYV

Kérem, olvassa el a 11. pontban lévő **Vendég felhasználói kézikönyvet** a funkciók megismeréséhez! A Foglalások menüpont alatt a két napon belüli foglalások is törölhetők. Amennyiben szükséges időpont fenntartása, kérjük, az Új foglalás menüpont alatt válassza a saját nevét, és foglalja le az időpontot, ezáltal a lehetőség eltűnik a vendégek elől.

## SUPER ADMIN FELHASZNÁLÓI KÉZIKÖNYV

**FONTOS!** A Hairdress Air alkalmazás első regisztráló felhasználója lesz a rendszer super adminja. Több super admin beállításához vagy a jog átruházásához kérjük, keresse fel a rendszer fejlesztőjét!

Kérem, olvassa el a 11. pontban lévő Vendég felhasználó kézikönyvet a funkciók egy részének megismeréséhez! A foglalások menüpont alatt látható valamennyi, a rendszerben keletkezett foglalás, a jövőbeli eseményeket vendégtől, fodrásztól, szolgáltatástól és dátumtól függetlenül törölheti. Jelenleg a foglalás módosítására nincs lehetőség, amennyiben ilyen kérés érkezik, kérem, törölje az eredeti foglalást, és kérje meg a vendéget az új foglalás megtételére! Ezt azonban Ön is elvégezheti.

Az **Új foglalás** menüpont alatt válassza ki annak a felhasználónak az e-mail címét, akinek a nevére rögzíteni kívánja az időpontot. A rögzített foglalás a vendég saját adatlapján is meg fog jelenni. Amennyiben nem biztos a vendég e-mail címében, azt ellenőrizheti a **Felhasználók** menüpont alatt, ahol valamennyi, a Hairdress Air alkalmazásban regisztrált felhasználót megtalálja.

A Felhasználók menüpont alatt lehetősége van rögzíteni a dolgozókat. Ehhez az előzetesen regisztrált fodrász neve mellett, a táblázat jobb oldalán lévő műveletek szakaszban kattintson a **Feljogosítás** gombra! Ezzel a felhasználó rögzítésre kerül a fodrászok között, és válaszhatóvá válik az **Új foglalás** menüpont alatt.

Szolgáltatások kezeléséhez kattintson a **Szolgáltatások** menüpontra! Itt láthatja a rendszerben rögzített szolgáltatásokat. Új szolgáltatás rögzítéséhez kattintson az Új szolgáltatás felvétele gombra! Értelem szerint töltse ki a mezőket! Az Ár mező csak számokat fogad el. Miután végzett, kattintson a Szolgáltatás hozzáadása gombra! Amennyiben az új sorban üres értékeket talál, frissítse az oldalt az F5 gomb megnyomásával! Ezzel az új szolgáltatás megjelent az **Árlista** menüpontban is.

Módosításhoz válassza ki a módosítani kívánt szolgáltatást, és kattintson a Módosítás gombra! Itt lehetősége van új nevet és új árat beállítani. A szolgáltatás törléséhez kattintson a Törlés gombra!