**Eötvös Loránd Tudományegyetem**

**Informatikai Kar**

**Programtervező Informatikus szak**

**Parkoló helyeket nyilvántartó rendszer**

**Témavezető:**

Dr. Nikovtis Tibor

adjunktus, Ph.D.

**Szerző:**

Hováth Kristóf

Programtervező informatikus BSc.

Budapest, 2020

Tartalomjegyzék

[1. Bevezetés 3](#_Toc39829425)

[2. Felhasználói dokumentáció 4](#_Toc39829426)

[Telepítés 4](#_Toc39829427)

[Webes felület használata. 4](#_Toc39829428)

[**Bejelentkezési képernyő.** 5](#_Toc39829429)

[**Parkolóházak listáját tartalmazó képernyő** 6](#_Toc39829430)

[**Parkolóház adatait tartalmazó képernyő** 6](#_Toc39829431)

[**Parkoló adatait tartalmazó képernyő** 7](#_Toc39829432)

[**Saját adatok** 8](#_Toc39829433)

[**Nyilvántartás** 9](#_Toc39829434)

[**Napló** 10](#_Toc39829435)

[Mobil alkalmazás 10](#_Toc39829436)

[**Bejelentkezés** 10](#_Toc39829437)

[**Parkolóházak listáját tartalmazó képernyő** 11](#_Toc39829438)

[**Saját adatok képernyő** 11](#_Toc39829439)

[3. Fejlesztői dokumentáció 12](#_Toc39829440)

[Architektúra 12](#_Toc39829441)

[Háttérszolgáltatás 12](#_Toc39829442)

[Maven 12](#_Toc39829443)

[Adatbázis 14](#_Toc39829444)

[Modellek 14](#_Toc39829445)

[Repositorik 17](#_Toc39829446)

[4. Összefoglalás 18](#_Toc39829447)

[5. További fejlesztések 19](#_Toc39829448)

# Bevezetés

Bevezetés. Parkoló rendszerekre szükség van.

# Felhasználói dokumentáció

A Parkoló Tároló egy parkoló helyeket nyilvántartó és kezelő rendszer, amely a felhasználóknak segít felmérni az egyes parkolóházakban található szabad parkolóhelyeket, segít megtalálni hova lehet beparkolni és megjegyzi, hogy hova parkoltunk. Minden felhasználó kezelhet legfeljebb öt autót, amikkel beparkolhat, illetve kiparkolhat a nyilvántartott parkolóhelyekből.

A rendszert két féle felhasználók használhatják. Az egyszerű felhasználók, akik hozzáférnek a parkolókkal kapcsolatos adatokhoz és használhatják a parkolókat, vagyis a saját autóikkal ki- és beparkolhatnak, és az adminisztrátor jogkörrel rendelkező felhasználók, akik ezeken felül hozzáférnek egy naplóhoz, amin keresztül a rendszerben történő eseményeket visszanézhetik, más felhasználók nevében cselekedhetnek, valamint szerkeszthetik is a rendszerben lévő adatokat. Hozzáadhatnak új parkolóhelyeket, törölhetnek parkolóházakat, megváltoztathatják azok neveit stb.   
(Ezen kívül van még egy féle felhasználó, aki csak kis mértékben tér el az adminisztrátoroktól. Ez a fajta jogkör később kerül kifejtésre a Nyilvántartás felület részletezésénél.)

A rendszerrel való interakcióra egy webes, böngészőből elérhető felület, valamint egy mobil applikáció nyújt lehetőséget.

A következőkben a rendszer használatának részletes leírása kerül tárgyalásra.

## Telepítés

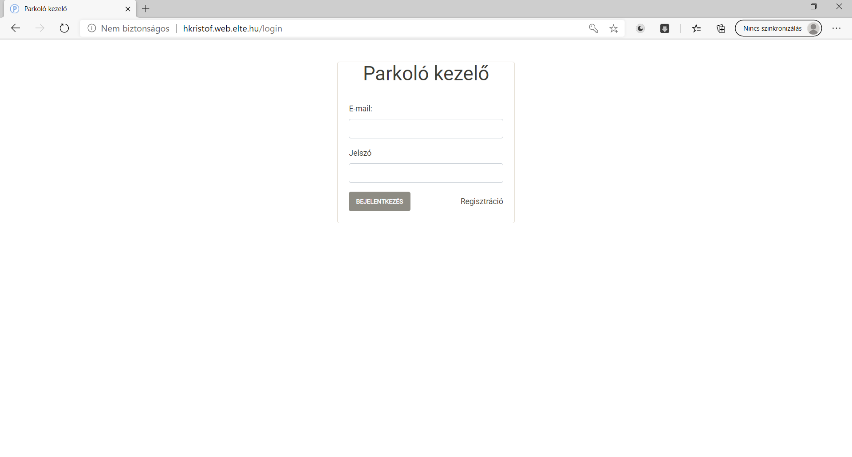
Telepítési útmutató…

## Webes felület használata.

A böngészőben elérhető alkalmazás elsősorban az adminisztrátoroknak szól. Csak itt kérdezhetőek le a naplóbejegyzések, és innen lehet más felhasználók nevében tevékenykedni.

A következőkben menüpontról menüpontra kerül részletes kifejtésre melyik funkció mire alkalmas és hogyan érhető el.

### **Bejelentkezési képernyő.**



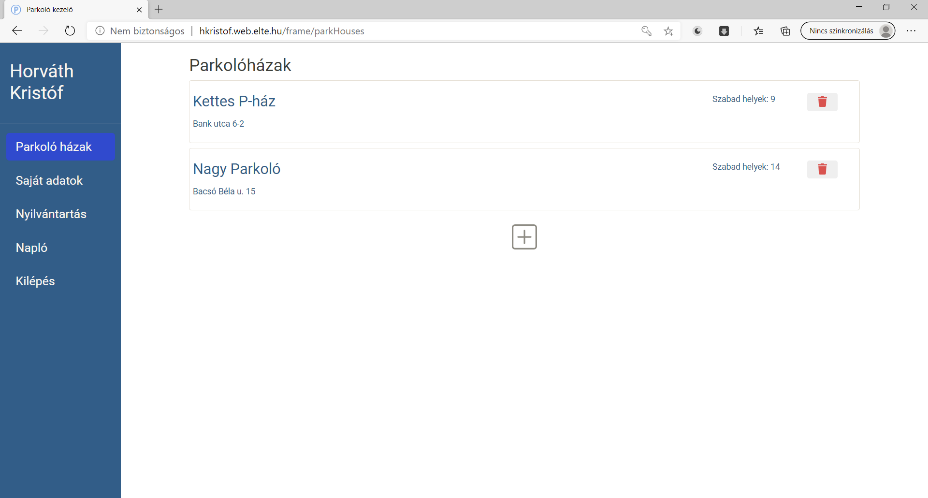
1. ábra

Bejelentkező képernyő

Az oldalt megnyitva legelőször egy hagyományos bejelentkező felület jelenik meg. A bejelentkező űrlap egy email címet és egy jelszót kér a felhasználótól az azonosításhoz. Ha a felhasználó még nem regisztrált be a rendszerbe akkor a jobb alul lévő „Regisztáció” feliratú gombra kattintva az űrlap ált alakul a regisztrációs műveletet elősegítő űrlappá. Az itt megjelenő mezők segítségével be lehet vinni vezeték- illetve keresztnevet, emailt és jelszót (amit kétszer kell beírni a két jelszó mezőbe). Ha mind a két jelszó mezőben szereplő szöveg megegyezik akkor a „Küld” gombra kattintva rögzülnek a bevitt adatok a rendszerben, majd az űrlap ismét a bejelentkezéshez használatos formát veszi fel. Itt ha a felhasználó egy létező email cím és a hozzá tartozó érvényes jelszót megadva rányom a „Bejelentkezés” gombra akkor a rendszer belépteti és átirányítja az alkalmazás kezelőfelületére. Ellenkező esetben piros hiba üzenettel tájékoztatja az alkalmazás a felhasználót, hogy nem sikerült belépni.

Bejelentkezés után függ az adott felhasználó jogaitól, hogy mi jelenik meg. Bizonyos funkciók le vannak tiltva az egyszerű felhasználók elől. Ezek mindig az adott rész tárgyalásánál fognak kifejtésre kerülni.

### **Parkolóházak listáját tartalmazó képernyő**



*2. ábra*

*Parkolóház lista*

Minden felhasználó esetében a kezdő oldal a jelenleg a rendszerben tárolt parkolóházak listája. Az adminisztrációs jogkörrel rendelkező felhasználók itt tudnak új parkolóházat felvenni a rendszerbe a lista alatt található plus jelet ábrázoló gombbal. A gombra rákattintva megjelenik egy felugró ablak az új parkolóház létrehozásához szükséges űrlappal. Ha kitöltésre került az összes szükséges mező akkor a „Ok” gomb megnyomásával eltűnik a felugró ablak és a listában megjelenik az új parkolóház.   
Az adminisztrátor joggal bíró felhasználók számára elérhető minden parkolóház listaelemének a végén egy szemetest ábrázoló gomb. Ezzel értelemszerűen el lehet távolítani az adott parkolóházat a rendszerből, annak minden parkolójával együtt. A törlés gombra kattintva megjelenik egy felugró ablak, ami rákérdez meg egyszer, hogy biztosan ki akarja e törölni a felhasználó a parkolóházat. Igenre kattintva eltűnik az ablak, valamint a parkolóház a listából.

Az összes felhasználó képes rákattintani az egyes parkolóházakra ezzel egy új oldalra lépve az adott parkolóház részletesebb adataival.

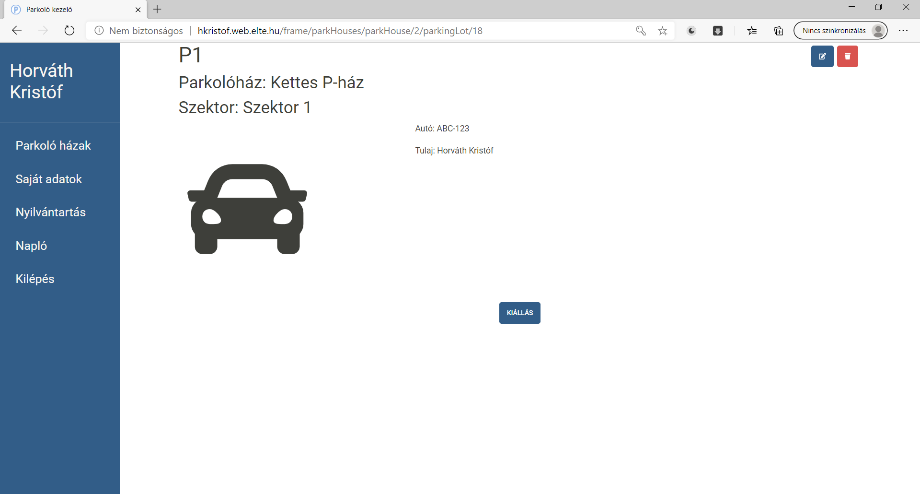
### **Parkolóház adatait tartalmazó képernyő**

<képernyőkép>

Ezen a felületen a kiválasztott parkolóház adatai találhatók. A felső részen látható a parkolóház neve, címe, valamint a benne megtalálható elemetek számozásának terjedelme.   
Adminisztrációs jogkörrel rendelkező felhasználóknak itt lehetőségük van szerkeszteni a parkolóház nevét, címét a jobb felül található ceruza ikonnal ellátott gomb segítségével. A gomb megnyomásával felugró ablakban megjelenik az űrlap, amin keresztül megváltoztathatóak az adatok. A szerkesztés gomb mellett található egy törlés gomb, aminek segítségével innen is törölhető a parkolóház. Ebben az esetben törlés után vissza navigál az alkalmazás a parkolóházak listájához.

Ezek alatt szerepelnek a megtalálható szektorok. Ezek emeletek szerint sorba rendezve jelennek meg. A szektorra rákattintva egy legördülő panel jelenik meg ami tartalmazza az adott szektorhoz tartozó parkolóhelyeket, ezek szintén név szerinti sorrendben jelennek meg. Amelyik parkoló el van foglalva azon egy autó ikon jelenik meg. Egy parkolóra kattintva át navigál a rendszer annak a parkolónak a részletes adatainak oldalára.

### **Parkoló adatait tartalmazó képernyő**



*3. ábra*

*Parkoló adatai képernyő*

Ezen a felületen az adott parkoló részletes adatai láthatóak. Itt van lehetőség beállni az adott parkolóba vagy éppen kiállni, ha már foglaljuk azt. Az adminisztrátorok beállhatnak más felhasználók nevében és ki is álhatnak helyettük. Sima felhasználók csak a saját autójukkal tevékenykedhetnek.

Jobb felül találhatóak a parkoló szerkesztésére és törlésére alkalmas gombok. Szerkeszteni a parkoló nevét lehet. A ceruza ikonnal ellátott gombra kattintva megjelenik egy felugró ablakban az erre alkalmas szövegmező.

Ha a parkolóban nem áll autó akkor középen mindössze egy „Üres” felirat látható, alatta egy beállás gombbal. Arra rákattintva, ha a felhasználó egyszerű felhasználó akkor megjelennek a saját autóinak a rendszámai egy felugró listában. Az egyikre rányomva eltűnik a felugró ablak és a parkoló el lesz foglalva az által az autó által. Ha a felhasználó egy adminisztrátor akkor először a felhasználók listája jelenik meg (ebben a listában abc sorrend szerint vannak rendezve a felhasználók, de mindig a bejelentkezett felhasználó van legfelül). Ha rányom a listában található egyik névre akkor annak a felhasználónak az autói jelennek meg. Ha nincs autója a választott felhasználónak akkor azt kiírja az alkalmazás. Ezután ugyanúgy, mint az egyszerű felhasználók esetében rá lehet nyomni valamelyik autóra, amelyet ennek hatására beparkoltnak fog tekinteni a rendszer.

Ha a parkolóban áll autó akkor azt egy nagy autó ikonnal jelzi a rendszer, valamint kiírja mellé az autó tulajdonosát is. Egyszerű felhasználók ilyenkor, ha a saját autójuk áll a parkolóban akkor van lehetőségük kiállni a „kiállás” feliratú gomb segítségével. Ha nem az ő autójuk áll bent akkor nem tudnak semmi változtatást csinálni ezen a felületen. A „kiállás” gomb nem jelenik meg nekik. Adminisztrátoroknak ellenben megjelenik akkor is, ha nem az ő autójuk áll bent.

### **Saját adatok**

<képernyőkép>

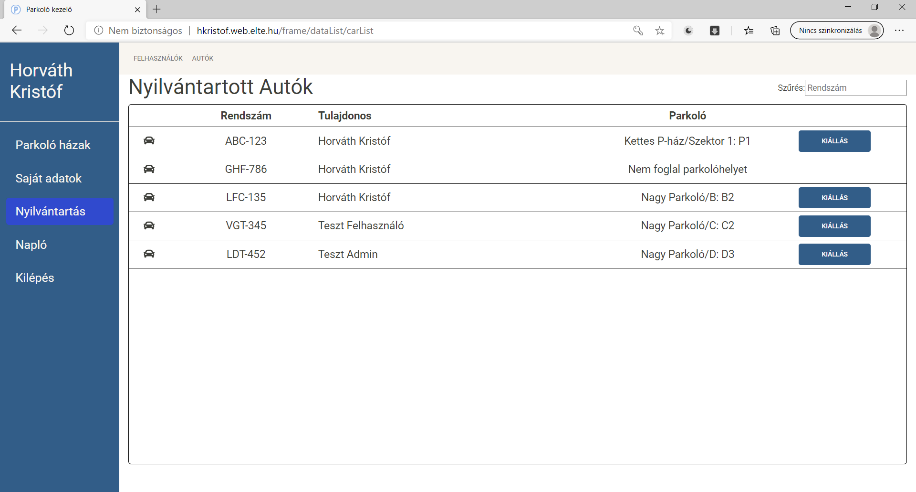
A saját adatok felületet a bal oldalon található menüben tudjuk elérni. Ez az oldal tartalmaz minden információt a bejelentkezett felhasználóról. Látható a neve, email címe, hogy adminisztrátor e vagy sem, illetve az autóinak a listája is itt szerepel.

Jobb felül a „Jelszó változtatás” gombbal lehetőség van jelszót változtatni. Rá kattintva megjeleni az ehhez szüksége űrlap egy felugró ablakon keresztül. Itt meg kell adni a régi jelszót és kétszer az újat. Ha nem egyeznek a jelszavak vagy nem helyes a régi jelszó akkor piros szöveggel jelzi ezt a rendszer. Ellenkező esetben megtörténik a változtatás és bezáródik a felugró ablak.

Ki lehet választani az egyes autókat felhasználó autóinak listájából. Ha rákattintunk az egyikre akkor a jobb oldali felületen megjelennek a hozzá tartozó információk. Hogy ha bent áll egy parkolóban akkor le lehet olvasni, hogy melyik parkolóház melyik szektorában található. Ilyenkor megjelenik itt is egy kiállást lehetővé tevő gomb, melynek funkciója és működése megegyezik a parkoló adatait tartalmazó képernyőn lévő „Kiállás” gombbal. Ugyan itt van lehetőség törölni is az adott gépjárművet, a szemetes ikonnal ellátott gomb segítségével. Ha az autó nem foglal éppen parkolóhelyet akkor ezt közli velünk a felület, és nem jelenik meg a „Kiállás” gomb.

Ha a felhasználónak nincsen összesen öt autója akkor van lehetősége új autót felvennie az autók listája alatti plusz jellel ellátott gomb segítségével. Rákattintva megjelenik a felugró ablak, ahol az új autó rendszámát megadva hozzá adásra kerül az új autó. Ha a felhasználó autóinak a száma elérte az ötöt akkor nincs lehetősége többet hozzáadni, az ehhez szükséges gomb nem jelenik meg többé.

### **Nyilvántartás**



*4. ábra*

*Nyilvántartás képernyő*

Ehhez a felülethez csak az adminisztrátorok férnek hozzá. Itt érhetőek el a rendszerben nyilván tartott felhasználók és autók.

#### Felhasználók

Idenavigáláskor elsőként a felhasználók listája jelenik meg. Leolvasható innen az illetők nevei, email címei, és hogy mennyi autót kezelnek. Jobb felül van lehetőség konkrét felhasználó nevére is keresni. Rákattintva valamelyik felhasználóra egy a Saját adatok-hoz hasonló felületre irányít át a rendszer. Itt meg lehet nézni a tényleges autókat és hogy esetlegesen melyik parkolóban állnak. Más felhasználó autójának törlésére adminisztrátorként sincs lehetőség.

Jogosultságokat szerkeszteni a „Jogosultság változtatása” feliratú gombbal lehet, ami felhasználók listájának egyes elemeinek a végén található. Ha az adott felhasználó egyszerű felhasználó akkor gombra nyomáskor a felugró ablak arra fog rákérdezni, hogy adminisztrátorrá szeretné e tenni, ha már adminisztrátor akkor pedig, hogy el akarja e venni ezt a titulust? „Ok” gombra kattintva eltűnik a felugró ablak és végbe megy a változtatás.

Az adminisztrátorok között van egy kitüntetett jogkör, ez a Fő Adminisztrátor. Ő ugyanazokkal a jogokkal rendelkeznek, mint a sima adminisztrátorok azt leszámítva, hogy nekik nem lehet szerkeszteni a jogosultságát. Ő maga, ha át kívánják ruházni ezt a titulust valaki másra azt megteheti. Ennek következtében mindig csak pontosan egy ilyen titulussal rendelkező felhasználó létezik. Átruházni úgy lehetséges, hogy ha rákattint az adott felhasználóra és a megjelenő részletes adatokat tartalmazó felületen, jobb felül helyezkedik el a „Fő Adminisztrátor jog átadása” feliratú gomb, ami csak akkor jelenik meg ha a bejelentkezett felhasználó éppen a Fő Adminisztrátor.

#### Autók

A felső navigáló menü segítségével válthatunk a felhasználók és az autók listája között.

Ez a felület nagyon hasonló a felhasználók felületéhez. Az autók szintén egy listában jelennek meg, amiben leolvasható a rendszám, a tulajdonos neve, és hogy melyik parkolóban áll (vagy ha nem áll parkolóban). Lehet keresni rendszámra a jobb felül elhelyezkedő szövegmezővel. Ha valamelyik autó épp parkolóban áll akkor van lehetőség erről a felületről is végrehajtani a kiállást, a listaelemek végén szereplő „Kiállás” feliratú gombbal. (Ez persze csak abban az esetben látható a gomb, ha az adott autó parkol valahol.)

### **Napló**

<képernyőkép>

Ezt a menüpontot is csak az adminisztrátori jogkörrel rendelkező felhasználók látják és férnek hozzá. Itt tudják lekérdezni a rendszerben történő eseményeket, hogy ki mikor hova parkolt be és honnan parkolt ki. A bejegyzések időrendi sorrendben jelennek meg, és a legújabb időponttal rendelkező van legfelül. A bejegyzések listája feletti bevitelimezők segítségével lehetőség van időpontra, a bejegyzés eseményét kiváltó felhasználó nevére és a bejegyzés kategóriákra szűrni.

## Mobil alkalmazás

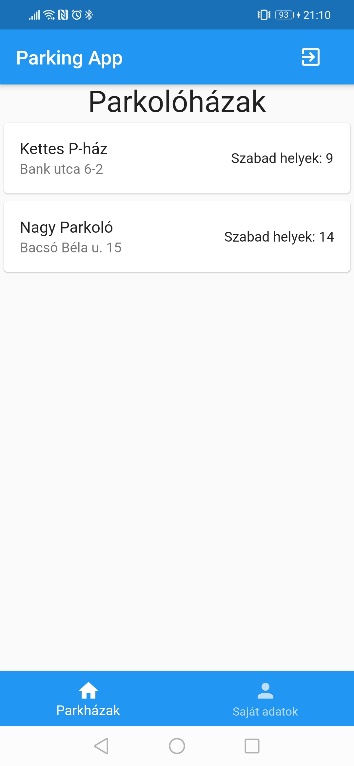
A mobil applikáció elsősorban az egyszerű felhasználók részére van. Az adminisztrációs funkciók nagy része nem elérhető benne. A mindenki számára elérhető funkciók hasonlóképpen működnek, mint a webes alkalmazás esetében.

### **Bejelentkezés**

Az alkalmazás megnyitásakor egy bejelentkező felület jelenik meg. Helyes emailcím és jelszó megadása után a „Bejelentkezés” gombra kattintva be lehet lépni az alkalmazásba. Be regisztrálni is van lehetőség a „Belépés” gomb alatti regisztráció feliratra kattintva. Ekkor a webes felülethez hasonlóan átalakul az űrlap és megjelennek plusz mezők, mint a „Vezetéknév” és „Keresztnév”, illetve a „Jelszó ismét”. A mezőket megfelelően kitöltve a „Regisztráció” gombra kattintva rögzülnek az adatok a rendszerben és vissza navigál az alkalmazás a bejelentkező űrlaphoz.

Az alkalmazás elmenti a belépési adatokat a mobil eszköz saját tárhelyén ezért ameddig ki nem jelentkezik a felhasználó addig automatikusan bejelentkezik a további indításoknál.

### **Parkolóházak listáját tartalmazó képernyő**

Belépés után a parkolóházak listája fogadja a felhasználót. Jobb felül a vissza nyíl ikonnal ellátott gombbal lehet kijelentkezni, aminek hatására az applikáció visszanavigál a bejelentkező képernyőre.

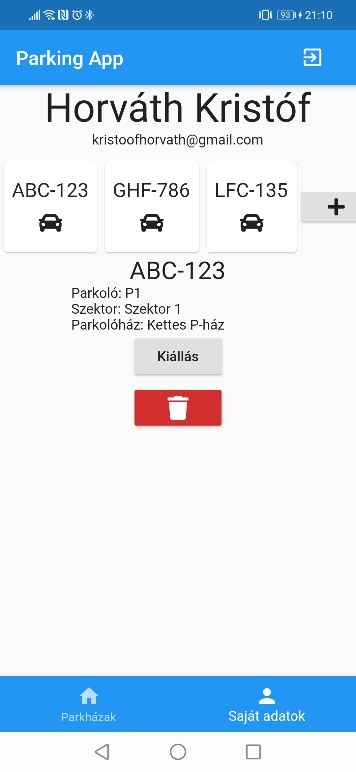
*5. ábra*

*Parkolóház lista*

Az egyes parkolóházakra nyomva megjelennek annak a parkolóháznak a szektorai. A szektorokra rányomva a webes felülethez hasonlóan legördülő panel jelenik meg ami tartalmazza a szektorban elhelyezkedő parkolókat. Itt a foglalt parkolók egy autó ikonnal vannak jelölve.

Ha rányomunk egy parkolóra akkor megjelenik a parkoló részletes információit tartalmazó felület. Itt lehetőség van kiparkolni a parkolóból, ha a felhasználó autója parkol bent éppen, ha üres a parkoló akkor be lehet állni az egyik birtokolt autóval. Ha tíz órán belül nem áll ki a parkolóból a felhasználó akkor kap egy emlékeztetőt a mobil készülékére. Ha adminisztrátor a bejelentkezett felhasználó akkor nem kell, hogy a saját autója álljon a parkolóban, bárkit kiállíthat.

### **Saját adatok képernyő**

A parkolóházak listájának felületéről átnavigálhatunk - a képernyő alján lévő navigációs menü segítségével - a Saját adatok felületre. Itt a felhasználó nevét és email címét lehet megtekinteni, valamint a birtokolt autóit. A „plusz” gomb segítésével lehet új autót felvenni amennyiben ötnél kevesebbel rendelkezik a felhasználó.

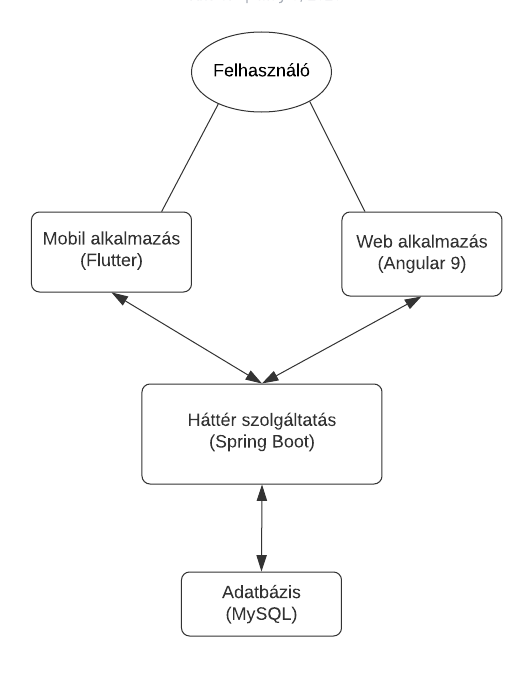
*6. ábra*

*Saját adatok képernyő*

Egy autóra rányomva megjelenik annak információi. Ha parkol valahol akkor a parkolóhelyet tartalmazó parkolóház és szektor lesz leolvasható, valamint a kiállást elősegítő gomb is megjelenik. Ha nem parkol sehol akkor csak a „nem foglal parkolóhelyet” felirat jelenik meg. Az autó törlésére is lehetőség van ezen a felületen a piros, szemetes ikonnal ellátott gomb segítségével.

# Fejlesztői dokumentáció

## Architektúra

A rendszer három fő komponensből tevődik össze.

*7. ábra  
Architektúra*

A megjelenítésért és a felhasználóval való kommunikációért két felület a felelős. Az egyik egy Android eszközökön futtatható mobil alkalmazás, a másik pedig egy böngészőkből elérhető webes felület. A vizuális felületek kiszolgálását a háttérszolgáltatás végzi. Itt történik az adatbázissal való kommunikáció és az adatok előállítása a felületek számára. A kliens alkalmazások és a háttérszolgáltatás http kérésekkel kommunikálnak egymással.

Az adatok tárolására MySQL[[1]](#footnote-1) relációs adatbázis szerver került kialakításra.

## Háttérszolgáltatás

A háttér szolgáltatás Java nyelven íródott és a Spring Boot[[2]](#footnote-2) keretrendszert használja. Ez lehetővé teszi, hogy a lefordított alkalmazás az egyetlen .jar kiterjesztésű állományból álljon, ami tartalmaz egy beépített Tomcat[[3]](#footnote-3) webszervert és ennek segítségével http kéréseket fogadjon.

### Maven

A fordítást és a fordított állományok előállítását a Maven[[4]](#footnote-4) projekt építő eszköz végzi, valamint a fejlesztéshez szükséges külső függőségek kezelését.

#### Függőségek

Ezeket a függőségeket a háttérszolgáltatás projektjének gyökérkönyvtárában található pom.xml-ben (Project Object Model) kell megadni, melynek következtében a következő futtatásnál minden hiányzó könyvtárat letölt egy központi tárhelyről. A Parkoló Kezelő rendszer háttérszolgáltatása a következő fontosabb függőségeket használja:

* spring-boot-starter-data-jpa
  + Java osztályoknak az adatbázis táblákba való transzformációért felelős könyvtárakat tartalmazza.
* spring-boot-starter-web
  + A legfontosabb Spring Boot szolgáltatásokat tartalmazza.
* spring-boot-starter-security
  + Autentikációhoz való könyvtárak.
* mysql-connector-java
  + MySQL adatbázis szerverhez való kapcsolódáshoz van rá szükség.

#### Futtatható állomány előállítása.

Ahhoz, hogy a Maven minden függőséget, amire a Spring Boot keretrendszernek szüksége van belepakoljon egy darab .jar kiterjesztésű állományba, szükség van, a *spring-boot-maven-plugin* nevezetű bővítményre. Ezt szintén a pom.xml-ben tudjuk a projekthez adni. A *configuration* tagek között lehet beállítani a main osztályt, illetve a futtatható állomány nevét.

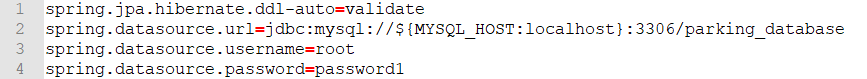
*8. ábra*

*Sprint-boot-maven-plugin*

Ezek után a projekt gyökérkönyvtárában kiadva a   
„mvn clean package”  
parancsot, előáll a /target mappa, amiben megtalálhatóak a .class file-ok a .jar kiterjesztésű futtatható állomány, amit a  
„java -jar ParkingApp.jar”   
futtathatunk.

### Adatbázis

#### Konfiguráció

Ahhoz, hogy a háttérszolgáltatás kommunikálni tudjon a MySQL szerverrel néhány konfigurációs adatot meg kell adnunk. Ezt Spring Boot alkalmazások esetében az *src/main/resources/* mappában lévő *application.properties* nevű fájlban lehet megtenni. Az adatbázisszerverhez való kapcsolódás érdekében itt meg kell adnunk az szerver elérhetőségét, felhasználónevet és a jelszót, amikkel hozzáférhetünk az adatbázishoz, ahogy azt a *7. ábra* mutatja. Az 1. sorban lévő beállítás azt mondja meg, hogy a háttérszolgáltatás minden indításkor ellenőrizze az adatbázis helyességét. Ezt fejlesztés alatt érdemes update-re átírni ugyanis ekkor, ha az osztály modellekben változás történik akkor a következő indításkor a Hibernate frissíti az adatbázis táblákat is. 

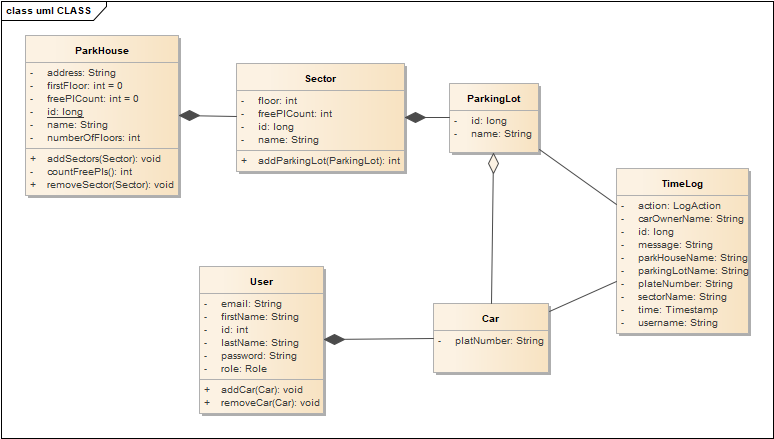
*9. ábra*

*Adatbázis konfiguráció*

#### Egyed-kapcsolatok

*10. ábra  
 Adatbázis egyedkapcsolat diagramja*

### Modellek

A háttérszolgáltatás hat modellt (saját típust) határoz meg amik a megvalósított funkciók alapjait képzik. Ezeknek a modelleknek megfelelően generálódnak le az adatbázis táblák. Minden új rekord beillesztésekor az elsődleges kulcs automatikusan generálódik. Az elsődleges kulcs a Car esetében a plateNumber (rendszám) attribútum, a többi modell esetében pedig az id. 

*11. ábra*

*Entitások osztálydiagramja*

User

A rendszerben szereplő felhasználókat a User osztály reprezentálja. Az osztály adattagjai a következők:

* id: Automatikusan generált egyedi azonosító.
* firstName: A felhasználó vezetékneve.
* lastName: A felhasználó keresztneve.
* email: A felhasználó emailcíme, érvényes emailnek, és egyedinek kell lennie.
* password: A felhasználó jelszava, amivel be tud jelentkezni. Legalább hat karakter hosszúnak kell lennie. Az adatbázisba kódolva kerül elmentésre. A kódolást a Spring Boot keretrendszer által biztosított *BCryptPasswordEncoder[[5]](#footnote-5)* osztály végzi.
* role: A felhasználó jogköre. Ez egy három értékből álló enumerációs típus. A következő értékekkel:
  + ROLE\_USER: Alap felhasználó
  + ROLE\_ADMIN: Adminisztrációs jogkörrel rendelkező felhasználó.
  + ROLE\_FIRST\_USER: A rendszer felállítása után elsőként beregisztráló felhasználó ezt a jogkört kapja. Az ROLE\_ADMIN jogkörhöz képest annyi plusz funkcióval bír, hogy ezt a felhasználót nem lehet megfosztani az adminisztrációs jogkörétől.
* ownedCars: A felhasználó tulajdonában lévő autók listája.

Tagfüggvények:

* addCar(Car): Hozzáadja a paraméterben kapott autót a felhasználó autóihoz.
* removeCar(Car): Eltávolítja a paraméterben megadott autót a felhasználó autói közül.

ParkHouse

A parkolóházak, amik tartalmazzák a parkolóhelyeket. A következő attribútumokat tartalmazza:

* id: egyedi azonosító szám.
* name: A Parkolóház neve, nem lehet üres.
* address: A Parkolóház címe. Ez sima szöveg, nincs semmilyen megszorítás, üresen is lehet hagyni.
* freePlCount: A parkolóházban jelenleg üresen álló parkolók száma, az értéke automatikusan kerül kiszámításra a countFreePls privát tagfüggvény segítségével.
* firstFloor: A parkolóház legalsó szintjének a száma. A parkoklóházban lévő szektorok mind egy-egy szinten helyezkendek el.
* numberOfFloors: A parkolóházban lévő szintek száma.
* secotrs: A prkolóházban fellelhető szektorok listálya.

Tagfüggvények:

* freePlcount() : Ez a függvény @Postload annotációval van ellátva. Ez gondoskodik róla, hogy mindig lefusson miután betöltöttük az adatbázisból az objektumot. A függvény összeszámolja a setors listában lévő szektorok freePlCount attribútumainak értékeit.
* addSectors(Sectors): A paraméterben kapott szektor listát hozzáfűzi a jelenlegi szektorokhoz.
* removeRestor(Sector): A paraméterben kapott szektort eltávolítja a szektorok közül.

#### Sector

A parkolóházakon belül szektorokba csoportosítva tárolódnak a parkolók. Egy parkolóházban több szektor is lehet és nincs elméleti felső korlátja a szektorok számának.

* id: Automatikusan előállított egyedi azonosító.
* name: A szektor megnevezése. Sima szöveg típus, nem lehet üres.
* floor: Az emeletnek a száma, amin elhelyezkedik a szektor a parkolóházon belül.
* freePlCount: A szektorban fellelhető üres parkolóhelyek száma. @Formula annotációval van ellátva az attribútum, ami azt teszi lehetővé, hogy SQL lekérdezéssel kerüljön kitöltésre. Itt összeszámolja a parking\_lots táblából azokat a rekordokat amelyiknek az id mezője megegyezik a szektor objektum id attribútumával (tehát az adott szektorhoz tartozik a rekord) és ahol a plate\_number mező null (tehát nem parkol benne épp autó).
* parkHouse: A parkolóház, aminek a része a szektor.
* parkingLots: A szektorban lévő parkolóhelyeknek a listája.

Tagfüggvények:

* addParkingLot(ParkingLot): Hozzáad egy új parkolót a szektorhoz.

#### ParkingLot

A parkolóhelyeket reprezentálja, amikbe be lehet parkolni autókkal.

* id: Automatikusan előállított egyedi azonosító
* name: A parkoló neve, sima szöveg, nem lehet üres.
* occupyingCar: Az autó amelyik épp a parkolóban áll. Ha null akkor nem áll bent autó.
* sector: A szektor amelyikben megtalálható a parkoló.

#### Car

Az autókat a Car nevezetű osztály reprezentálja.

Adattagok:

* plateNumber: Az autó rendszáma. Egyben elsődleges kulcsként is szolgál az adatbázisban
* owner: Az autót birtokló felhasználó.
* occupiedParkingLot: A foglalt parkoló. Ha épp nem foglal el parkolóhelyet, akkor ez az érték null.

#### TimeLog

A naplóban eltárolt bejegyzéseket a TimeLog osztály reprezentálja. Az adattagjai a következők:

* id: Automatikusan generált egyedi azonosító.
* time: Az időpont amikor az esemény történt, percre pontosan.
* action: A bejegyzett esemény típusa. Három féle eseményt ment el a napló. Felhasználó beregisztrálását, parkolóból való kiállást, illetve beállást. Ezeket a LogAction nevű enumerációs típus tartalmazza.
* userName: A bejegyzett eseményt kiváltó felhasználó neve (vezetéknév és keresztnév szóközzel elválasztva).
* message: A bejegyzet esemény szöveges reprezentációja. Ezt a szöveget látják az adminisztrátorok a naplót megjelenítő felületen.
* plateNumber: Amennyiben parkolás történt akkor a parkoló autó rendszáma.
* carOwnerName: Amennyiben parkolás történt a parkoló autó tulajdonosának neve (vezeték és keresztnév szóközzel elválasztva)
* parkingLotName: A parkoló neve, ahol a ki- beállás történt.
* sectorName: Annak a szektornak a neve, ahol az a parkoló van, ahova a be- kiállás történt.
* parkHouseName: Annak a parkolóháznak a neve, ahova a be- kiparkolás történt.

#### LofFilter

Az alap modelleken kívül létezik még egy LogFilter osztály is. Ez a naplóban való keresés funkció miatt létezik. Az adat tagjai a LogFilter osztálynak azok a paraméterek, amik mentén lehet szűrni az adatbázisban eltárolt TimeLog objektumoknak megfelelő rekordokat. Ezek az adattagok a következők:

* userName: Az elnaplózott eseményt kiváltó felhasználó neve.
* action: Az a bejegyzés típus amire szűrni szeretne a felhasználó.
* startTime: Annak az időintervallumnak a kezdeti ideje amire szűrni szeretnénk.
* endTime: Annak az időintervallumnak a vége amire szűrni szeretnénk.

### Repositorik

# Összefoglalás

Összefoglalás.

# További fejlesztések

Jobb program készítése.

1. MySQL: <https://www.mysql.com/> [↑](#footnote-ref-1)
2. Spring Boot: <https://spring.io/projects/spring-boot> [↑](#footnote-ref-2)
3. Apache Tomcat: <http://tomcat.apache.org/> [↑](#footnote-ref-3)
4. Apache Maven: <http://maven.apache.org/> [↑](#footnote-ref-4)
5. ByCryptPasswordEncoder: <https://docs.spring.io/spring-security/site/docs/4.2.14.RELEASE/apidocs/org/springframework/security/crypto/bcrypt/BCryptPasswordEncoder.html> [↑](#footnote-ref-5)