



EÖTVÖS LORÁND TUDOMÁNYEGYETEM  
INFORMATIKAI KAR  
PROGRAMOZÁSELMÉLET ÉS SZOFTVERTECHNOLÓGIAI  
TANSZÉK

## Budapesti Busz Infrastruktúra Szimuláció

*Témavezető:*

Pintér Balázs  
egyetemi docens

*Szerző:*

Nagy Ambrus Szemere  
programtervező informatikus BsC

*Budapest, 2025*

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>5</b>
<b>2. Felhasználói dokumentáció</b>	<b>6</b>
2.1 Az alkalmazás funkciói.....	7
2.1.1 Felhasználók kezelése.....	7
2.1.2 Buszok vásárlása.....	7
2.1.3 Buszok értékesítése.....	9
2.1.4 Buszok kezelése.....	10
2.1.5 Garázs kezelése.....	10
2.1.6 Műszaki hibák kezelése.....	11
2.1.7 Menetrendek kezelése.....	13
2.1.8 Vonalak áttekintése.....	14
2.1.9 A menü.....	15
<b>3. Fejlesztői dokumentáció</b>	<b>16</b>
3.1 A rendszer felállítása.....	16
3.1.1 Előzetes követelmények.....	16
3.1.2 Az alkalmazás felállítása.....	17
3.2 Követelményelemzés.....	18
3.2.1 A felhasználói felület terve.....	22
3.2.2 A menü.....	22
3.2.3 A főoldal.....	23
3.2.4 A piac.....	24
3.2.5 A műszaki hibák.....	25
3.2.6 A vonalak.....	26
3.2.7 A garázsok.....	27
3.2.8 A menetrendek.....	27

3.3 Az alkalmazás rétegei.....	29
3.4 Az adatbázis réteg.....	31
3.4.1 A buszokat tartalmazó tábla.....	32
3.4.2 A garázsokat tartalmazó tábla.....	32
3.4.3 A vonalakat tartalmazó tábla.....	32
3.4.4 A műszaki hibákat tartalmazó tábla.....	33
3.4.5 A menetrendeket tartalmazó tábla.....	33
3.4.6 A felhasználókat tartalmazó tábla.....	33
3.4.7 A piaci hirdetéseket tartalmazó tábla.....	34
3.4.8 Segédtablák.....	34
3.5 A backend réteg moduljai.....	35
3.5.1 A buszokért felelős modul.....	35
3.5.2 A garázsokért felelős modul.....	37
3.5.3 A vonalakért felelős modul.....	38
3.5.4 A műszaki hibákért felelős modul.....	39
3.5.5 A piacért felelős modul.....	41
3.5.6 A menetrendekért felelős modul.....	44
3.6 Tesztelés.....	48
3.6.1 A busz modul tesztelése.....	48
3.6.2 A garázs modul tesztelése.....	48
3.6.3 A műszaki hiba modul tesztelése.....	48
3.6.4 A vonal modul tesztelése.....	49
3.6.5 A piac tesztelése.....	49
3.6.6 A menetrendek tesztelése.....	50
3.6.7 A felhasználók tesztelése.....	51

## *Tartalomjegyzék*

---

3.6.8 A tesztelés lebonyolítása.....	51
<b>4. Összegzés</b>	<b>52</b>
<b>Köszönetnyilvánítás</b>	<b>53</b>
<b>Táblázatjegyzék</b>	<b>54</b>
<b>Ábrajegyzék</b>	<b>55</b>

# 1. fejezet

## Bevezetés

A projektem a budapesti tömegközlekedés, pontosabban a buszok, az aktivitásuk, és a közösségi közlekedésben részt vett szerepük köré fog épülni. Az alkalmazás kiegészítve egy webes felülettel lehetőséget biztosít a felhasználónak betekintést nyerni a buszok világába, és a vonalak menedzselésébe, a buszok elosztásába, karbantartásába. Ezen felül a menetrendek megszervezésére is lesz lehetőség, amihez az alkalmazás egy segédeszközt is biztosít.

Ahogy az megfigyelhető, Budapest belvárosában a transzportáció igen kevés kötött pályát tartalmaz. Igaz ugyan, hogy vannak oszlopos villamosvonalak, viszont átfogó, nagy terhelésű vágányvonalak egyáltalán nincsenek kiépítve. A vonatok és HÉV-ek az agglomerációs településeket, illetve néhány külvárosi területet fedeznek, de egyikük sem szeli keresztül a várost. Erre a feladatra a metrók, a villamosok, illetve az autóbuszok vállalkoznak. Metróból mindössze 4 vonal lett kiépítve, a villamosjáratok száma pedig nagyjából 40 vonal. Ehhez képest a buszvonalak száma megközelíti a 250-et.

Ezekből az adatokból, illetve tekintve a buszsávok gyakoriságát, és azt, hogy az éjszakai közlekedési hálózatot is nagyjából csak buszvonalak adják ki, egyértelmű, hogy az autóbuszok azok, akik a hátukon viszik a közösségi közlekedés terhének túlnyomó részét.

Ez a projekt a budapesti buszok felé érdeklődést mutató személyek számára készült, akik szeretik megfigyelni, hogy milyen buszokon utaznak, és érdeklődést mutatnak a főváros tömegközlekedési gerincét ellátó buszflotta felépítése felé. Akik szeretnék átérezni, milyen egy garázst menedzselni, vagy egy vonal szolgáltatását ellátni, vagy egyszerűen csak gyűjteni olyan buszokat, amik fontosak nekik, azoknak ez a webalkalmazás egy teret biztosít, hogy kiélhessék ezen igényeiket.

## 2. fejezet

# Felhasználói dokumentáció

Ebben a szekcióban az alkalmazás használatára vonatkozó felhasználói élmény és eseménykezelés található. Bemutatásra kerülnek az egyes oldalak funkciói, a ki-be jelentkezés, illetve az értesítő- és hibaüzenetek.

Az alkalmazás lehetőséget biztosít a felhasználóinak járműveik beszerzésére és értékesítésére, azoknak tárolására, műszaki hibáinak nyilvántartására, menetrendek feltöltésére és azoknak üzembe helyezésére.

## 2.1 Az alkalmazás funkciói

### 2.1.1 Felhasználók kezelése

Az alkalmazás felhasználókat kezel, minden felhasználónak teret nyújtva saját tömegközlekedési infrastruktúra kiépítésére, járművei kezelésére és menetrendjei szerkesztésére. A regisztráció és bejelentkezés az első lépés az alkalmazás használatához.

The login screen features a light blue background with a rounded rectangle border. At the top center is a bus icon. Below it, the text 'WELCOME BACK' is displayed in bold, followed by 'Sign in to continue' in a smaller font. There are two input fields: 'USERNAME' with the placeholder 'Your username' and 'PASSWORD' with the placeholder 'Your password'. Below the password field is a blue 'LOGIN' button. At the bottom is a light blue 'SWITCH TO REGISTER' button.

(a) Bejelentkezés

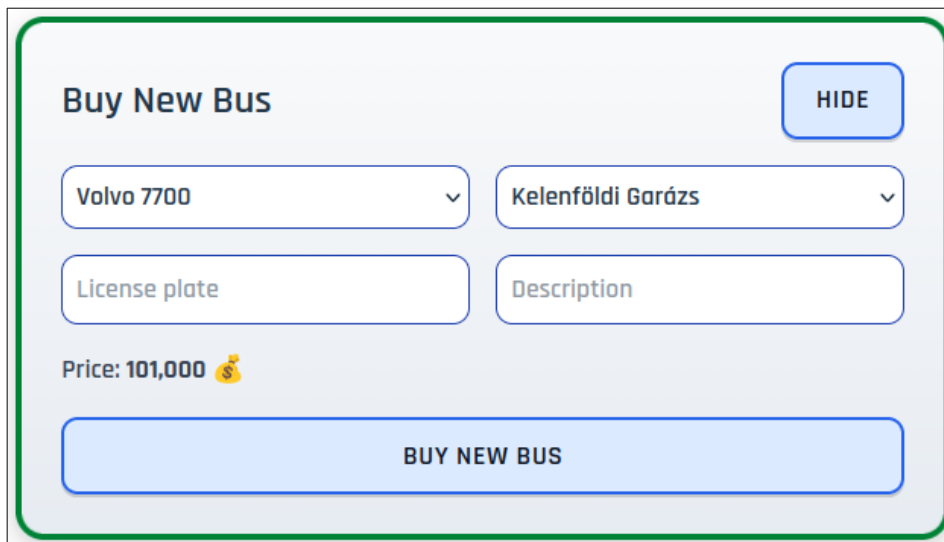
The registration screen has a similar layout to the login screen. It features a bus icon at the top, followed by 'CREATE ACCOUNT' in bold and 'Register to get started' in a smaller font. There are two input fields: 'USERNAME' with the placeholder 'Your username' and 'PASSWORD' with the placeholder 'Your password'. Below the password field is a blue 'REGISTER' button. At the bottom is a light blue 'SWITCH TO LOGIN' button.

(b) Regisztráció

2.1. ábra. Bejelentkezési ablak

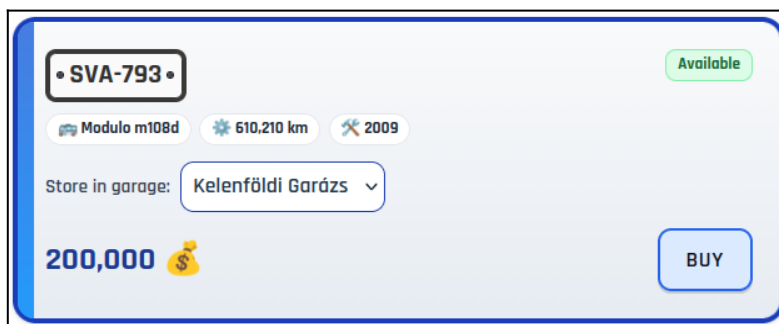
### 2.1.2 Buszok vásárlása

A felhasználónak lehetősége van vásárolni használt, illetve új buszokat a piacon. Használt buszok esetén más felhasználók ajánlatai közül lehet válogatni, ha pedig új járművet szeretne beszerezni, ki tudja választani annak típusát, meg tudja adni a rendszámát, illetve a leírását. Mind a két esetben el kell döntenie, hogy melyik garázsban szeretné tárolni azt.



The screenshot shows a 'Buy New Bus' window. At the top left is the title 'Buy New Bus' and at the top right is a 'HIDE' button. Below the title are two dropdown menus: 'Volvo 7700' and 'Kelenföldi Garázs'. Under these are two text input fields labeled 'License plate' and 'Description'. Below the inputs, the price is displayed as 'Price: 101,000' with a money bag icon. At the bottom is a large blue button labeled 'BUY NEW BUS'.

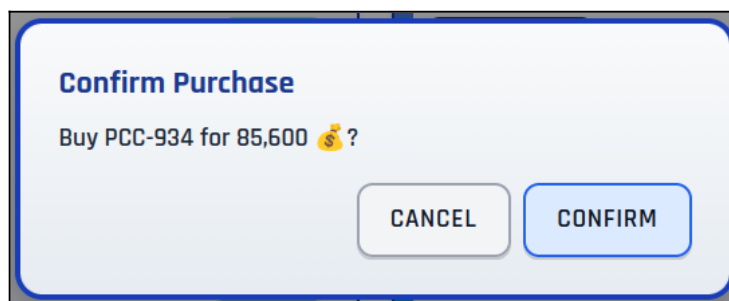
2.2. ábra. Új busz vásárlása



The screenshot shows a 'Used Bus' purchase interface. At the top left is a license plate 'SVA-793' in a black box. At the top right is a green 'Available' button. Below these are three icons with text: a bus icon for 'Modulo m108d', a gear icon for '610,210 km', and a wrench icon for '2009'. Below these is a 'Store in garage:' label followed by a dropdown menu showing 'Kelenföldi Garázs'. At the bottom left, the price is '200,000' with a money bag icon. At the bottom right is a blue 'BUY' button.

2.3. ábra. Használt busz vásárlása

A rendszer minden használt, avagy új vásárlás előtt megerősítést kér a felhasználótól, a véletlenül kezdeményezett vásárlások elkerülése végett.



The screenshot shows a 'Confirm Purchase' dialog box. The title is 'Confirm Purchase'. Below the title, it says 'Buy PCC-934 for 85,600' with a money bag icon and a question mark. At the bottom are two buttons: 'CANCEL' and 'CONFIRM'.

2.4. ábra. Vásárlás megerősítése

Amennyiben a felhasználó egy már létező rendszámmal, vagy a vásárláshoz nem elegendő pénzzel próbál járművet beszerezni, a rendszer értesíti a hibás beszerzéshez kapcsolatos hibákról.





(a) Foglalt rendszám - hibaüzenet

(b) Elégtelen valuta - hibaüzenet

2.5. ábra. Vásárlással kapcsolatos hibaüzenetek

### 2.1.3 Buszok értékesítése

A felhasználónak lehetősége van piacra bocsájtani a járműveit. Ilyenkor megadhat egy árat, aminek fejében értékesítené a szóban forgó tételt, ez után az felkerül a piacra. Ekkor még az adott jármű teljes mértékben funkcionális, azonban amint egy másik felhasználó megveszi azt, kikerül a forgalomból. Ha a tulajdonos úgy dönt, hogy mégsem szeretne megválni egy bizonyos busztól, az ajánlatot le tudja mondani.

A 'Create Listing' form with a light blue background and a blue border. At the top left is the title 'Create Listing' in bold blue text. At the top right is a 'HIDE' button in a blue box. Below the title is a dropdown menu labeled 'Select Your Bus' with a downward arrow. Below that is a text input field labeled 'Listing price' with a small up/down arrow on the right. At the bottom is a large blue button labeled 'LIST BUS'.

2.6. ábra. Busz feltétele a piacra

Amennyiben a felhasználó egy már a piacon lévő buszhoz szeretne újabb értékesítési hirdetést létrehozni, a rendszer tudatja vele, hogy az adott jármű már fent van a piacon. Ilyenkor meg kell szüntetnie az aktív hirdetését, mielőtt egy újat tudna létrehozni.

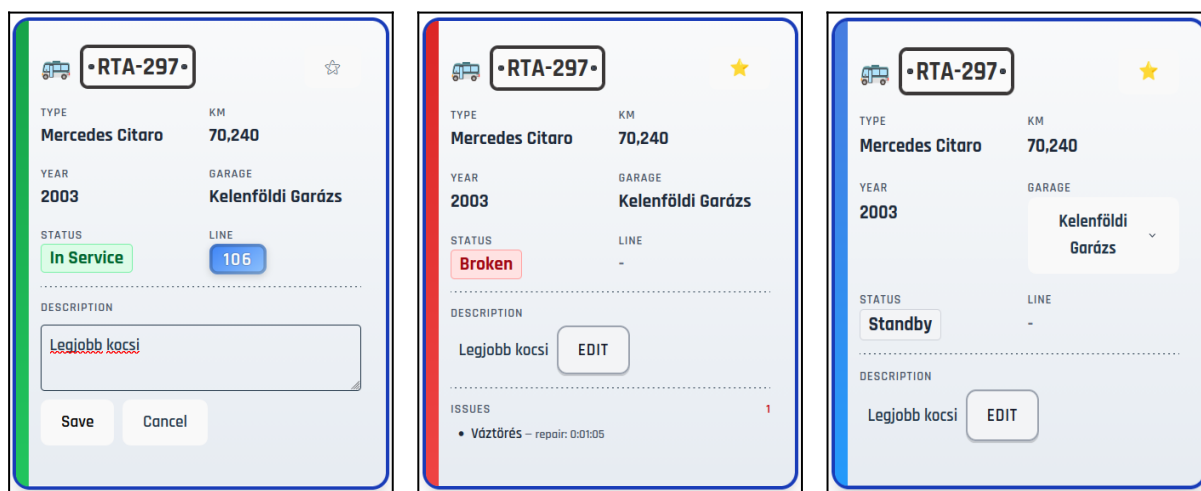
A 'Your listing' card with a light blue background and a blue border. At the top left is a license plate '•SVA-793•' in a black box. At the top right is a yellow button labeled 'Your listing'. Below the license plate are three icons: a car icon labeled 'Modulo m108d', a gear icon labeled '610,210 km', and a wrench icon labeled '2009'. Below these is a green button labeled 'No issues'. Below that is the price '200,000' in large blue text with a yellow money bag icon. At the bottom is a red button labeled 'CANCEL'.

2.7. ábra. Saját hirdetés lemondása

### 2.1.4 Buszok kezelése

A felhasználó számára az az alkalmazás fő oldala egy összefogó áttekintést biztosít a fentartott járművekről, az alábbi funkciókat ellátva.

- Adatok megjelenítése: rendszám, típus, kilométerállás, évjárat, aktív vonal (ha van), műszaki hiba (ha van).
- Leírás illetve garázs megjelenítése és módosítása.
- Státusz kiemelt jelzése.
- Kedvencek nyilvántartása: Itt a felhasználó képes megjelölni kedvenceket, illetve szűrőt alkalmazni a megjelenítésükre.



(a) Aktív, leírás módosítása

(b) Hibás, kedvenc

(c) Passzív, garázs módosítható

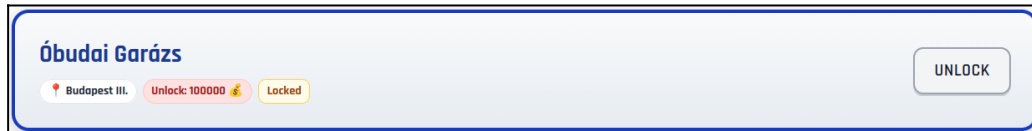
2.8. ábra. Buszok kezelése

### 2.1.5 Garázs kezelése

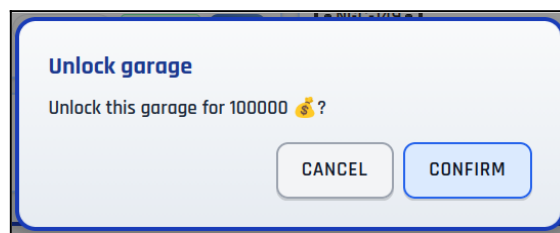
A garázsok nyilvántartására egy külön oldalt ajánl az alkalmazás, ami egy konszolidáltabb nézőpontból közelíti meg a járműflotta áttekintését. Itt is megtekinthetők a buszok bizonyos adatai, garázsonkénti bontásban, illetve megvásárolhatóak azok a telephelyek, amik még nem elérhetőek. Amennyiben a felhasználónak egyetlen garázshoz sincs hozzáférése, az elsőt díjmentesen feloldhatja. A garázs feloldását egy megerősítő üzenet előzi meg.



2.9. ábra. Feloldott garázs állapota

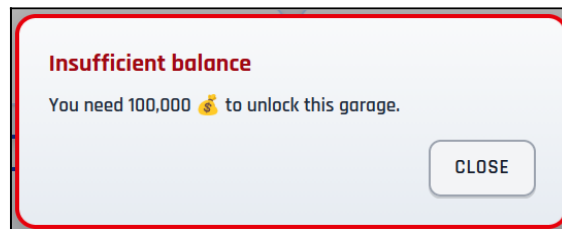


2.10. ábra. Feloldatlan garázs



2.11. ábra. Garázs megvásárlása

Amennyiben a felhasználó úgy próbál garázst vásárolni, hogy nem rendelkezik elegendő valutával, a rendszer értesíti a felmerülő hibáról.



2.12. ábra. Garázs sikertelen vásárlása

### 2.1.6 Műszaki hibák kezelése

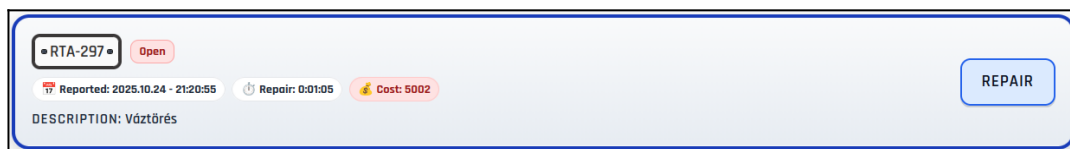
A felmerülő problémák jelentésére külön felület van, ahol ezeket rögzíteni esedékes. Ilyenkor az adott busz kikerül a forgalomból, és addig szervízben van, amíg fennálló hibával rendelkezik. Ezeknek javítására, illetve a javítás megszakítására is ezen a felületen van mód.



A form for reporting a technical issue. It contains the following fields and elements:

- A dropdown menu with the value "RTA-297".
- A text input field containing "2025.10.24 - 21:20:55".
- A text input field containing "00:01:05".
- A dropdown menu with the value "5002".
- A text input field containing "Váztörés".
- A blue button labeled "CONFIRM".

2.13. ábra. Műszaki hiba jelentése

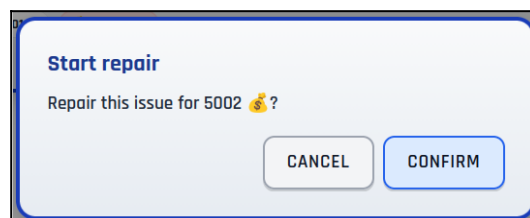


A horizontal status bar for the technical issue. It contains the following elements:

- A dropdown menu with the value "RTA-297".
- A red button labeled "Open".
- A status icon (bug) and text "Reported: 2025.10.24 - 21:20:55".
- A status icon (clock) and text "Repair: 0:01:05".
- A status icon (price tag) and text "Cost: 5002".
- A blue button labeled "REPAIR".
- Text "DESCRIPTION: Váztörés".

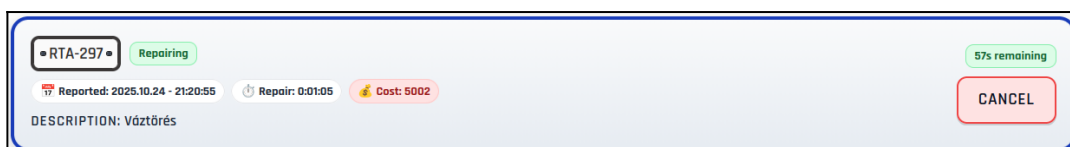
2.14. ábra. Műszaki hiba

Amennyiben a felhasználó úgy dönt, hogy meg szeretne javítani egy fennálló problémát, a rendszer egy megerősítő üzenetet követően elkezd a javítási folyamatot.



A confirmation dialog box titled "Start repair". It contains the text "Repair this issue for 5002 🙌?". At the bottom, there are two buttons: "CANCEL" and "CONFIRM".

2.15. ábra. Műszaki hiba javításának megerősítése

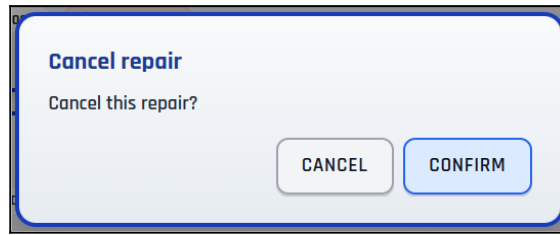


A horizontal status bar for the technical issue during repair. It contains the following elements:

- A dropdown menu with the value "RTA-297".
- A green button labeled "Repairing".
- A status icon (bug) and text "Reported: 2025.10.24 - 21:20:55".
- A status icon (clock) and text "Repair: 0:01:05".
- A status icon (price tag) and text "Cost: 5002".
- A green progress bar labeled "57s remaining".
- A red button labeled "CANCEL".
- Text "DESCRIPTION: Váztörés".

2.16. ábra. Műszaki hiba javítás alatt.

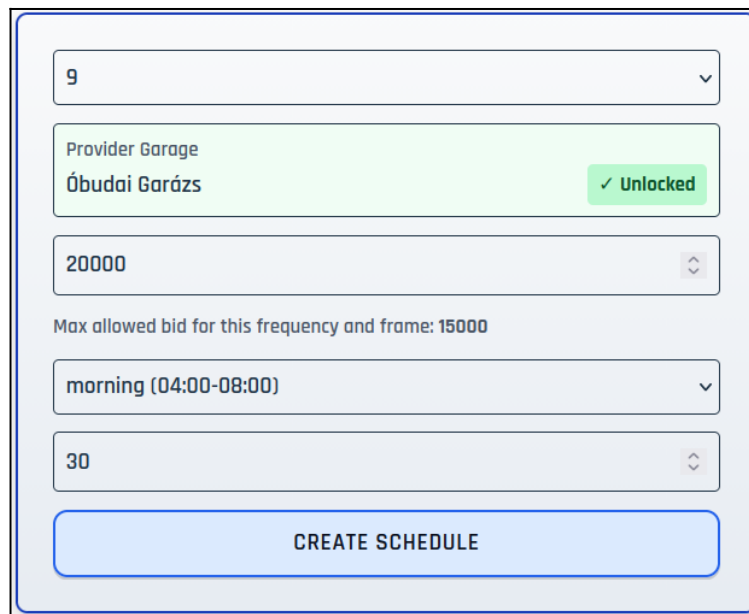
Amennyiben a felhasználó úgy dönt, hogy megszakítana egy aktív javítási folyamatot, a rendszer egy megerősítő üzenetet követően megszakítja azt. Ilyenkor a tulajdonos a javítási költségtől eszik.



2.17. ábra. Javítás megszakítása

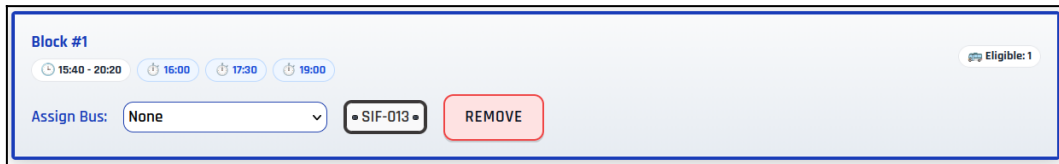
### 2.1.7 Menetrendek kezelése

A menetrendek kezelése talán az alkalmazás legfontosabb komponense. A felhasználók itt versengenek egy vonal bizonyos időközön belüli üzemeltetésére. Egy menetrend létrehozásakor meg kell adni a vonalat, az üzemeltetési ajánlat árát, az időkeretet, illetve az indulások gyakoriságát. Az üzemeltetési ár felső határa arányos a gyakorisággal, illetve függ a napszaktól, a sűrűbb ellátást igénylő órák jobban fizetnek. Minden vonal egy garázshoz tartozik, csak azon vonalakhoz lehetséges menetrendet készíteni, amikhez a szükséges garázs elérhető.



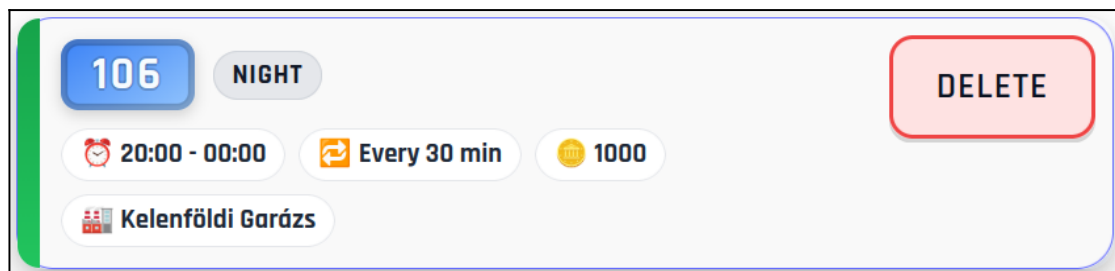
2.18. ábra. Menetrend létrehozása

Miután a paraméterek megadásra kerültek, a rendszer automatikusan létrehoz forgalmi számokat, mindegyiken feltüntetve az indulási időpontokat. A forgalmi számokhoz a felhasználó hozzá tudja rendelni a megfelelő garázsban tárolt és tartalékon lévő járműveit.



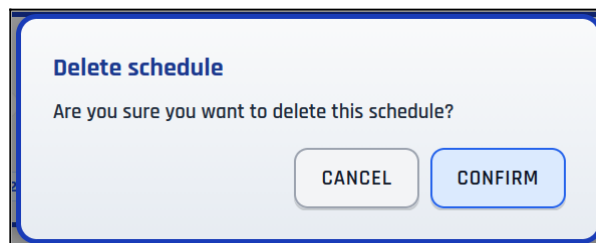
2.19. ábra. Busz hozzárendelése a menetrendhez

Egy menetrend akkor lép aktív stádiumba, ha az adott vonalon és időkeretben a leggyakoribb ellátást, megegyező gyakoriság esetén pedig a legolcsóbb szolgáltatást biztosítja, illetve minden forgalmi szám fel van töltve arra alkalmas járművekkel. Az aktuális időkeretben aktív menetrendek az üzemeltetési árral arányosan, időközönként kerülnek kifizetésre.



2.20. ábra. Menetrend aktív státuszban

Amennyiben a felhasználó törölni kíván egy menetrendet, a rendszer egy figyelmeztető üzenetet küld a döntés biztosítására. Ezt követően, miután a menetrend törlésre került, minden ahhoz rendelt busz automatikusan felszabadul.



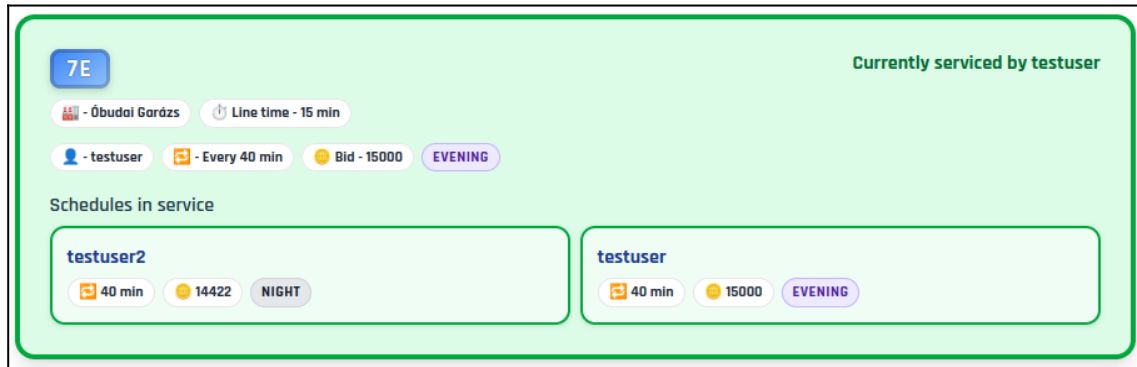
2.21. ábra. Menetrend törlése

### 2.1.8 Vonalak áttekintése

Az átláthatóság érdekét szolgálva egy külön oldal található a vonalak állapotának vizsgálatára. Itt láthatóak a vonalak, és fontos adataik, illetve, hogy jelenleg üzemeltetés alatt állnak-e, továbbá fel vannak tüntetve az adott időszakokban az érintett vonalra aktív menetrendek.



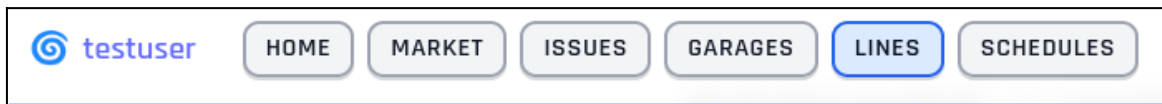
2.22. ábra. Vonal ellátás nélkül



2.23. ábra. vonal ellátva

### 2.1.9 A menü

A navigálást biztosító menü az oldal tetején helyezkedik el egy sávban, ahol különböző gombok látják el a hozzájuk tartozó oldal elérését. Az oldal, amin a felhasználó tartózkodik, kéken meg van jelölve.



2.24. ábra. Menü

A navigáción felül a menüsor jobb oldalán látható a felhasználó egyenlege, illetve ugyanitt található a kilépés gomb, amire rákattintva a felhasználó visszakerül a bejelentkezési panelre.



2.25. ábra. az egyenleg és kijelentkezés

## 3. fejezet

# Fejlesztői dokumentáció

Ez a szekció az alkalmazás tervezési elemeit, a megvalósításhoz szükséges koncepciókat, illetve egyéb szakmai részleteket tartalmaz. Szó kerül benne a felépítés rétegeitől kezdve a tesztelésig minden fontos tudnivalóról.

## 3.1 A rendszer felállítása

### 3.1.1 Előzetes követelmények

Ahhoz, hogy fel lehessen állítani az alkalmazást, egy pár előkövetelményt biztosítani kell. A környezet létrehozása előtt a következő csomagok installációja elengedhetetlen:

- python
- Node.js (version 18, vagy újabb)
- npm
- pip

Miután ezek az installációk megtörténtek, és a kódbázis el lett helyezve egy projekt mappába, megkezdhető a Flask alkalmazás és a felhasználói felület konfigurációja.

Az működéshez még szükség van az adatbázis létrehozására is, ezt az adatbázis réteget bemutató részleg alapján érdemes elkészíteni, a kapcsolódáshoz szükséges adatokat pedig a `flask_app/config.py` fileban tárolni.



#### 3.1.2 Az alkalmazás felállítása

Az előkészületek teljesítése után megkezdhető a backend réteg elindítása. Ezt egy python virtuális környezettel tanácsos kezdeni, ahol a szükséges könyvtárak lesznek elhelyezve. A folyamat a projekt mappából kiindulva a kindulva az alábbi parancsokkal kivitelezhető:

- `cd flask_app`
- `python3 -m venv .venv`
- `source .venv/bin/activate`
- `pip install Flask`
- `pip install Flask-Cors`
- `pip install Flask-Bcrypt`
- `pip install APScheduler`
- `pip install psycpg2-binary`
- `python app.py`

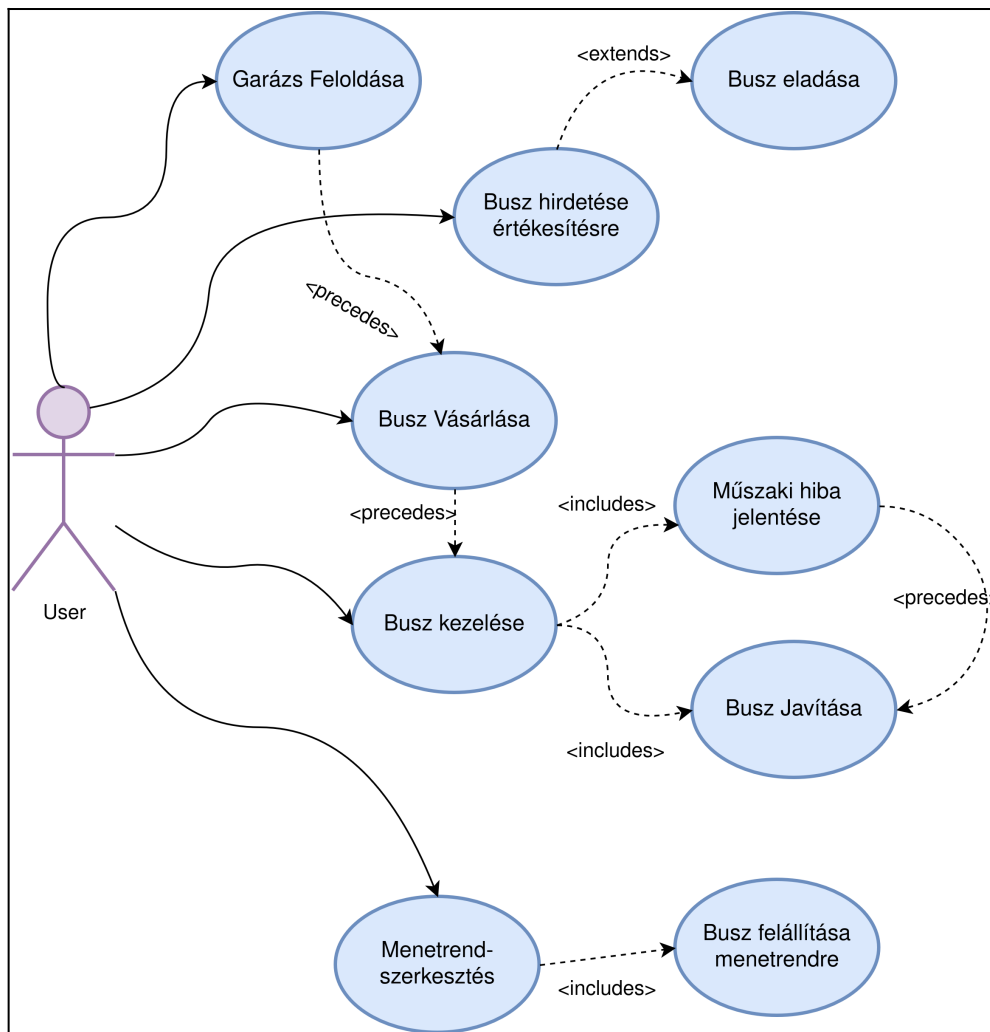
A felhasználói felület konfigurációja pedig a következő parancsokkal elérhető:

- `cd frontend`
- `npm install`
- `npm run dev`

## 3.2 Követelményelemzés

Ebben a szekcióban található a projekt kivitelezésének szakmai elvárása. Definiálásra kerül a felhasználó által végrehajtható tevékenységeinek halmaza, az egyes funkciók által végrehajtott akciók elvárt működése, illetve a felhasználói felület tervezete.

Az felhasználók hatáskörének nagyvonalú funkcionálisának szemléltetésére a 3.1. diagram szolgál.



3.1. ábra. Use Case Diagram

A program által kielégítendő működési és funkcionális elvek részletezését given-when-then hármassok jellemzik. Ezek a kategorizációk tárgyalják, hogy egy adott helyzetben bizonyos akciók milyen következményt kell, hogy eredményezzenek. A given oszlopban található, hogy mi a fennálló körülmény, a when oszlop tartalmazza a végrehajtandó akciót, a then pedig az akció eredményét a fenntartást tekintetbe véve.

Busz vásárlása		
Given	When	Then
A felhasználó új buszt vásárolna	Van elegendő egyenlege és helyes paramétereket ad meg	A busz a felhasználó tulajdonába kerül a busz árának értékében
A felhasználó új buszt vásárolna	A felhasználónak nincs elegendő egyenlege	A vásárlás sikertelen
A felhasználó új buszt vásárolna	A felhasználó foglalt rendszámot ad meg	A vásárlás sikertelen
A felhasználó új buszt vásárolna	A felhasználónak nincs feloldott garázsa	A vásárlás nem elérhető

3.1. táblázat. A busz vásárlásának követelményleírása

Busz eladása		
Given	When	Then
Az eladásra bocsájtott buszra nem vonatkozik aktív hirdetés	A felhasználó meghirdeti a buszt értékesítésre	A busz felkerül a piacra
Az eladásra bocsájtott buszra vonatkozik aktív hirdetés	A felhasználó meghirdeti a buszt értékesítésre	A hirdetés sikertelen
A felhasználónak van egy aktív hirdetése	A felhasználó lemondja a hirdetést	A hirdetés visszavonásra kerül
A felhasználónak egy piacon hirdetett busza nem áll szolgálatban	A buszt megvásárolja egy másik felhasználó	A az adás-vétel a busz árának fejében teljesül
A felhasználónak egy piacon hirdetett busza szolgálatban áll	A buszt megvásárolja egy másik felhasználó	A az adás-vétel a busz árának fejében teljesül, és a busz kikerül az értékesítő által üzemeltetett szolgáltatásból
A felhasználónak egy piacon hirdetett busza műszaki hibás	A buszt megvásárolja egy másik felhasználó	A az adás-vétel a busz árának fejében teljesül, a műszaki hiba miatt a jármű a forgalomból kivonva kerül át az új tulajdonoshoz

3.2. táblázat. A busz eladásának követelményleírása

Busz kezelése		
Given	When	Then
A felhasználónak van egy busz a tulajdonában és	A felhasználó áthelyezné a járművet egy másik általa	A busz átkerül az új garázsba

Busz kezelése		
nincs szolgálatban, vagy szervízben	feloldott garázsba	
A felhasználónak van egy busz a tulajdonában és szolgálatban, vagy szervízben van	A felhasználó áthelyezné a járművet egy másik általa feloldott garázsba	Az áthelyezés nem elérhető
A felhasználónak van egy busz a tulajdonában	A felhasználó megváltoztatja a busz leírását	A leírás megváltozik
A felhasználónak van egy busz a tulajdonában	A felhasználó ki, vagy be kapcsolja a “kedvenc” jelzőt az adott járműre	A jármű a kedvencek közé, vagy ki kerül

3.3. táblázat. A busz kezelésének követelményleírása

Műszaki hiba kezelése		
Given	When	Then
A felhasználónak van egy tartalékos busz a tulajdonában	A felhasználó jelent egy műszaki hibát az adott buszra	A műszaki hiba rögzítésre kerül és a jármű hibás állapotot vesz fel
A felhasználónak van egy szolgálatban álló busz a tulajdonában	A felhasználó jelent egy műszaki hibát az adott buszra	A műszaki hiba rögzítésre kerül, a jármű hibás állapotot vesz fel és kikerül a forgalomból
A felhasználónak van egy hibás busz a tulajdonában	A felhasználó jelent egy műszaki hibát az adott buszra	A műszaki hiba rögzítésre kerül
A felhasználónak van egy hibás busz a tulajdonában	A felhasználónak van elegendő egyenlege és megjavítja az adott problémát	A javítási ár fejében elkezdődik a helyreállítás, végeztével a jármű kikerül a szervízből
A felhasználónak van egy több hibával rendelkező busz a tulajdonában	A felhasználónak van elegendő egyenlege és megjavítja az adott problémát	A javítási ár fejében elkezdődik a helyreállítás, végeztével az érintett probléma megszűnik, de a busz szervízben marad
A felhasználónak van egy hibás busz a tulajdonában	A felhasználónak nincs egyenlege és megjavítaná az adott problémát	A javítás megkezdése sikertelen

3.4. táblázat. A műszaki hibák kezelésének követelményleírása

Garázs kezelése		
Given	When	Then
A felhasználónak nincs feloldott garázs	A felhasználó felold egy kiválasztott garázst	A garázs feloldása díjmentesen teljesül
A felhasználónak van feloldott garázs	A felhasználó felold egy kiválasztott garázst és van elegendő egyenlege	A garázs feloldása a feloldási díj fejében teljesül
A felhasználónak van feloldott garázs	A felhasználó felold egy kiválasztott garázst és nincs elegendő egyenlege	A garázs feloldása sikertelen

3.5. táblázat. A garázs kezelésének követelményleírása

Menetrend-szerkesztés		
Given	When	Then
A felhasználónak fel van oldva egy adott garázs	A felhasználó generál egy menetrendet egy olyan vonalhoz, ami az adott garázshoz tartozik	A menetrend létrejön
A felhasználónak nincs feloldva egy adott garázs	A felhasználó generálna egy menetrendet egy olyan vonalhoz, ami az adott garázshoz tartozik	A menetrend nem hozható létre
A felhasználónak van egy menetrendje	A felhasználó hozzárendel egy járművet az adott garázból a menetrendhez	A jármű felkerül a menetrendre
A felhasználónak van egy menetrendje, ami az adott vonalon és időkeretben a leggyakoribb forgalmú, egyező ütemterv esetében pedig a legolcsóbb szolgáltatás	A felhasználó a menetrend teljesítéséhez megkívánt összes buszt hozzárendeli	A menetrend aktív stádiumba kerül
A felhasználónak van egy menetrendje	A felhasználó törli az adott menetrendet	A menetrend törlésre kerül, a hozzárendelt buszok pedig tartalékra állnak
A felhasználónak van egy aktív menetrendje	Az egyik hozzárendelt busz valamilyen okból kiáll a forgalomból	A menetrend deaktiválódik

3.6. táblázat. A menetrend-szerkesztés követelményleírása

#### 3.2.1 A felhasználói felület terve

A felhasználói felület tervezete egy infrastruktúra menedzseléséhez kellően letisztult, az adatok megjelenítésére és a navigáció, illetve a különböző akciók megvalósítására főleg kártyákat és gombokat használó weboldal.

A megvalósításhoz ebben a környezetben a react kézenfekvő elemei állnak szolgálatra, mind a megjelenítésben, mind a tevékenységek kivitelezésében és végül, de nem utolsó sorban, a modulokkal való kommunikációban, HTTP requesteken keresztül.

Az alábbi oldalakon részletesebb vázlattervei találhatóak a legfelső réteg vizuális reprezentációjának megalapozására. A gombok felirattal vannak ellátva, a funkciójuk deklarációjának érdekében, a kártyák pedig vagy egy mezőt leíró címmel és alatta illetve mellette a mező tárolt értékének helyet biztosító szövegdobozzal, vagy alternatív esetben a kártyához tartozó fontos információk helyét jelölő üres, lekerekített téglalapokkal vannak ellátva.

#### 3.2.2 A menü

A menü egy szimplisztikus, gombokkal feltöltött sáv az oldal tetején, ami navigációt valósít meg a felépített komponensek között, megjeleníti a felhasználó vagyonát, illetve kijelentkezésre is ad lehetőséget.

Az itt található gombok a navigáció célpontját egyértelműen leíró címekkel vannak kiegészítve, és ebből természetesen adandóan, mind a címhez tartozó, megfelelő oldalhoz vezetnek.

username HOME MARKET ISSUES GARAGES LINES SCHEDULES 111,111 LOGOUT

3.2. ábra. A menü terve

### 3.2.3 A főoldal

Az alkalmazás kezdő oldalán láthatóak a felhasználó által birtokolt buszok. Ezen az oldalon a járművek egy kártyás nézetben találhatók, feltüntetve azok fontos adatait. A bal felső sarokban helyezkedik el a rendszám, a kártyán pedig a típus, évjárat, kilométerszám és a státusz, illetve a kártya legalján találhatóak a műszaki problémák, amennyiben vannak.

Ugyanezen az oldalon lehet állítani a garázst, a megfelelő mezőre kattintva, a jármű leírását változtatni, továbbá, a kedvenceket állítani a kis csillag ikon segítségével.

## YOUR BUSES

SHOW FAVOURITES

XXX-111

☆

TYPE

KM

YEAR

GARAGE

STATUS

LINE

DESCRIPTION

ISSUES

XXX-111

☆

TYPE

KM

YEAR

GARAGE

STATUS

LINE

DESCRIPTION

ISSUES

XXX-111

☆

TYPE

KM

YEAR

GARAGE

STATUS

LINE

DESCRIPTION

ISSUES

XXX-111

☆

TYPE

KM

YEAR

GARAGE

STATUS

LINE

DESCRIPTION

ISSUES

XXX-111

☆

TYPE

KM

YEAR

GARAGE

STATUS

LINE

DESCRIPTION

ISSUES

XXX-111

☆

TYPE

KM

YEAR

GARAGE

STATUS

LINE

DESCRIPTION

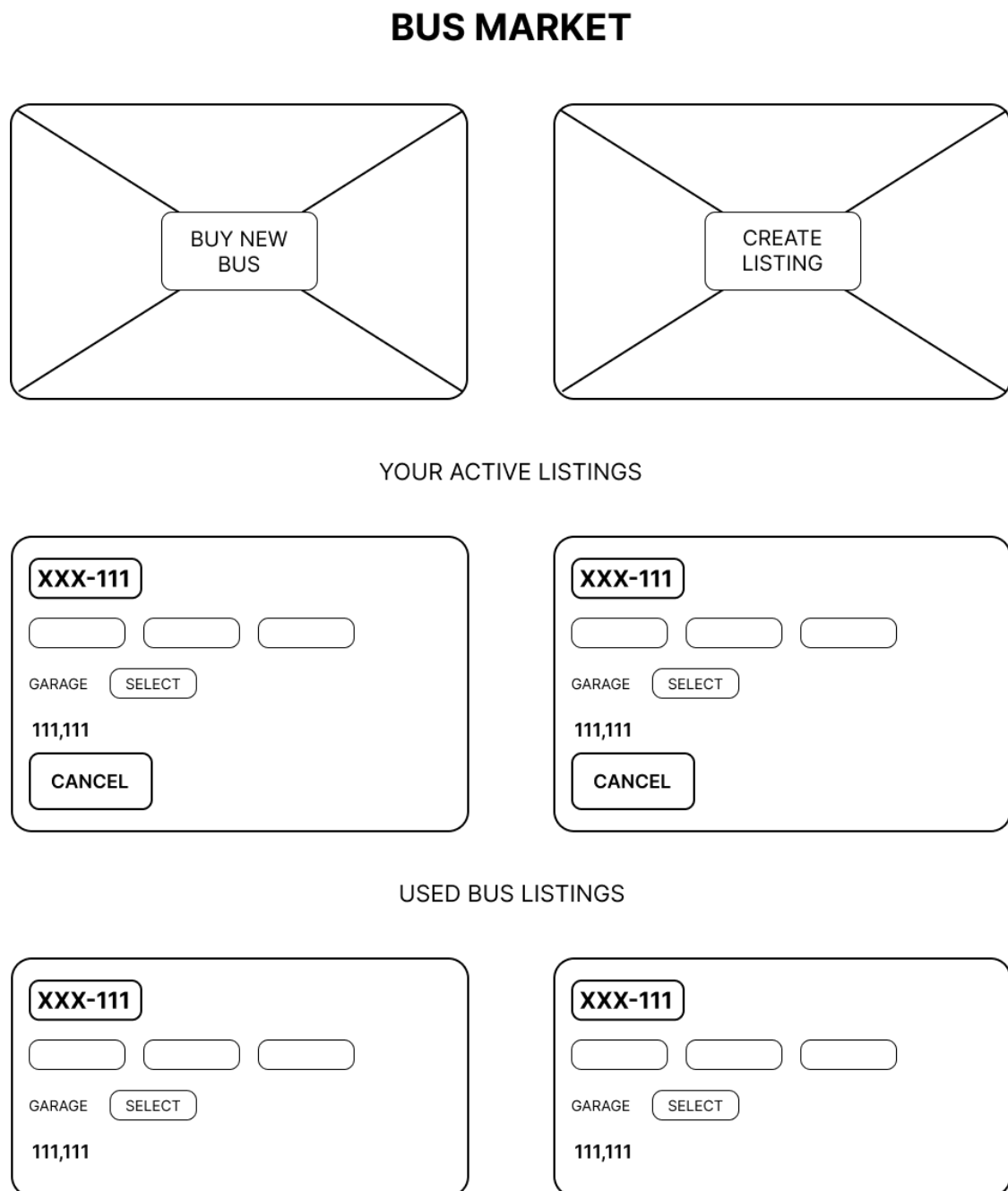
ISSUES

3.3. ábra. A főoldal drótvázterve

#### 3.2.4 A piac

A piac oldalon bonyolodik le a vásárlás és eladás kezelése. A bal fenti sarokban található blokkban lehet vadonatúj buszt beszerezni, a jobb felsőben pedig értékesítésre bocsájtani a meglévőket.

A lentebbi szekciók tartalmazzák a saját, illetve mások hirdetéseit, az adott rendszámmal, alatta pedig a busz adataival és az irányárral. Ezeken a kártyákon keresztül tud a felhasználó vásárolni használt buszt, vagy saját hirdetését lemondani.



3.4. ábra. A piac oldal drótvázterve



#### 3.2.5 A műszaki hibák

Ezen az oldalon a felhasználó saját buszaihoz tartozó műszaki hibák találhatók, egy átfogó, összegzett nézetben.

Itt van lehetőség új műszaki hiba bejelentésére, illetve a fennálló problémák orvoslására. A hibákat tároló kártyák feltűntetik az adott rendszámot, alatta pedig a hiba bejelentésének idejét, a javítási árat és időt, illetve a fennakadást meghatározó leírást is.

### YOUR BUS ISSUES

REPORT NEW ISSUE

XXX-111

REPAIR

DESCRIPTION

XXX-111

REPAIR

DESCRIPTION

XXX-111

REPAIR

DESCRIPTION

3.5. ábra. A műszaki hibák oldalának drótvázterve

#### 3.2.6 A vonalak

Ezen az oldalon találhatóak a vonalak egy paneles nézetben, ahol minden doboz egy vonal ellátását, illetve azok adatait tartalmazza.

A bal felső sarokban található a vonalszám, alatta pedig a hozzá tartozó információk, mint a menetidő, a garázsmeneti idő, a biztosító garázs, illetve az aktuális időszámban éppen szolgáltatást biztosító menetrend adatai.

Lejjebb a kisebb egységekben vannak feltüntetve a vonalhoz tartozó, legjobb szolgáltatást biztosító beosztások, minden időkeretben, jelezve a hozzájuk tartozó felhasználót és a szükséges információkat.

#### SERVICE LINES

106

Currently serviced by ...

Schedules in service

USERNAME

USERNAME

105

Currently out of service

Schedules in service

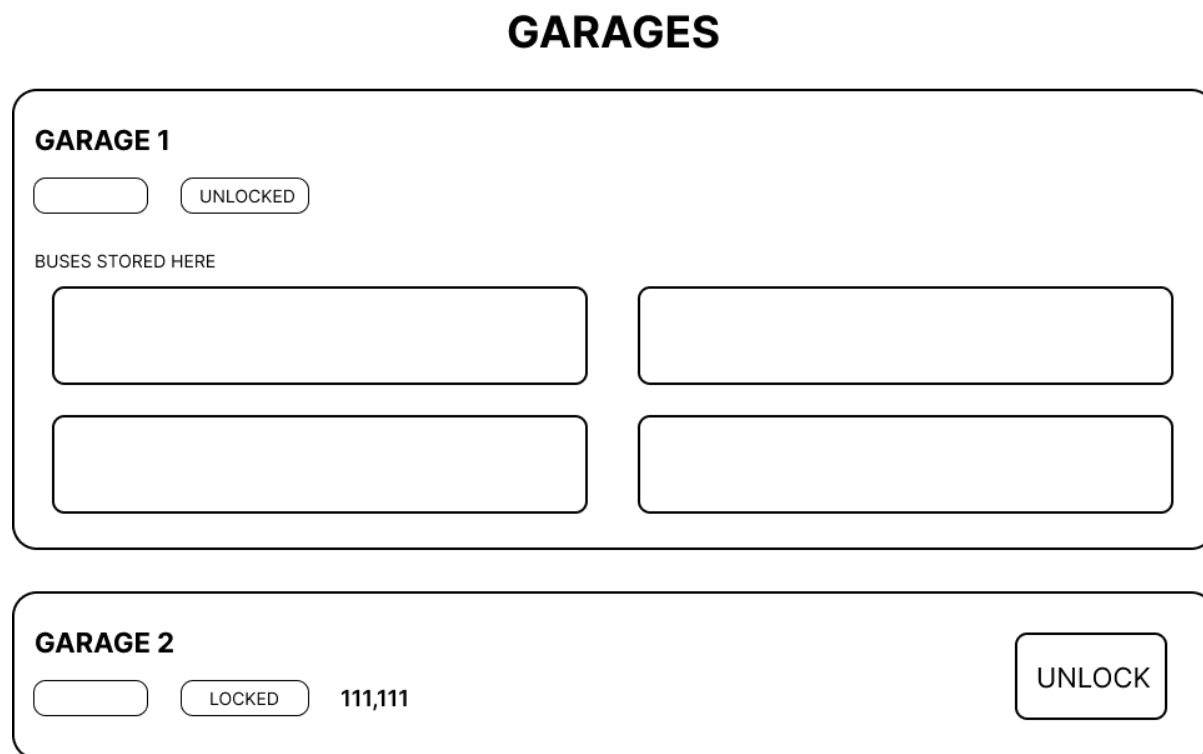
No active schedules

3.6. ábra. A vonalak oldal drótvázterve

### 3.2.7 A garázsok

Ezen az oldalon találhatóak a garázsok és a felhasználó által azokban tárolt járművek, egy csoportos nézetben megvalósítva.

Minden garáznál megjelenítésre kerül a helyszínük, a bennük tárolt járművek kompakt nézetű adatai, kisebb táblákban, illetve a még feloldásra nem került garázs esetén a vásárlási ár, illetve az ahhoz szükséges gomb.



3.7. ábra. A garázsok oldalának drótvázterve

### 3.2.8 A menetrendek

Ezen az oldalon valósul meg a menetrendek kezelése. Az oldal tetején nyílik opció új menetrend létrehozására, alatta pedig a felhasználó által készített menetrendek találhatóak.

Ezekben a táblákban az adott vonal, illetve a szolgáltatás fontosabb adatai láthatóak, mint a biztosító garázs, a gyakoriság, a vállalt irányár, és az időkeret.

Ezekre kattintva megnyílik az adott menetrend kiterjesztett nézete, ahol ismét láthatóak a fontosabb információk, illetve egy extenzív nézet a forgalmi indulásokat csoportosító blokkokhoz. A blokkokban láthatóak az oda tartozó indulási idők, illetve itt van lehetősége a felhasználónak buszokat rendelni a blokkokhoz. Amennyiben már van hozzárendelt jármű, egy gomb is megjelenik, amivel el lehet azt távolítani a blokkból.

A jobb alsó sarokban található gombbal lehet elmenteni a feljebb részletezett beállításokat.

## SCHEDULES

CREATE SCHEDULE

### YOUR SCHEDULES

106

DELETE

105

DELETE

106

SLOTS:

**BLOCK 1**

Assign Bus: **XXX-111**

REMOVE

**BLOCK 2**

Assign Bus:

SAVE ASSIGNMENTS

3.8. ábra. A menetrendek oldalának drótvázterve

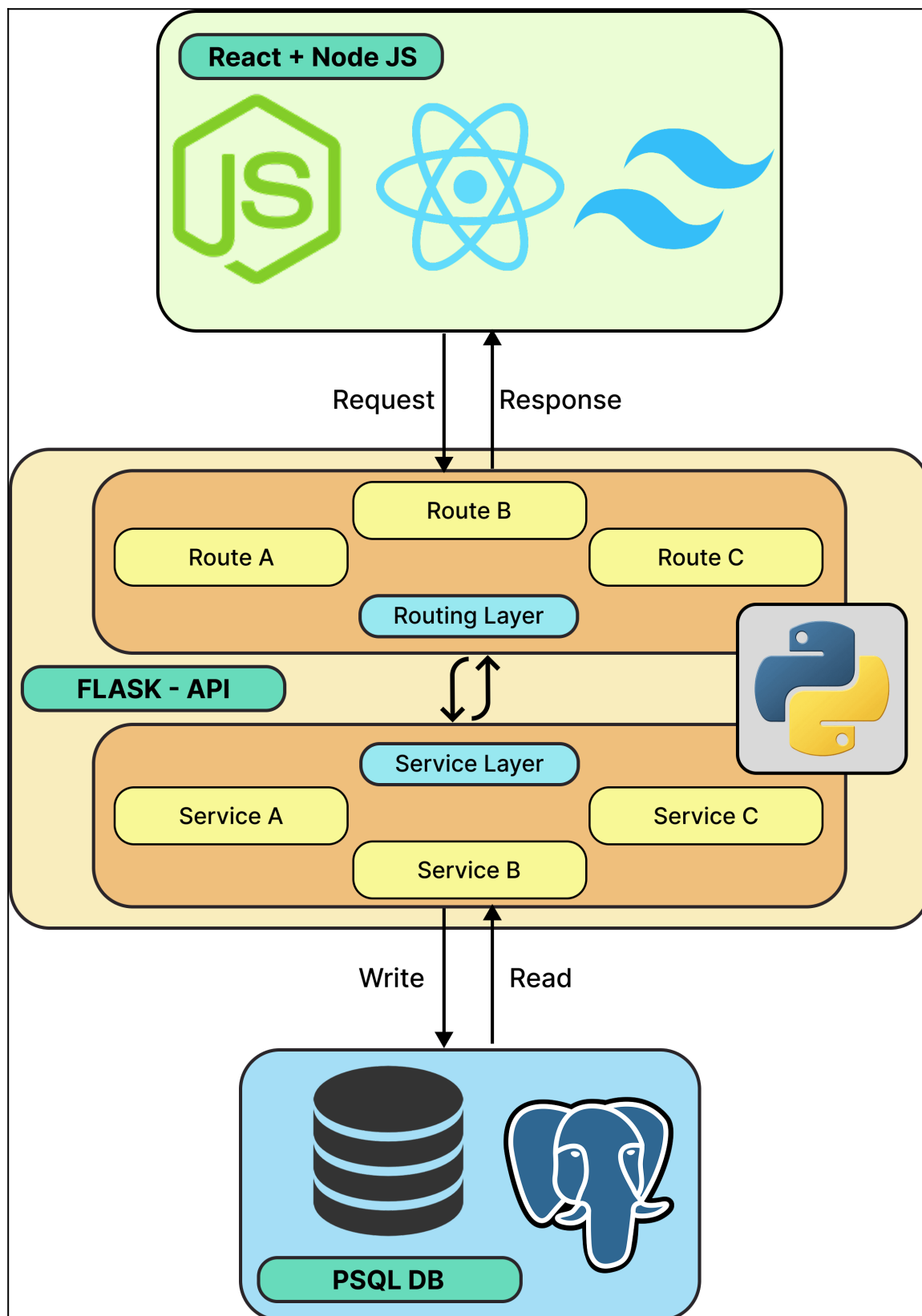
## 3.3 Az alkalmazás rétegei

Az applikáció egy klasszikus három szintes megvalósításon alapul, ezek az adatbázis, a backend és a frontend rétegek.

Az adatbázis vállal felelősséget a weboldal használatához szükséges elemi információk tárolásáért, illetve többek között bizonyos adatok megfelelő formaiságáért, korlátozásokon keresztül. A dolgozat megvalósításához ebben az esetben egy PSQl adatbázist választottam.

A backend réteg fogja közvetíteni az adatfolyamot a felhasználói felület és az adattár között. Ezen belül két szinten valósul meg a logika, az egyik a szolgáltatási szint, ami az adatbázis írására, olvasására, adatok módosítására, hibakezelésre, bemenetek ellenőrzésére, illetve egyéb logikai funkciók ellátására alkalmas. A másik az irányítási szint, ahol sokkal kevesebb komputációs művelet zajlik, célja inkább az API-endpointok felfedése a frontend réteg felé és a HTTP-requestek használatának lehetővé tétele. A backend réteget egy python Flask-applikációban készítettem el.

A felhasználói felület teszi az ember által kezelhetővé az alkalmazást. Elérhetővé teszi a vizuális interfész használatát az alkalmazáshoz, illetve némi bemeneti validációt is kezel. A frontend esetében alkalmazott technológiának a React + Node JS párost választottam, az esztétikai kiegészítésre pedig Tailwind CSS-t használtam.



3.9. ábra. A rendszer architektúrája nagy vonalakban



### 3.4.1 A buszokat tartalmazó tábla

Ebben a táblában a buszok adatai tárolandók. Innen táplálkozik az alkalmazás, amikor egy busz tulajdonost vált, új vonalra kerül, vagy esetleg műszaki hibával rendelkezik.

Oszlop neve	Típus	Leírás
plate (PK)	varchar	A busz rendszáma
type	varchar	A busz típusa
km	int	A busz kilométerszáma
year	int	A busz évjárata
garage (FK)	int	A buszhoz tartozó garázs
description	varchar	A busz leírása
status	varchar	A busz státusza
line (FK)	varchar	A busz beosztott vonala
owner (FK)	varchar	A busz tulajdonosa
favourite	boolean	Kedvencek közé tartozik-e

3.7. táblázat. A busz tábla

### 3.4.2 A garázsokat tartalmazó tábla

Ebben a táblában a garázsok tárolandók. Viszonylag kevés adatnak ad helyet, viszont annál többször használt, fontos idegen kulcs.

Oszlop neve	Típus	Leírás
id (PK)	int (serial)	A garázs azonosítója
name	varchar	A garázs neve
location	varchar	A garázs helyszíne

3.8. táblázat. A garázs tábla

### 3.4.3 A vonalakat tartalmazó tábla

Ebben a táblában a vonalak tárolandók. Hasonlóképp a garázsokhoz, itt is inkább referenciaként fontos adatok vannak.

Oszlop neve	Típus	Leírás
name (PK)	varchar	A vonal neve
garage_id (FK)	int	A garázs azonosítója, amely a vonalat ellátja
travel_time_garage	int	A garázsmenti idő (perc)
travel_time_line	int	A vonal menetideje (perc)

3.9. táblázat. A vonal tábla



### 3.4.4 A műszaki hibákat tartalmazó tábla

Ebben a táblában a műszaki hibák tárolandók. Fontos szerepe van amikor a felhasználók áttekintenek a fennálló problémáikat, illetve amikor a piacon látni szeretnék, melyik járműnek mi rejlik a váza alatt.

Oszlop neve	Típus	Leírás
id (PK)	int (serial)	A műszaki hiba azonosítója
bus (FK)	varchar	A hibához tartozó busz rendszáma
time	date	A hiba bejelentésének időpontja
repair_time	interval	A javításhoz szükséges idő
repair_cost	int	A javítás ára
description	varchar	A hiba leírása

3.10. táblázat. A műszaki hibákat tartalmazó tábla

### 3.4.5 A menetrendeket tartalmazó tábla

Ebben a táblában a felhasználók által létrehozott menetrendek tárolandók. Ugyan a hozzárendelésekre külön segédtábla szolgál, mégis létfontosságú adatokat tárol annak kiszámításához, hogy melyik menetrend kerülhet aktív stádiumba.

Oszlop neve	Típus	Leírás
id (PK)	int(serial)	A menetrend azonosítója
username (FK)	varchar	A menetrend készítője
line_name (FK)	varchar	A menetrend által érintett vonal neve
garage_id (FK)	int	A menetrendet ellátó garázs azonosítója
start_time	time	A menetrend kezdete
end_time	time	A menetrend vége
frequency	int	A buszok gyakorisága (perc)
bid_price	int	A szolgáltatás irányára
status	varchar	A menetrend státusza
frame	varchar	A kiválasztott időkeret

3.11. táblázat. A menetrendeket tartalmazó tábla

### 3.4.6 A felhasználókat tartalmazó tábla

A felhasználók nevei, jelszavuk és elérhető valutájuk található ebben a táblázatban. A jelszavak hasheléssel titkosítva vannak tárolva.

Oszlop neve	Típus	Leírás
username (PK)	varchar	A felhasználónév
password	varchar	A jelszó
balance	varchar	A felhasználó egyenlege

3.12. táblázat. A felhasználókat tároló tábla

### 3.4.7 A piaci hirdetéseket tartalmazó tábla

Ebben a táblában tárolódnak a felhasználók által létrehozott hirdetések, így törzseleme a piac működésének, az adatok megjelenítéséhez szükséges lekérdezéseken túl a felhasználók közötti adás-vétel is itt kerül rögzítésre.

Oszlop neve	Típus	Leírás
id (PK)	int(serial)	A hirdetés azonosítója
bus_plate (FK)	varchar	A hirdetéshez tartozó busz rendszáma
seller_username	varchar	A hirdetés tulajdonosa
price	int	A meghirdetett irányár
status	varchar	A hirdetés státusza
created_at	time	A hirdetés létrehozásának időpontja
sold_at	time	Az értékesítés időpontja

3.13. táblázat. A hirdetéseket tartalmazó tábla

### 3.4.8 Segédtablák

Az alkalmazás megvalósításához két segédtablára van szükség. Ezeknek a tábláknak nincsen elsőszámú egyedi azonosítójuk, csupán arra valók, hogy összeköttetést biztosítsanak bizonyos érték párok között.

Az első ilyen tábla a felhasználók és a garázsok kapcsolatának tárolását valósítja meg, egészen pontosan értékpárokat tárol, amiben egy bizonyos felhasználó, és egy általa birtokolt garázs található.

Oszlop neve	Típus	Leírás
username (FK)	varchar	A felhasználónév
garage-id (FK)	int	A felhasználó által feloldott garázs

3.14. táblázat. A felhasználók és garázsok birtokos viszonyát tároló tábla

A második segéd tábla pedig azért felelős, hogy tárolja, melyik busz, melyik menetrenden hanyadik pozíciót tölti be.

Oszlop neve	Típus	Leírás
schedule_id (FK)	int	A menetrend azonosítója
block_idx	int	A menetrend egyik pozíciójának azonosító száma
bus_plate (FK)	varchar	A pozícióhoz rendelt busz rendszáma

3.15. táblázat. A menetrendek beosztásait tároló tábla

## 3.5 A backend réteg moduljai

Ebben a rétegben valósul meg a logika, ami kiszolgálja az alkalmazást. Minden komponens egy bizonyos elemért felelős, kommunikál az adatbázis réteg szükséges elemeivel, és információt dolgoz fel, illetve továbbít a fenebbi rétegek felé.

Az alábbi szekciókban az egyes komponensek részletesebb leírása található. A fontosabb függvények tevékenységeinek definiálása, illetve szekvencia-diagramok szolgálnak a modulok funkcionális követelményeinek lefektetésére.

### 3.5.1 A buszokért felelős modul

Ez a modul látja el maguknak a buszoknak a kezelését, létrehozását, módosítását. A komponens elemeinek részletezését a 3.16. táblázat tartalmazza.

get_buses_for_user		
Bemenet	Kimenet	Tevékenység
Adott felhasználónév	A felhasználó által birtokolt buszok	A függvény lekérdezi az adatbázisból azokat a buszokat, ahol a tulajdonos mező egyezik a megadott felhasználónévvel.

3.16. táblázat. A buszokat lekérdező függvény

create_bus		
Bemenet	Kimenet	Tevékenység
A busz létrehozásához szükséges adatok(rendszám, típus, garázs, évjárat)	Értesítő üzenet és státuszkód	A függvény a megadott paraméterek alapján, illetve az ezeken kívül szükséges adatokat alapértelmezettre állítva létrehoz egy új buszt az adatbázisban, illetve validálja a megadott adatokat.

3.17. táblázat. A buszokat létrehozó függvény

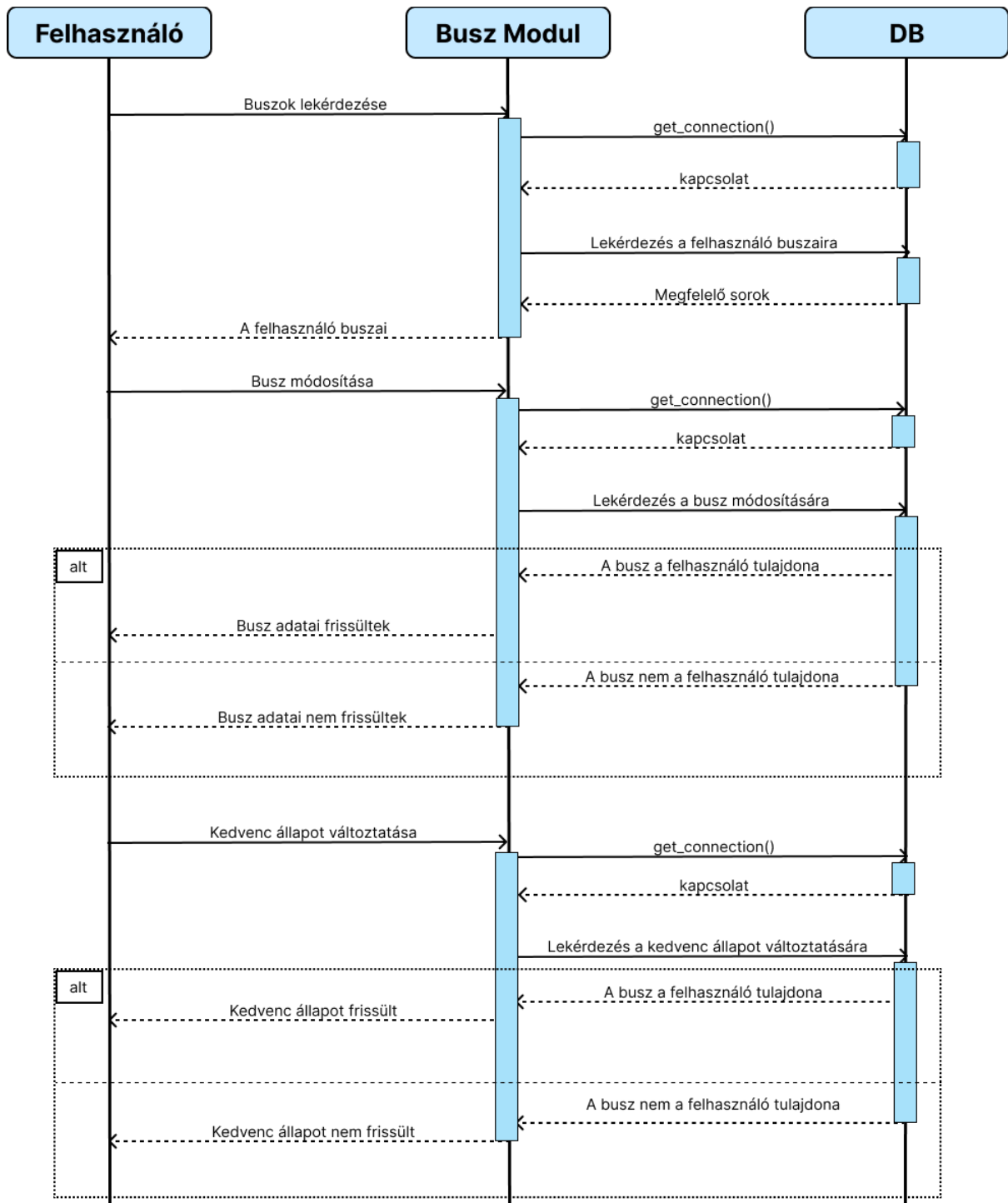
update_bus		
Bemenet	Kimenet	Tevékenység
Rendszám	Értesítő üzenet és státuszkód	A függvény a megadott rendszám és új paraméterek alapján módosítja egy busz adatait az adatbázisban, illetve validálja a megadott adatokat. Amennyiben a rendszám nem található, hibaüzenetet küld.
A busz új adatai		

3.18. táblázat. A buszokat módosító függvény

toggle_favourite		
Bemenet	Kimenet	Tevékenység
Adott felhasználónév	Értesítő üzenet és státuszkód	A függvény a megadott felhasználónév és rendszám vizsgálata után, amennyiben egyezést talál, tehát a felhasználó a birtokosa az érintett busznak,

toggle_favourite		
Adott rendszám		módosítja a kedvenc mező értékét az adatbázisban, ellenkező esetben pedig hibaüzenetet küld.

3.19. táblázat. A kedvenc jelzõt változtató függvény



3.11. ábra. A busz modul szekvencia-diagramja

### 3.5.2 A garázsokért felelős modul

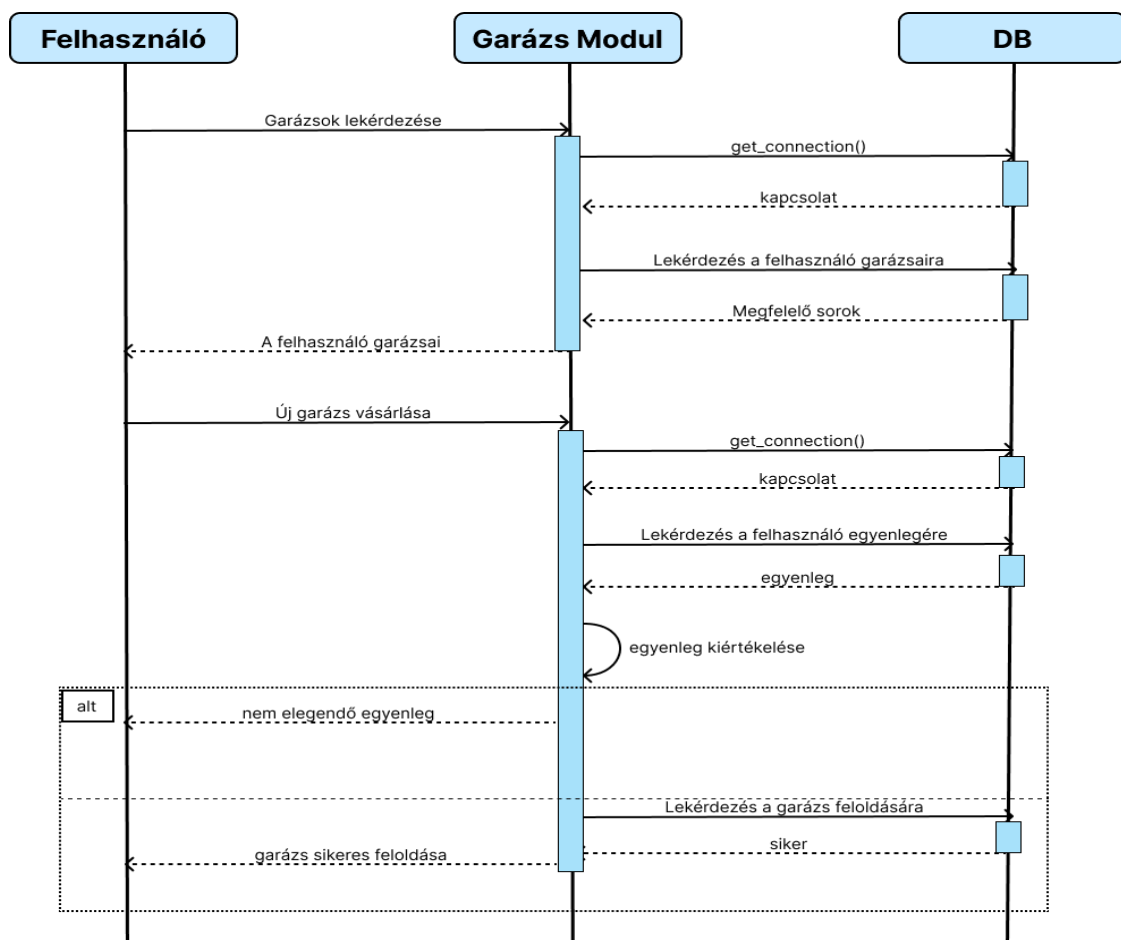
Ez a modul intézi a garázsok körüli logikát, azok feloldását a felhasználó számára, illetve adataik továbbítását a fentebbi rétegekbe.

list_garages_for_user		
Bemenet	Kimenet	Tevékenység
Adott felhasználónév	A felhasználó által feloldott garázsok	A függvény megkeresi a felhasználók és az általuk feloldott garázsokat tartalmazó táblában azokat az értékeket, ahol a megadott felhasználó megtalálható, majd at így kinyert garázsok azonosítója alapján megadja azon garázsok többi mezőjének értékét is.

3.20. táblázat. A garázsokat lekérdező függvény

unlock_garage_for_user		
Bemenet	Kimenet	Tevékenység
Adott felhasználónév	Értesítő üzenet és státuszkód	A függvény ellenőrzi, hogy a megadott garázs fel van-e oldva már a felhasználónak, illetve, hogy van-e a feloldásra egyenlő egyenlege. Amennyiben igen, létrehozza a felhasználó-garázs kapcsolatokat nyilvántartó táblában az új kapcsolatot, ellenkező esetben hibaüzenetet dob.
Adott garázs azonosító		

3.21. táblázat. A garázst feloldó függvény



3.12. ábra. A garázs modul szekvencia-diagramja

### 3.5.3 A vonalakért felelős modul

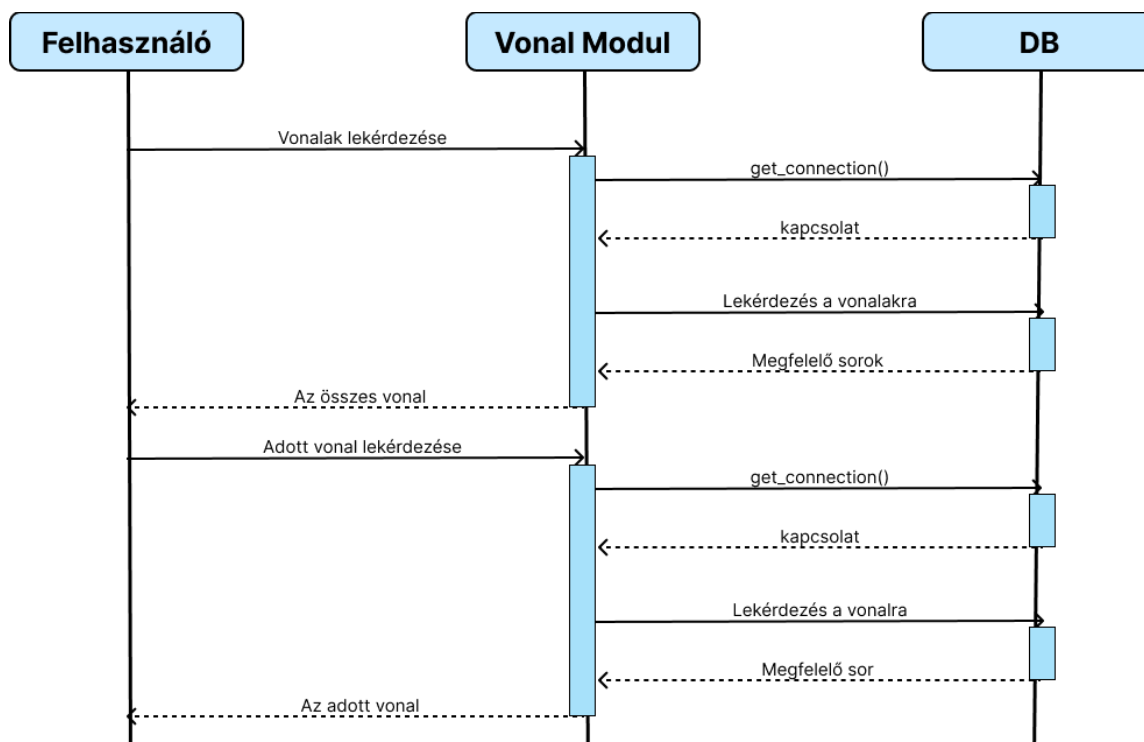
Ez a modul felel a vonalak információjának eléréséért. Nem igazán terjedelmes, de fontos alapvető információk elérését valósítja meg.

list_lines		
Bemenet	Kimenet	Tevékenység
-	A vonalak adatai	A függvény lekérdezi a vonalak adatait az adatbázisból.

3.22. táblázat. A vonalakat lekérdező függvény

get_line		
Bemenet	Kimenet	Tevékenység
Adott vonal neve	A vonal adatai	A függvény lekérdezi az adott vonal adatait az adatbázisból.

3.23. táblázat. Egy vonalat lekérdező függvény



3.13. ábra. A vonal modul szekvencia-diagramja

### 3.5.4 A műszaki hibákért felelős modul

Ez a modul felel a műszaki hibák léterhozásáért, megjavításáért, illetve adataik közvetítéséért.

list_issues_by_bus		
Bemenet	Kimenet	Tevékenység
Adott rendszám	A busz műszaki hibái	A függvény lekérdezi az adott busz műszaki hibáinak adatait az adatbázisból.

3.24. táblázat. Adott busz műszaki hibáit lekérdező függvény

list_issues_for_user		
Bemenet	Kimenet	Tevékenység
Adott felhasználónév	A felhasználó összes buszának műszaki hibái	A függvény megkeresi a buszokat tároló táblában azokat az értékeket, ahol az adott felhasználó a tulajdonos, majd lekérdezi az így megtalált buszok műszaki hibáinak adatait az adatbázisból.

3.25. táblázat. A felhasználó műszaki hibáit lekérdező függvény

list_all_issues		
Bemenet	Kimenet	Tevékenység
-	Az összes műszaki hiba	A függvény lekérdezi az összes hibát az adatbázisból

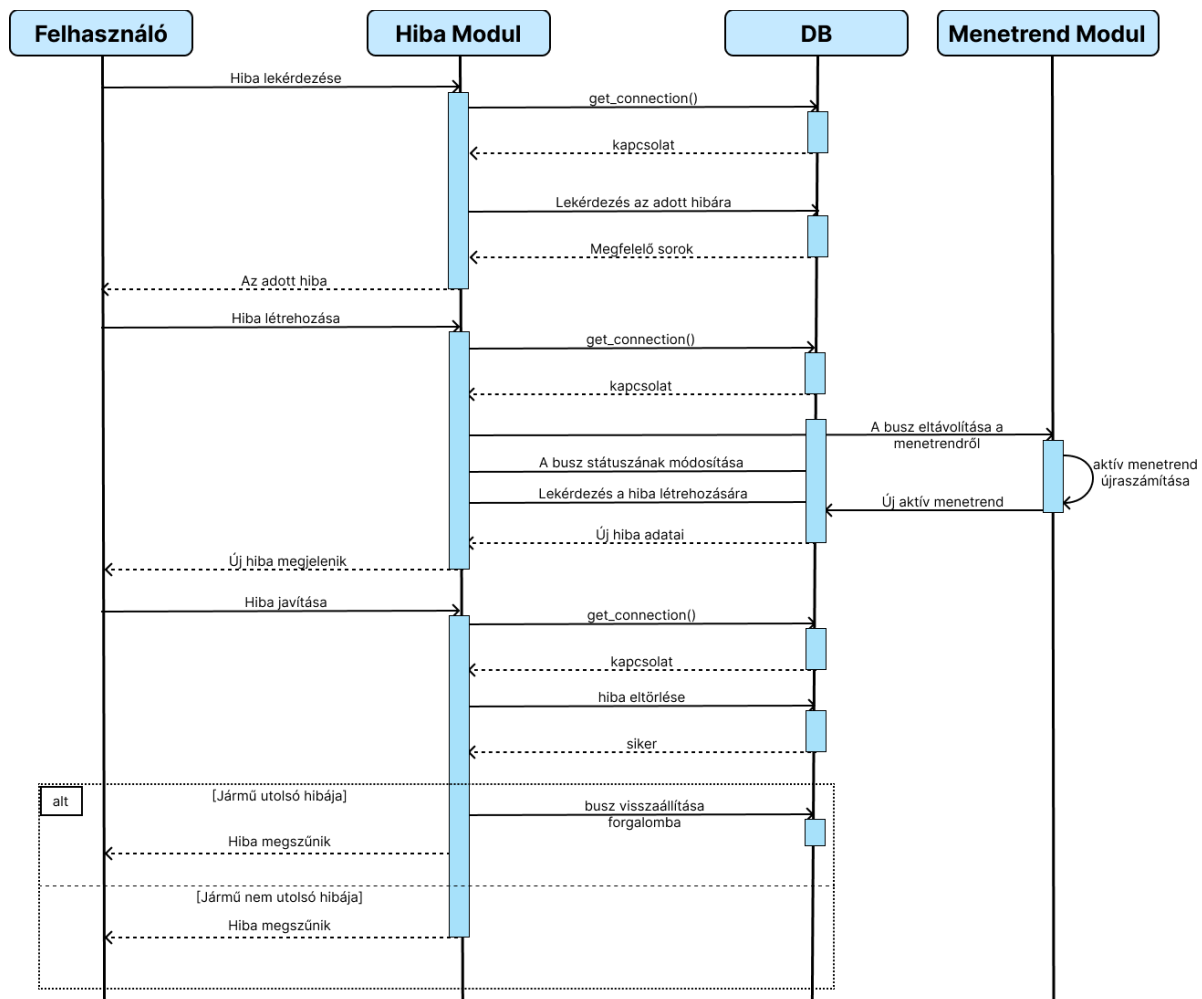
3.26. táblázat. Az összes műszaki hibát lekérdező függvény

create_issue		
Bemenet	Kimenet	Tevékenység
Az új műszaki hiba adatai	Értesítő üzenet és státuszkód	A függvény létrehoz egy új műszaki hibát a megadott adatok alapján. Ezen felül módosítja az érintett busz státuszát, jelezve, hogy szervízben van. Amennyiben a jármű egy menetrendi beosztáson szerepel, onnan eltávolítja, leszedi a vonalról, majd újraszámítja az érintett vonalon, hogy melyik menetrend kaphat aktív státuszt.

3.27. táblázat. Műszaki hibát létrehozó függvény

remove_issue		
Bemenet	Kimenet	Tevékenység
Adott műszaki hiba azonosítója	Értesítő üzenet és státuszkód	A függvény eltávolítja az adott műszaki hibát a rendszerből. Amennyiben az érintett busznak ezután nem marad több problémája, kikerül a szervízből.

3.28. táblázat. Műszaki hibát törlő függvény



3.14. ábra. A műszaki hibák moduljának szekvencia-diagramja



### 3.5.5 A piacért felelős modul

Ez a modul felel a piac megvalósításáért, az új buszok vásárlásáért, a használt buszok adás-vételéért, és az anyagi kötelezettségek lebonyolításáért.

Az eredeti tervek szerint a buszok eladása úgy zajlott, hogy a felhasználó pénzért vissza adta a rendszernek a járművet, ami aztán felkerült a piacra. Később ezt felváltotta a hirdetés alapú vásárlás, de megmaradt az eredeti felállás is, amivel magától a rendszertől lehet buszokat vásárolni.

list_market_buses		
Bemenet	Kimenet	Tevékenység
-	A piacon elérhető, még nem tulajdonolt buszok adatai	A függvény lekérdezi az adatbázisból az összes buszt, amelynek még nincs tulajdonosa, és visszaadja azok adatait.

3.29. táblázat. A piac buszait lekérdező függvény

list_new_bus_models		
Bemenet	Kimenet	Tevékenység
-	Az újonnan megvásárolható buszmodellek listája	A függvény visszaadja a definiált új buszmodelleket, listaként, a front-end megjelenítéséhez megfelelő formátumban.

3.30. táblázat. Az újonnan elérhető modelleket lekérdező függvény

list_active_listings		
Bemenet	Kimenet	Tevékenység
Aktív hirdetések listája	Az aktív hirdetések listája	A függvény lekérdezi az aktív státuszú hirdetéseket az adatbázisból, és az azokhoz tartozó buszadatokat is összekapcsolja

3.31. táblázat. A piaci hirdetéseket lekérdező függvény

create_listing		
Bemenet	Kimenet	Tevékenység
Az eladó felhasználóneve	Értesítő üzenet és státuszkód	A függvény létrehoz egy új hirdetést az adott buszhoz, ha az valóban a felhasználó tulajdona és még nem aktív hirdetés alatt áll. Az adatbázisba beilleszti az új hirdetést, majd visszatér egy megerősítő-, vagy hibaüzenettel.
Az érintett busz rendszáma		
Az eladási ár		

3.32. táblázat. Új hirdetést létrehozó függvény

cancel_listing		
Bemenet	Kimenet	Tevékenység
Az eladó felhasználóneve	Értesítő	A függvény ellenőrzi, hogy a megadott hirdetés aktív-e és a felhasználóé-e. Amennyiben igen,

cancel_listing		
A hirdetés azonosítója	üzenet és státuszkód	visszavonja a hirdetést és visszatér egy megerősítő üzenettel.

3.33. táblázat. Hirdetést lemondó függvény

purchase_listing		
Bemenet	Kimenet	Tevékenység
A vásárló felhasználóneve	Értesítő üzenet és státuszkód	A függvény lebonyolítja az adott aktív hirdetés megvásárlását. Ellenőrzi a vevő egyenlegét, a busz tulajdonjogát, a garázs elérhetőségét, majd frissíti az érintett rekordokat. A buszt áthelyezi a vevő garázsába és alaphelyzetbe állítja.
A hirdetés azonosítója		
A célgarázs azonosítója		

3.34. táblázat. Hirdetés megvásárlását megvalósító függvény

buy_bus		
Bemenet	Kimenet	Tevékenység
A vásárló felhasználóneve	Értesítő üzenet és státuszkód	A függvény lehetővé teszi, hogy egy elérhető, tulajdonos nélküli buszt megvásároljon egy felhasználó. Levonja az összeget a felhasználó egyenlegéből, a busz tulajdonosát és helyét frissíti az adatbázisban, majd visszaad egy megerősítő üzenetet. Amennyiben a megadott garázs nem elérhető a felhasználó számára, vagy nincs elegendő egyenlege, hibaüzenetet küld.
A busz rendszáma		
A célgarázs azonosítója		

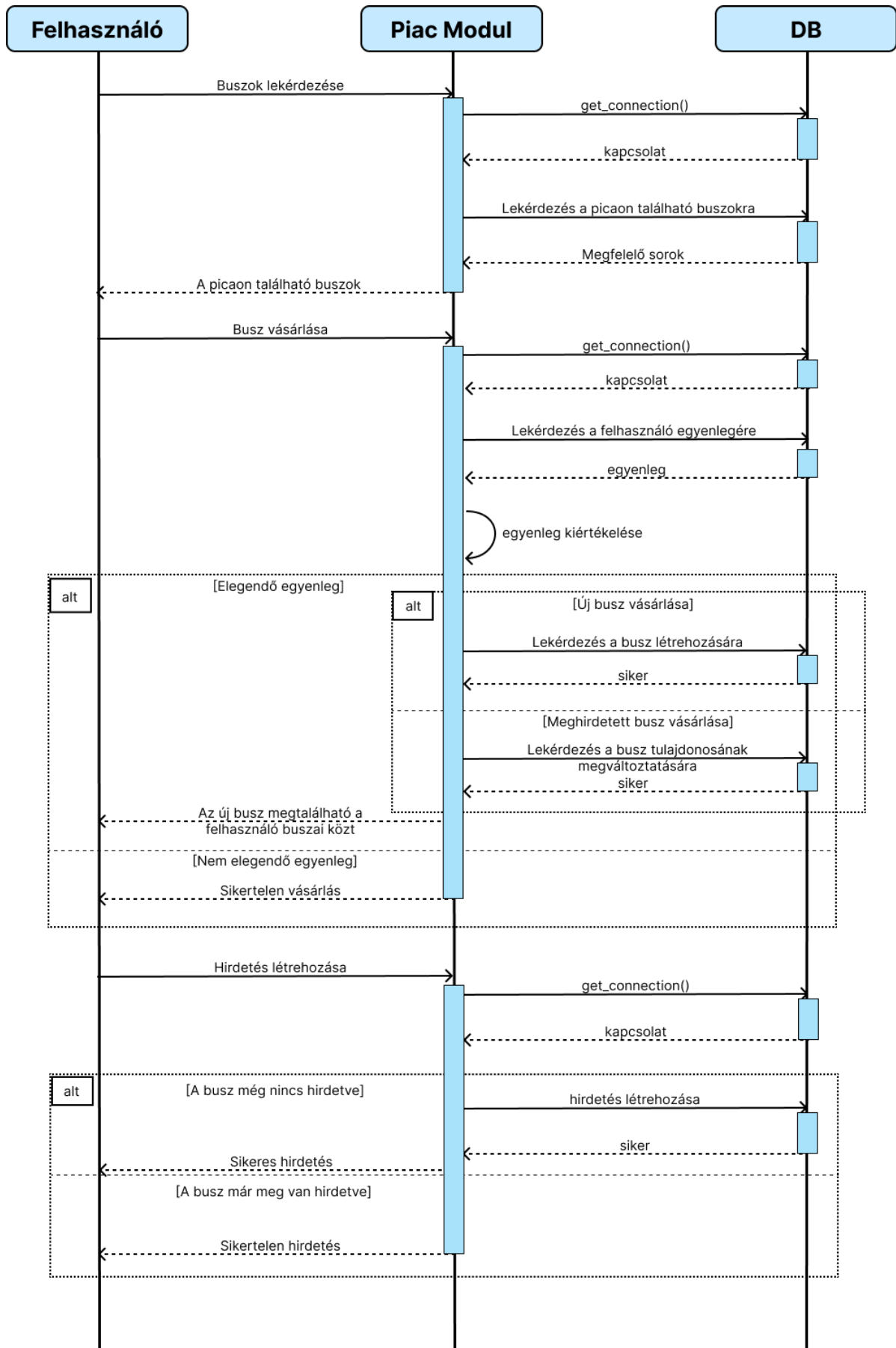
3.35. táblázat. Busz megvásárlását megvalósító függvény

purchase_new_bus		
Bemenet	Kimenet	Tevékenység
A vásárló felhasználóneve	Értesítő üzenet és státuszkód	A függvény ellenőrzi a megadott adatok helyességét, majd létrehoz egy új buszt az adatbázisban. Levonja a vételárat a felhasználó egyenlegéből, beállítja a busz tulajdonosát és adatait, és visszatér a frissített egyenleggel. Amennyiben a megadott garázs nem elérhető a felhasználó számára, vagy nincs elegendő egyenlege, hibaüzenetet küld.
Az új busz adatai		
A célgarázs azonosítója		

3.36. táblázat. Új busz megvásárlását megvalósító függvény

sell_bus		
Bemenet	Kimenet	Tevékenység
Az eladó felhasználóneve	Értesítő üzenet és státuszkód	A függvény létrehoz egy új hirdetést az adott buszhoz, ha az a megadott felhasználó tulajdonában van-e és még nincs aktív hirdetés alatt. Amennyiben nincs probléma, a hirdetés azonnal aktív státuszba kerül, ellenkező esetben hibaüzenetet dob.
A busz rendszáma		

3.37. táblázat Busz eladását megvalósító függvény



3.15. ábra. A piac modul szekvencia-diagramja

### 3.5.6 A menetrendekért felelős modul

Ez a modul felel a menetrendek generálásáért, annak eldöntéséért, hogy melyik menetrend nyeri az aktív státuszt, illetve a buszok elhelyezéséért is a generált indulásávokra.

get__schedule		
Bemenet	Kimenet	Tevékenység
A menetrend azonosítója	A menetrend adatai	A függvény lekérdezi az adott menetrend részletes adatait az adatbázisból.

3.38. táblázat. Adott menetrendet lekérdező függvény

list__schedules_for__user		
Bemenet	Kimenet	Tevékenység
Adott felhasználónév	A felhasználó menetrendjei	A függvény lekérdezi az adott felhasználóhoz tartozó összes menetrendet az adatbázisból.

3.39. táblázat. Felhasználó menetrendjeit lekérdező függvény

create__schedule		
Bemenet	Kimenet	Tevékenység
Adott felhasználónév	Értesítő üzenet és státuszkód	A függvény a megadott adatok alapján létrehoz egy menetrendet, majd lefuttatja a licitgyőztes menetrend kiszámítását.
Menetrend adatai		

3.40. táblázat. Menetrendet létrehozó függvény

delete__schedule		
Bemenet	Kimenet	Tevékenység
A menetrend azonosítója	Értesítő üzenet és státuszkód	A függvény törli az érintett menetrendet, a buszokat mentesíti a hozzárendelésektől, majd lefuttatja a licitgyőztes menetrend kiszámítását.

3.41. táblázat. Menetrendet törölő függvény

list__schedules_for__line		
Bemenet	Kimenet	Tevékenység
Adott vonal neve	A vonalhoz tartozó menetrendek listája	A függvény lekérdezi a megadott vonalhoz tartozó összes menetrendet.

3.42. táblázat. Adott vonalhoz készített menetrendeket lekérdező függvény

get_line_winners		
Bemenet	Kimenet	Tevékenység
-	A vonalhoz tartozó aktív menetrendek listája	A függvény lekérdezi a megadott vonalhoz tartozó összes aktív menetrendet, azaz azokat, amelyek jelenleg a nyertes ütemtervek.

3.43. táblázat. Adott vonal győztes menetrendjeit kiszámító függvény

select_winner_for_line_frame		
Bemenet	Kimenet	Tevékenység
Adott vonal neve	Adott vonalon, az adott időpontot nyerő menetrend azonosítója.	A függvény kiválasztja az időszávon belüli érvényes és legjobb ajánlatot, és frissíti az érintett menetrendek státuszait. Csak olyan menetrend nyerhet, ami fel van töltve buszokkal.
Időszá		

3.44. táblázat. Adott vonal adott időszájában győztes menetrendet kiszámító függvény

has_all_assignments		
Bemenet	Kimenet	Tevékenység
A menetrend azonosítója	Logikai érték (Igaz/Hamis)	A függvény ellenőrzi, hogy az adott menetrend összes helye fel van-e töltve buszokkal.

3.45. táblázat. Függvény, ami megvizsgálja hogy adott menetrend fel van-e töltve

generate_slots		
Bemenet	Kimenet	Tevékenység
Kezdeti idő	Időpontok listája	A függvény legenerálja az indulási időpontokat a megadott intervallum és gyakoriság alapján.
Záróidő		
Gyakoriság (perc)		

3.46. táblázat. Az indulási időket kiszámító függvény

plan_buses_for_schedule		
Bemenet	Kimenet	Tevékenység
A menetrend azonosítója	A buszbeosztási terv adatai	A függvény a vonal menetidejét és a garázsmeneti időt beszámítva kiszámolja, hány busz szükséges, mikor indulnak, és mikor tartanak pihenőt

3.47. táblázat. Az indulásokhoz buszbeosztást kiszámító függvény

bid_cap_for_frequency		
Bemenet	Kimenet	Tevékenység
Az indulás gyakorisága percben	Maximális licitkorlát	A függvény kiszámítja az adott gyakoriságú menetrendhez tartozó licitkorlátot az időszáv figyelembevételével.
Időszáv neve		

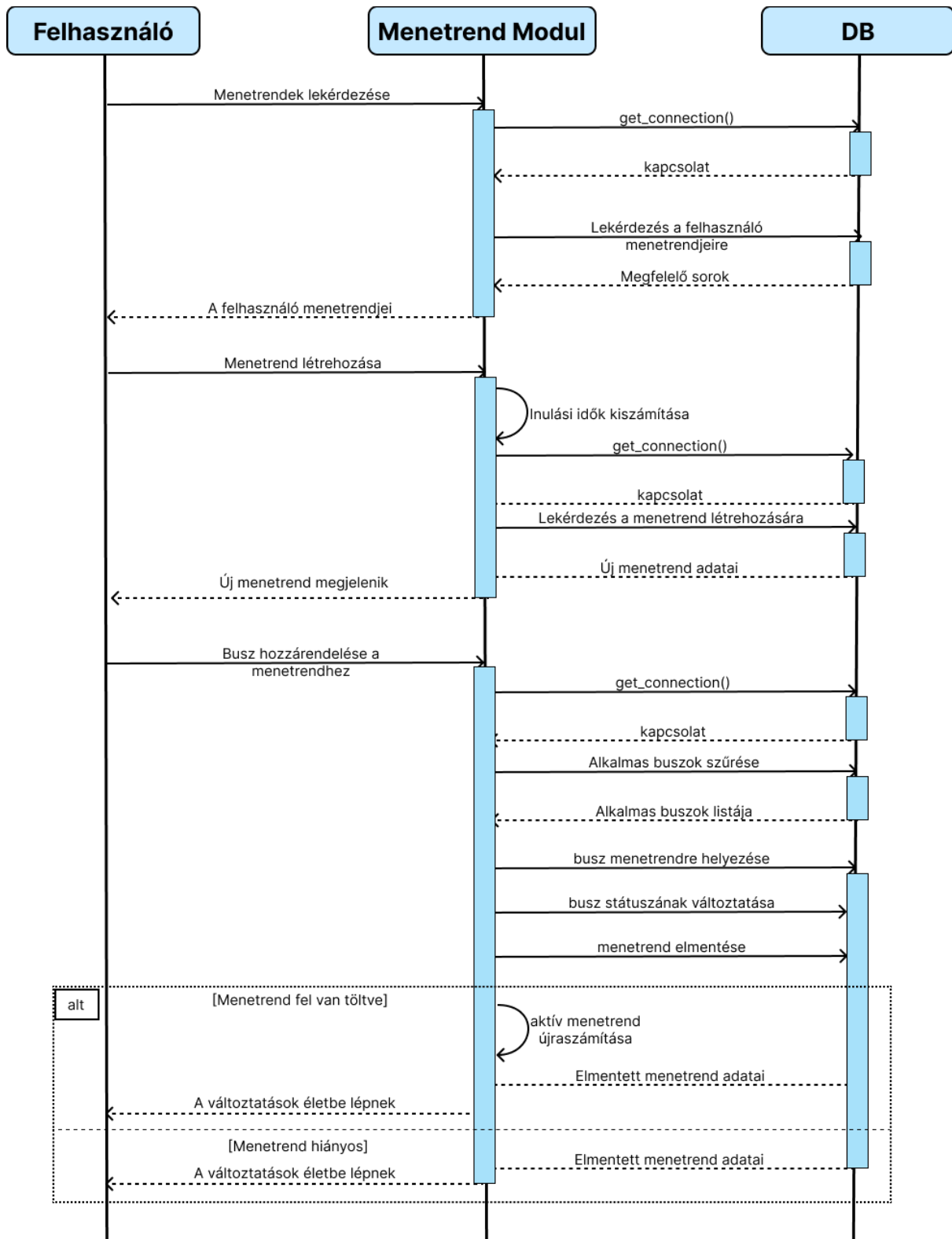
3.48. táblázat. Adott gyakorisághoz maximális licitet kiszámító függvény

payout_for_active_schedules		
Bemenet	Kimenet	Tevékenység
Aktuális idő	-	A függvény lebonyolítja az adott időpontban aktív menetrendekhez tartozó kifizetéseket és frissíti a buszok kilométer-állásait az aktuális időszávban.

3.49. táblázat. Az aktív menetrendeket kifizető függvény

save_manual_assignments		
Bemenet	Kimenet	Tevékenység
A menetrend azonosítója	Értesítő üzenet és státuszkód	A függvény ellenőrzi a beosztás esetén felmerülő hibákat, majd elmenti a felhasználó által megadott buszbeosztásokat az adott menetrendhez és ezeket az adatokat rögzíti a buszbeosztásokat tartalmazó táblába az adatbázis rétegben. Ezen felül frissíti a buszok megfelelő adatait a menetrend függvényében.
Adott felhasználónév		
A beosztott buszok		

3.50. táblázat. A buszbeosztásokat elmentő függvény



3.16. ábra. A menetrend modul szekvencia-diagramja

## 3.6 Tesztelés

A tesztelés keretein belül a backend réteg egyes moduljai kerülnek vizsgálat alá. Minden szolgáltatás függvényenként külön-külön tesztesetekkel van ellátva, értékelve a kedvező és kedvezőtlen kimeneteket is.

### 3.6.1 A busz modul tesztelése

A buszokért felelős modul tesztelése négy függvény helyes működésének ellenőrzését foglalja magába. Sorra kerül a buszok listázásának, létrehozásának, módosításának és kedvenc státuszuk változtatásának kiértékelése.

A *test\_get\_buses\_for\_user* eset a felhasználók buszainak listázását teszteli, tartalmaz egy esetet egy létező és egy nem létező felhasználó buszainak lekérdezésére, ahol a létező felhasználó esetén egy adott sor kiválasztását, nem létező felhasználó esetén pedig üres adatot vár.

A *test\_create\_bus* eset egy busz létrehozását teszteli, ahol egy sikeres új busz létrehozását, illetve egy hiányos adathalmaz esetében sikertelen műveletet szimulál.

A *test\_update\_bus* eset egy busz adatainak módosítását teszteli, ahol két esetet vizsgál: egy sikeres frissítést, és egy sikertelen frissítést, egy olyan rendszámmal, ami nem létezik.

A *test\_toggle\_favourite* eset a kedvenc státusz módosítását teszteli, vizsgál egy sikeresen végrehajtott módosítást, illetve egy olyan helyzetet, ahol a busz nem a megadott felhasználó tulajdona, azaz sikertelen műveletet.

### 3.6.2 A garázs modul tesztelése

A garázs modul két függvénye kerül tesztelés alá, a *test\_unlock\_garage* eset a garázs feloldását vizsgálja tüzetesen, lefedve a sikeres elsődleges, illetve fizetésen keresztüli feloldást, ezen felül pedig ellenőrzi annak sikertelenségét hiányos egyenleg vagy már korábban feloldott garázs esetébrn.

A *test\_list\_garages\_for\_user* pedig a felhasználó számára elérhető garázsok lekérdezését ellenőrzi.

### 3.6.3 A műszaki hiba modul tesztelése

A hibákat érintő tesztesetek a listázási, létrehozási és törlési funkciók megfelelő működését vizsgálják.



A *test\_list\_issues\_by\_bus* eset a buszokhoz tartozó műszaki hibák listázását vizsgálja, a *test\_list\_issues\_for\_user* eset pedig a felhasználó által birtokolt buszok problémáinak listázását ellenőrzi.

A létrehozás és az eltörlés megvalósítását a *test\_create\_issue* eset ellenőrzi a hiba jelentését tekintve, a *test\_remove\_issue* eset pedig markánsan vizsgálja az ertörlés különböző körülményeit, többek között a nem létező hiba, illetve a buszok utolsó és nem utolsó hibája eltávolításának következményeit.

#### 3.6.4 A vonal modul tesztelése

Mivel ez az egyik legkisebb modul, itt csak két függvényt ellenőriz a tesztelési terv, az egyik eset a *test\_list\_lines*, ami az összes vonal listázását szimulálja, a másik eset pedig a *test\_get\_line*, ami egy létező és egy nem létező vonal adatainak lekérdezését ellenőrzi, az előbbi esetben adatok visszatérésével, az utóbbiban pedig azok visszatérésének hiányával.

#### 3.6.5 A piac tesztelése

A piac modul megfelelő működésének ellenőrzése az egyik legextenzívebb tesztelési folyamat. Itt háromféle vásárlást is ellenőrizni kell, ugyanis az alkalmazás eredeti tervei szerint a buszok értékesítése azonnali lebonyolítású, kötött árú folyamat volt, teljesen más logikával, és csak később lett megvalósítva a hirdetés-alapú adásvétel, ebből kifolyólag pedig külön megvalósítás szolgál a buszok újonnan rendelésére, a mások hirdetésén keresztül vásárlásra, illetve az eredeti megoldásra, ami így a kész alkalmazásban egy “legacy” opció, ahol régiségeket lehet vásárolni a rendszertől. A tesztelés mindhárom fenti esetet lefedi, illetve kiterjed a listázásokra, hirdetések létrehozásának és lemondásának vizsgálatára is.

Az eredeti megvalósítás alapú buszvásárlást a *test\_list\_market\_buses* eset vizsgálja, ami az ebbe a kategóriába tartozó buszokat listázza a piacra, illetve a *test\_buy\_bus* eset ami vizsgál sikeres, illetve nem elérhető busz, nem létező felhasználó vagy elégtelen egyenleg okokból kifolyólag sikertelen vásárlást.

Az új buszok vásárlásának tesztelésére a *test\_purchase\_new\_bus* eset szolgál, ami lefedi a sikeres vásárlást, avagy a következő okokból sikertelen eseteket: nem létező modell, érvénytelen rendszám, nem feloldott garázs, foglalt rendszám, nem létező felhasználó, elégtelen egyenleg.

A hirdetések alapján folyósított adásvételt a *test\_purchase\_listing* eset vizsgálja, ami lefedi a sikeres vásárlást, illetve amennyiben sikertelen, a következő hibás eseteket: nem elérhető hirdetés, saját hirdetés, eladó-tulajdonos eltérés,

elégtelen egyenleg, és végül az az eset, amikor a felhasználónak nincs feloldva célgarázs.

Magukra a hirdetésekre további két eset, a *test\_create\_listing* és a *test\_cancel\_listing* figyel. Az előbbi a sikeres, avagy amennyiben nem a felhasználó a tulajdonos, vagy pedig már létezik hirdetés az adott buszra, sikertelen műveletet szimulál, az utóbbi pedig a hirdetésnek lemondását ellenőrzi. Amennyiben a hirdetés nem található, vagy a felhasználó nem a hirdető, a lemondás sikertelen, ha a kritériumok megfelelőek, akkor sikeres.

#### 3.6.6 A menetrendek tesztelése

A menetrend modul helyes működésének ellenőrzése az alkalmazás egyik legösszetettebb tesztelési folyamata, mivel ez a rendszer felel a járatok időbeosztásáért, a buszok hozzárendeléséért, illetve a vonalakhoz tartozó ütemezések automatikus kiválasztásáért. A tesztelés során vizsgálatra kerül a menetrendi időpontok helyes generálása, a buszok kiosztásának logikája, a menetrendekhez kapcsolódó hozzárendelések ellenőrzése, valamint a különböző ajánlatok közötti döntéshozatal is.

A *test\_generate\_slots* eset a rendszer által generált időintervallumokat vizsgálja. Ellenőrzi, hogy a megadott kezdő- és végidőpont, valamint az intervallumköz alapján a szolgáltatás helyesen állítja-e elő az időpontokat, és az elvárt formátumban tér-e vissza a létrehozott listával.

A *test\_plan\_buses\_for\_schedule* eset a buszok menetrendhez történő hozzárendelésének működését teszteli. Ennek során a szolgáltatás egy adott menetrend adatai alapján kiszámítja a szükséges időpontokat, meghatározza a beosztandó buszok számát, és létrehozza a hozzárendeléseket. A teszt lefedi a sikeres ütemezést, és ellenőrzi, hogy a visszaadott terv adatai helyesek-e.

A *test\_has\_all\_assignments* eset célja annak ellenőrzése, hogy az adott menetrendhez minden szükséges busz-hozzárendelés megtörtént-e. A vizsgálat kiterjed a menetrendhez kapcsolódó bejegyzések számának ellenőrzésére, és igazolja, hogy a szolgáltatás helyesen azonosítja a teljes, illetve a hiányos állapotokat.

A *test\_select\_winner\_for\_line\_and\_frame* eset a menetrendi ajánlatok közötti választás logikáját ellenőrzi, bővebben azt, hogy ugyanarra a vonalra és időszakra vonatkozó ajánlat közül a legkedvezőbb legyen kiválasztva, továbbá, hogy frissülnek-e az adatbázisban az érintett rekordok.

#### 3.6.7 A felhasználók tesztelése

A felhasználókat kezelő modulnak csak két függvénye kerül tesztelésre. A `test_create_user` szimulálja a felhasználó létrehozását, a `test_get_user_by_username` pedig egy felhasználó lekérdezésének folyamatát ellenőrzi, egy esetben sikeres, egyben pedig egy nem létező felhasználóra sikertelen lekérdezéssel.

#### 3.6.8 A tesztelés lebonyolítása

A tesztelés végrehajtásához az alábbi projekt struktúrát kell replikálni, hogy a unittest fel tudja fedezni a teszteseteket:

```
project/
├── flask_app/
│   ├── db/
│   ├── routes/
│   ├── services/
│   └── tests/
└── frontend/
```

Ezt követően a tesztek lefuttatásához a `flask_app` mappából, a következő parancsokat kell lefuttatni:

- `source .venv/bin/activate`
- `python -m unittest discover -s tests -p "test_*.py" -v`

## Összegzés

Ez a projekt egy hozzám nagyon közeli témához kapcsolódik. Mindig is nagyon érdekesnek tartottam a tömegközlekedési infrastruktúra működését, és mivel Budapesten ezt a feladatot főleg autóbuszok látják el, az itteni életben ez a tere a közösségi közlekedésnek az, ami a legkomplexebb és érdekesebb.

Amikor az ember már nem csak utazik, hanem megfigyel, egy idő után felfedezi azt, hogy melyik vonalon milyen típusú kocsik mozognak, akár melyik törzsrendszámok, azután azt, hogy melyik jármű melyik garázsból jön, azt, hogy melyik vonalat melyik garázs látja el. Azután felfedezi azt is, hogy hogyan valósul meg egy menetrend, hogy merre fut a garázsmenet, hogy ha pótlás van, akkor a járművek alapján következtetni tud, hogy melyik vonalról lettek lecsípve a buszok.

Ez az érdeklődés vitt bele engem is a budapesti infrastruktúra világába, ahol a mai napig minden utazásom egy új kaland, új információkkal. Innen eredt az ötlet is, hogy ezt a világot egy sajátos interpretációm szerint replikáljam, és ezen keresztül mélyebbre tudjak merülni az intrikált rendszerekbe, bele tudjak gondolni, hogy hogyan zajlik ennek a városi szolgáltatásnak a kivitelezése.

Ugyan az applikáció, amit építettem, nem feltétlenül kellően extenzív ahhoz, hogy szakmai feladatokat lásson el, vagy helyettesítsen, viszont azoknak, akik hozzám hasonlóan oda vannak a budapesti buszokért, kellemes játszótér lehet arra, hogy közelebb kerüljenek az érdeklődésük kibontakozásához.

## Köszönetnyilvánítás

Ezúttal szeretném megköszönni a konzulensemnek, Pintér Balázsnak a közös munkát, és azt, hogy vállalta ezt a szerepet és lehetőséget nyújtott nekem arra, hogy a saját témámat és ötletemet hozzam napvilágra ennek a szakdolgozatnak a keretei között.

Továbbá külön köszönetet szeretnék nyilvánítani Riba Gergelynek, aki régi barátom és a BKK menetrendjeinek felülvizsgálatával, továbbá járművezetőként szolgálja a város lakosait. Ő volt az a személy az életemben, aki felnyitotta a szemem a buszok világára, és elmélyítette a lelkesedésemet ezen infrastruktúrának bravúros megoldásai felé. Tapasztalatai és szaktudása sok olyan információval látott el, ami ebben a dolgozatban is hasznossá vált.

## Táblázatok jegyzéke

3.1. A busz vásárlásának követelményleírása .....	19
3.2. A busz eladásának követelményleírása .....	19
3.3. A busz kezelésének követelményleírása .....	20
3.4. A műszaki hibák kezelésének követelményleírása .....	20
3.5. A garázs kezelésének követelményleírása .....	20
3.6. A menetrend-szerkesztés követelményleírása .....	21
3.7. A busz tábla .....	32
3.8. A garázs tábla .....	32
3.9. A vonal tábla .....	32
3.10. A műszaki hibákat tartalmazó tábla .....	33
3.11. A menetrendeket tartalmazó tábla .....	33
3.12. A felhasználókat tároló tábla .....	33
3.13. A hirdetéseket tartalmazó tábla .....	34
3.14. A felhasználók és garázsok birtokos viszonyát tároló tábla .....	34
3.15. A menetrendek beosztásait tároló tábla .....	34
3.16. A buszokat lekérdező függvény .....	35
3.17. A buszokat létrehozó függvény .....	35
3.18. A buszokat módosító függvény .....	35
3.19. A kedvenc jelzőt változtató függvény .....	36
3.20. a garázsokat lekérdező függvény .....	37
3.21. a garázst feloldó függvény .....	37
3.22. A vonalakat lekérdező függvény .....	38
3.23. Egy vonalat lekérdező függvény .....	38
3.24. Adott busz műszaki hibáit lekérdező függvény .....	39

3.25. A felhasználó műszaki hibáit lekérdező függvény .....	39
3.26. Az összes műszaki hibát lekérdező függvény .....	39
3.27. Műszaki hibát létrehozó függvény .....	39
3.28. Műszaki hibát törölő függvény .....	39
3.29. A piac buszait lekérdező függvény .....	41
3.30. Az újonnan elérhető modelleket lekérdező függvény .....	41
3.31. A piaci hirdetéseket lekérdező függvény .....	41
3.32. Új hirdetést létrehozó függvény .....	41
3.33. Hirdetést lemondó függvény .....	42
3.34. Hirdetés megvásárlását megvalósító függvény .....	42
3.35. Busz megvásárlását megvalósító függvény .....	42
3.36. Új busz megvásárlását megvalósító függvény .....	42
3.37. Busz eladását megvalósító függvény .....	42
3.38. Adott menetrendet lekérdező függvény .....	44
3.39. Felhasználó menetrendjeit lekérdező függvény .....	44
3.40. Menetrendet létrehozó függvény .....	44
3.41. Menetrendet törölő függvény .....	44
3.42. Adott vonalhoz készített menetrendeket lekérdező függvény .....	44
3.43. Adott vonal győztes menetrendjeit kiszámító függvény .....	45
3.44. Adott vonal adott időszámban győztes menetrendet kiszámító függvény .....	45
3.45. Függvény, ami megvizsgálja hogy adott menetrend fel van-e töltve .....	45
3.46. Az indulási időket kiszámító függvény .....	45
3.47. Az indulásokhoz buszbeosztást kiszámító függvény .....	45
3.48. Adott gyakorisághoz maximális licitet kiszámító függvény .....	46
3.49. Az aktív menetrendeket kifizető függvény .....	46
3.50. A buszbeosztásokat elmentő függvény .....	46

## Ábrák jegyzéke

2.1. Bejelentkezési ablak .....	7
2.2. Új busz vásárlása .....	8
2.3. Használt busz vásárlása .....	8
2.4. Vásárlás megerősítése .....	8
2.5. Vásárlással kapcsolatos hibaüzenetek .....	9
2.6. Busz feltétele a piacra .....	9
2.7. Saját hirdetés lemondása .....	9
2.8. Buszok kezelése .....	10
2.9. Feloldott garázs állapota .....	11
2.10. Feloldatlan garázs .....	11
2.11. Garázs megvásárlása .....	11
2.12. Garázs sikertelen vásárlása .....	11
2.13. Műszaki hiba jelentése .....	12
2.14. Műszaki hiba .....	12
2.15. Műszaki hiba javításának megerősítése .....	12
2.16. Műszaki hiba javítás alatt .....	12
2.17. Javítás megszakítása .....	13
2.18. Menetrend létrehozása .....	13
2.19. Busz hozzárendelése a menetrendhez .....	14
2.20. Menetrend aktív státuszban .....	14
2.21. Menetrend törlése .....	14
2.22. Vonal ellátás nélkül .....	15
2.23. vonal ellátva .....	15
2.24. Menü .....	15
2.25. az egyenleg és kijelentkezés .....	15
3.2. A menü terve .....	22



3.3. A főoldal drótvázterve .....	23
3.4. A piac oldal drótvázterve .....	24
3.5. A műszaki hibák oldalának drótvázterve .....	25
3.6. A vonalak oldal drótvázterve .....	26
3.7. A garázsok oldalának drótvázterve .....	27
3.8. A menetrendek oldalának drótvázterve .....	28
3.9. A rendszer architektúrája nagy vonalakban .....	30
3.10. Az adatbázis diagramja .....	31
3.11. A busz modul szekvencia-diagramja .....	36
3.12. A garázs modul szekvencia-diagramja .....	37
3.13. A vonal modul szekvencia-diagramja .....	38
3.14. A műszaki hibák moduljának szekvencia-diagramja .....	40
3.15. A piac modul szekvencia-diagramja .....	43
3.16. A menetrend modul szekvencia-diagramja .....	47