

# **A Szállítók**

## **Témalabor dokumentáció**

Félév: 2022 őszi



## **Delivery App**

Fekete Sámuel (GJ8J3A) – Backend és web frontend

Farkasházi Levente (HFDKFC) – Backend és web frontend

Raisz Olivér (MWAI9Y) – Android (Client)

Domonkos Ádám (CWGYWC) – Android (Supplier)

## Projekt bemutatása

A fullstack alkalmazás célja egy független, közösségi szállítmányozási platform létrehozása. Egy szállításra alkalmas járművel rendelkező felhasználó el tud vállalni különböző szállítási munkákat. Azok a felhasználók, akik el szeretnének szállítani valamit, de nincs erre megfelelő járművük, fel tudnak adni hirdetéseket. Az alkalmazást adatokkal a felhasználók tevékenysége tölti meg, teljesen közösségi alapon működik. A szállítók és a kliensek külön alkalmazásokat használnak, így akár mindkét szerepet betölthetik. Egy admin felületről moderálhatóak a felhasználók és a hirdetések, valamint különböző hasznos statisztikák nézhetőek meg.

## Specifikáció – funkciók leírása

Az általános funkciókat minden alkalmazás megvalósítja. A Kliens (Client) alkalmazást használják azok a felhasználók, akik el akarnak szállíttatni valamit. A Szállító (Supplier) alkalmazást használják azok a felhasználók, akiknek van járművük és el tudják szállítani termékeket. Az Admin felületet használják az alkalmazást üzemeltetők, akik ellenőrzik és moderálják a megfelelő működést.

### Általános

- Google bejelentkezés implementálása
- Profil adatok a felhasználó Google fiókjából importálva: név, telefonszám, e-mail cím, profilkép

### Kliens Android app

- Új szállítandó termék felvétele (ár, méret, súly, időpont, honnan, hova, cím, leírás)
- A szállításra jelentkezett felhasználók listázása és bírálása (elutasíthat vagy elfogadhat egy jelentkező szállítót)

### Szállító Android app

- Profil megtekintése
- Jármű felvétele (típus, évjárat, kapacitás)
- Listázza a lehetséges munkákat, amikre lehet jelentkezni
- Listázza az aktuális munkákat (elbírálás alatt lévők, kezdésre vár, szállítás alatt)
- Listázza a korábbi munkák adatait
- Munkák megjelenítése térképen
- Munka befejezésekor kliens értékelése

### Admin weboldal

- Toplista a szállítókról
- Fuvarok megjelenítése, törlése
- Felhasználók kezelése (törlése, adminná tétele)
- Térképes nézet a munkákról

# Projekt elérhetősége

## Git repository

<https://github.com/Tschonti/a-szallitok>

## Szállító alkalmazás

<https://github.com/Tschonti/a-szallitok/tree/master/android/supplier/Apk>

## Kliens alkalmazás

[https://drive.google.com/drive/folders/1rZgQdJ\\_MTsqyoEgjTbAYQ19KrSHHMBoi?usp=sharing](https://drive.google.com/drive/folders/1rZgQdJ_MTsqyoEgjTbAYQ19KrSHHMBoi?usp=sharing)

## Backend

<https://a-szallitok-api.azurewebsites.net/>

## Web frontend

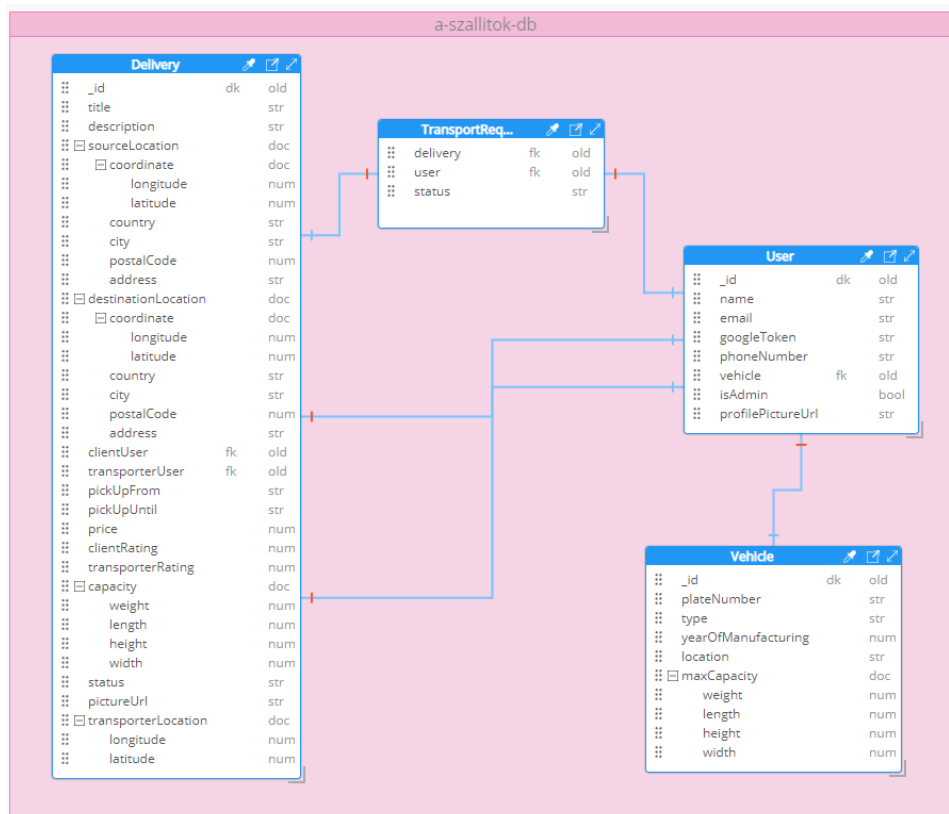
<https://victorious-pebble-08c001303.2.azurestaticapps.net/>

A Szállító és Kliens alkalmazások bármilyen Google fiókkal bejelentkezve használhatóak, a webes admin felület hasznos oldalait csak azok a felhasználók érhetik el, akik egy másik admin-tól admin jogot kaptak.

# Projekt felépítése

## Adatbázis

A projekt minden adata egy MongoDB dokumentum alapú adatbázisban tárolódik, tehát mindkét alkalmazás és a webes frontend is egy, közös helyről szerzi az adatokat. Így a felhasználók is közősek az alkalmazások között, nincs semmi akadály a annak, hogy egy felhasználó kliens és fuvarozó is legyen, de ilyenkor is csak egyszer kerül be az adatbázisba. Az adatbázis gyűjteményei a következők: Users, Deliveries, Vehicles, TransportRequests. A Users gyűjteményben tárolódnak a felhasználók adatai, aminek nagy része a Google fiókból kerül importálásra az első Firebase-s bejelentkezéskor, de itt tárolódik még az az információ is, hogy az adott felhasználó admin-e. A Deliveries gyűjteményben tárolódnak a fuvarok összes adata, köztük a kliens és a szállító felhasználóra mutató referencia. A TransportRequest egy kapcsolótábla a Users és Deliveries között, itt tárolódik az az információ, hogy egy fuvarozó jelentkezett egy fuvarra és hogy mi a státusza ennek kérésnek (elfogadásra vár, elfogadva, elutasítva). A Vehicles táblában pedig a fuvarozók járművei tárolódnak. Az adatbázis egy ingyenes MongoDB Atlas példányon fut.



## Backend

Az alkalmazás backendje egy Node JS szervert, Express keretrendszerrel. Feladata a három alkalmazás kéréseinek kiszolgálása és a kommunikáció az adatbázissal. A frontendekkel való kommunikációhoz REST-et használ, a backend API-jának az OpenAPI leírása [ezen a linken](#) megtalálható. A felhasználók azonosításához a Firebase JWT tokenjét használja, azaz minden frontend alkalmazásnak egy HTTP headerben küldenie kell a Firebase-től kapott, a jelenlegi bejelentkezett felhasználót azonosító token. A backend ezt a token-t elküldi a Firebase-nek, ami ez alapján visszaküldi a felhasználó alapadatait, és innentől a backend már be tudja azonosítani a felhasználót.

Az adatbázissal való kommunikációt pedig a Mongoose könyvtár segíti, ami biztosítja, hogy az adatbázis egy sémát követ, valamint könnyen használható metódusokat ad az adatok beszűrésére, lekérésére, szerkesztésére és törlésére.

A szervert egy ingyenes Azure App Service-ben fut, ezért az első betöltés eltarthat egy 20-30 másodpercig, ha az elmúlt időszakban senki nem használta.

## Web frontend

A webes admin felület egy React Single Page Application. A Firebase szolgáltatásain keresztül bejelentkezteti a felhasználót, majd a backend-től elkéri, hogy az adott felhasználó admin-e. Ha nem, akkor semmilyen tartalmat nem jelenít meg a számára. Bejelentkezés után egy Context-ben tárolja a felhasználó információit, így azt nem kell minden egyes oldal betöltésekor újra lekérni.

Az alkalmazást az Azure Static Web Apps szolgáltatással tettük publikusan elérhetővé.

## **Android**

ViewModel architektúra szerint készült. Az adatok elérése Repository mintát használva történik.

A bejelentkezést a Firebase Authentication szolgáltatásával végezzük, bármilyen Google fiókkal be lehet jelentkezni.

Az alkalmazások Navigation Componenttel készültek, tehát az alkalmazás képernyőit nem Activity-k, hanem Fragmentek adják.

# Haladási napló

## Fekete Sámuel

### 4. hét

- Git repository elkészítése
- Backend és frontend projektek inicializálása
- Specifikáció kidolgozása közösen
- Frontend oldalak tervének elkészítése Balsamiq-ban Levivel

### 5. hét

- adatbázis sémájának elkészítése közösen
- webes admin felülethez az endpointok megtervezése Levivel

### 6. hét

- Adatbázis sémájának javítása közösen
- Backend router vázának elkészítése (első pár endpoint mintaként)
- Google Firebase bejelentkezés kutatás (befejezetlen)

### 7. hét

- Google Firebase bejelentkezés befejezése
- Mongoose sémák elkezdése Levivel
- Middleware-ek elkezdése Levivel
- Postman Team setup Levivel

### 8-9. hét

- Az Android alkalmazásokat kiszolgáló middlewarek elkészítése (Levivel megosztva)
- Tesztelés az Android alkalmazásokkal

### 10. hét

- Frontend és Backend projektek deployolása Azureba Levivel
- React frontend projektben az autentikáció átírása Context-es megoldásra, a routing inicializálása
- Az utolsó backend endpoint elkészítése Levivel

### 11. hét

- Frontend app megjelenésének váza: Navbar és Footer elkészítése, dizájnolása
- Deliveries oldal elkészítése

### 12. hét

- Frontend autentikáció fejlesztése
- Figyelmeztető modalkok a frontendre destruktív műveletek előtt
- Kaszkádos törlés backenden
- Térkép oldal elkészítése Levivel
- Dokumentáció írás

## **Domonkos Ádám**

### 4. hét

- Specifikáció kidolgozása közösen
- Supplier alkalmazás inicializálása
- Supplier alkalmazás tervének elkészítése Balsamiq-ban

### 5. hét

- adatbázis sémájának elkészítése közösen
- API interfészeinek megírása a Supplier alkalmazáshoz

### 6. hét

- Adatbázis sémájának javítása közösen
- API interfészének javítása
- Android project létrehozása
- Főbb nézetek létrehozása
- MVVM architektúra kialakítása
- Retrofit implementálása

### 7. hét

- Google Auth megvalósítása
- API hívások véglegesítése
- Teljes UI elkészítése

### 8-9. hét

- Google Maps integrálása
- Jármű hozzáadása/szerkesztése
- Munka elvállalása/befejezése
- Client app segítése

### 10. hét

- Transporter alkalmazás tracking implementálása
- Location update endpoint használata

### 11. hét

- ScrollRefresh a lista nézetekre
- Broadcast receiver bootolásra, így újraindul a LocationService a telefon újraindításakor

### 12.hét

- Supplier alkalmazás befejezése
- Dokumentáció írás
- Customer alkalmazás javítása

## **Raisz Olivér**

### 4. hét

- Specifikáció kidolgozása közösen
- Git repó klónozása
- Mockup elkészítése az ügyfelek alkalmazásához

### 5. hét

- adatbázis sémájának elkészítése közösen
- API interfészeinek megírása az ügyfelek alkalmazásához

### 6. hét

- Android project létrehozása
- API interfész átnézése
- Backendhez szükséges dolgok setupolása

### 7. hét

- Material UI navigation drawer implementálása
- retrofit client
- repository alapjai
- új fuvar felvitele form alapok material UI 3-mal
- api hívás teszt a user aktív fuvarjaira
- (app nem indul - debugolás)

### 8-9. hét

- fragmentek közti navigálással kapcsolatos hibák kijavítása
- új fuvar felvételéhez kapcsolódó fragmenthez logika elkészítése, összekötés a backenddel
- Google-ös login elkészítése
- Tesztelés a backenddel

### 10. hét

- új delivery form dummy adatok helyett valósakat küld a backendnek
- form validációja
- pending deliveryk listázása (wip)

### 11. hét

- pending deliveryk listázása
- adatok megjelenítése ezekhez a deliverykhez
- lehetőség van elfogadni/elutasítani a deliveryt (wip)

### 12. hét

- Alkalmazás befejezése
- Fuvarok elfogadása és/vagy elutasítása a backenddel összekötve
- Új fuvar létrehozásánál a helyválasztás implementálása
- dokumentumírás



## **Farkasházi Levente**

### 4. hét

- Discord szerver elkészítése
- Specifikáció kidolgozása közösen
- Frontend oldalak tervének elkészítése Balsamiq-ban Samuval

### 5. hét

- adatbázis sémájának elkészítése közösen
- webes admin felülethez az endpointok megtervezése Samuval

### 6. hét

- Adatbázis sémájának javítása közösen
- Backend routerben a maradék endpointok definiálása
- Minta entitások készítése, amit az endpointok visszaadhatnak amíg nincs adatbázis

### 7. hét

- Mongoose sémák elkezdése Samuval
- A maradék mongoose sémák befejezése
- Middleware-ek elkezdése Samuval
- Postman Team setup Samuval

### 8-9. hét

- Az Android alkalmazásokat kiszolgáló middlewarek elkészítése (Samuval megosztva)
- Tesztelés az Android alkalmazásokkal

### 10. hét

- Frontend és Backend projektek deployolása Azureba Samuval
- Az utolsó backend endpoint elkészítése Samuval
- Tracking support a backenden

### 11. hét

- Users oldal elkészítése
- Toplist oldal elkészítése

### 12. hét

- Térkép oldal elkészítése Samuval
- Dokumentáció írás

# Együtműködés

## Fekete Sámuel

A fő kommunikációs platformunk egy Discord szerver volt, ahol a projekt négy részének négy külön csatornája volt, így az információk később is visszakereshetők voltak. Verziókezelésre és kódmegosztásra GitHub-ot használtunk, de nem használtuk ki minden lehetőségét, még a Issue-kat se írtunk ki, mert egymástól egészen függetlenül kellett dolgoznunk, és hetente legalább egyszer személyesen is találkoztunk, ahol meg tudtuk beszélni, hogy ki mit csinál a héten. Összességében nem volt mindig a leghatékonyabb az együtműködés, de megoldottuk a problémákat.

## Domonkos Ádám

A fő kommunikációs csatornánk egy Discord szerver, illetve egy Messenger csoport volt. Messengeren beszéltük a személyesebb, szervezéshez szükséges dolgokat, míg Discordon a szakmaiabb információkat küldtük el, hogy később jobban megtaláljuk őket. Verziókezelésre GitHubot használtunk, de nem volt szükség a minden képességére, mert főként különálló részegységekkel foglalkoztunk. Személyesen soka találkoztunk és főleg ott beszéltük meg a teendőket. Szerintem a fapados technológiák ellenére jól össze tudtunk dolgozni.

## Raisz Olivér

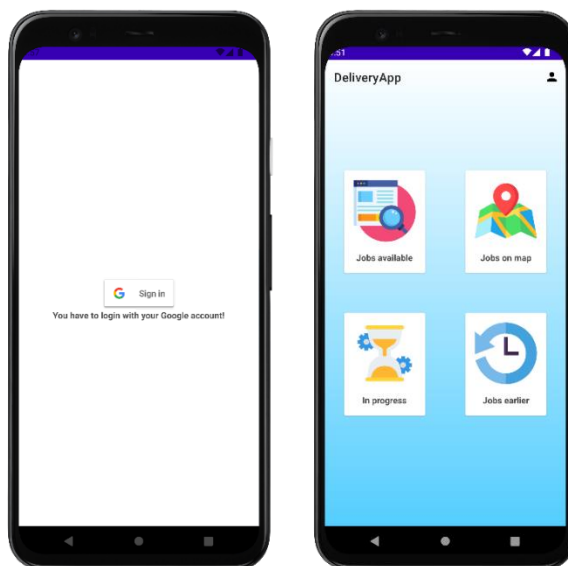
A jól strukturált git repó és a külön Discordos csatornák nagyon megkönnyítették a közös munkát. Számomra nagyon hasznos volt, hogy a backend swaggerje mindig up-to-date volt, így ritkán merült fel a backenddel kapcsolatban kérdésem. Sokat segített az is, hogy mindenki igyekezett lehetőségeihez mérten a leghamarabb válaszolni az üzenetekre.

## Farkasházi Levente

A munkát nagyban segítette a feladat elején létrehozott Discord szerver, ahol minden fontosabb információt vissza tudtunk nézni. A félév során heti szinten összeültünk élőben, ami nagyban segítette a feladatok letisztázását. A Discord szerver ekkor is nagyon jól jött, mert aki nem volt a közelben, az tudott ott csatlakozni. A folyamatos megbeszélések miatt nem volt olyan, hogy valami félrement volna. A messenger csoport nagyon jól működött a kevésbé szakmai kérdések esetén (pl. szervezés) és gyorsan megválaszolando hirtelen felmerült problémáknál. Verziókezelésre GitHub-ot használtunk, ami egészen jól működött. Nagyon megkönnyítette a munkát a Swagger is aminek köszönhetően mind mobilos, mind backend oldalról egyértelmű volt, hogy mi egy adott endpoint elvárt működése.

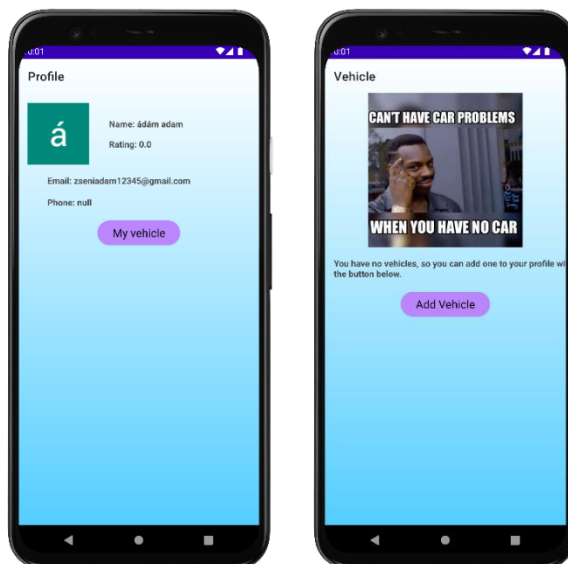
# Projekt bemutatása

## Szállító alkalmazás



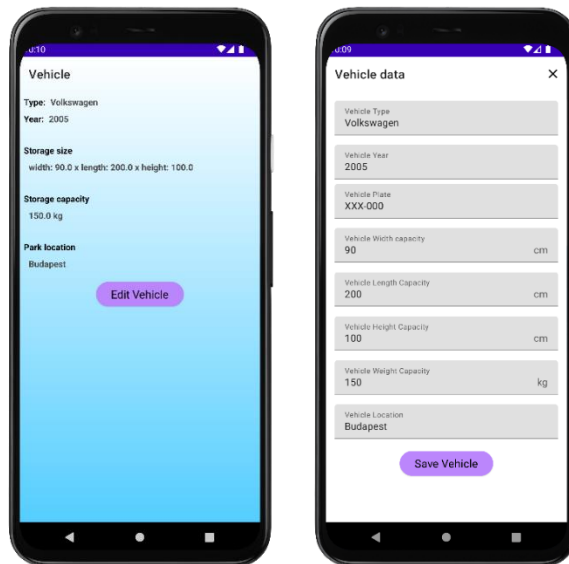
1. ábra: Bejelentkezés és főmenü képernyő

Az alkalmazásba a gomb megnyomása után lehet bejelentkezni bármilyen Google fiókkal. Sikeres bejelentkezés után az alkalmazás elküldi a backendnek a bejelentkezett felhasználó tokenjét, majd megnyílik a főmenü. Innen elérhetőek az alkalmazás fő funkciói.



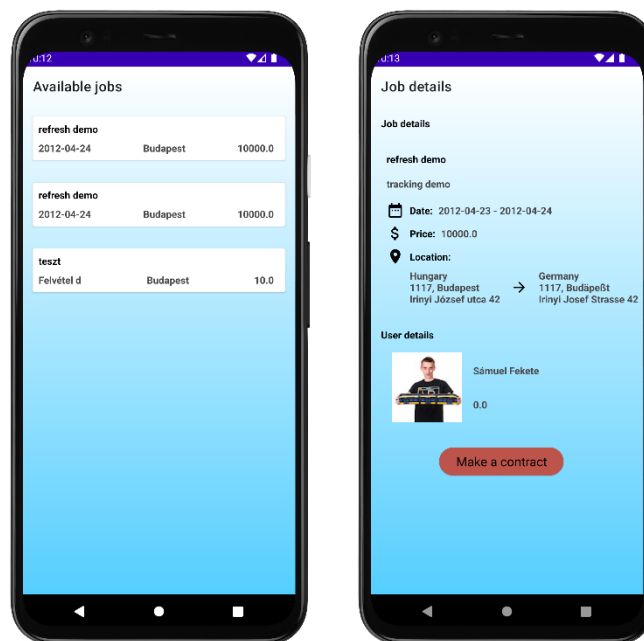
2. ábra: Profil és jármű hozzáadása képernyő

A menüben a jobb felső ikonra kattintva nézhető meg a felhasználó profilja. A név, e-mail cím, telefonszám, illetve profilkép a Google fiókjából szinkronizálódik. Az átlagos értékelést a backend számolja ki a korábbi munkákra kapott értékelések alapján. Itt lehet felvenni járművet, ha még nincs.



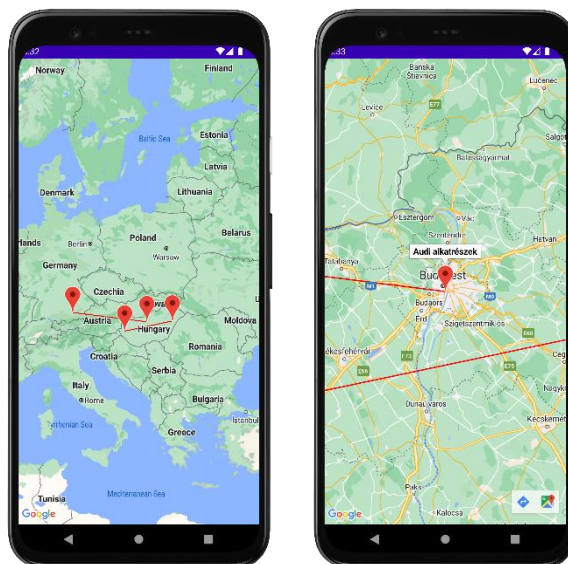
3. ábra: Jármű adatainak megjelenítése és beviteli form

Ha már meg van adva a járműnek az adatai, akkor az jelenik meg. A jobb oldali képen látható űrlappal lehet létrehozni, illetve módosítani a kocsit.



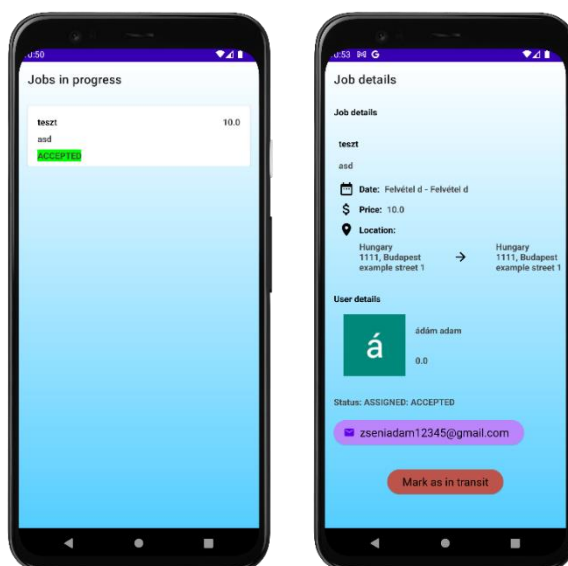
4. ábra: Elérhető fuvarokat megjelenítő képernyők

Az *Jobs available* menüpontban lehet válogatni az elérhető munkák közül. A *Make a contract* gombbal lehet jelentkezni.



5. ábra: Elérhető munkák térkép képernyője

A *Jobs on map* menüpontban meg lehet nézni az elérhető munkákat a térképre rajzolva. A kezdő- és végállomást egy vonal köti össze.



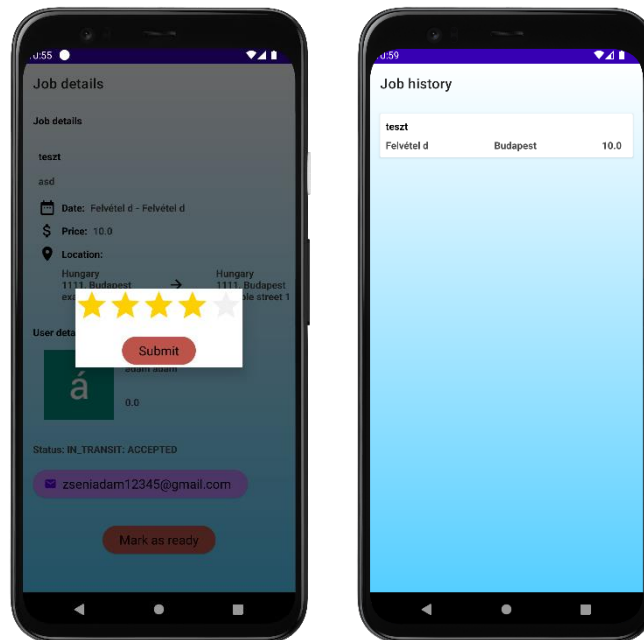
6. ábra: Folyamatban lévő fuvarokat megjelenítő képernyők

Az *In progress* menüpontban lehet megnézni az éppen folyamatban lévő fuvarokat. Ide kerülnek azok a munkák is, amikre jelentkezünk, de még elbírálásra várnak.

Lehetőségünk van felhívni, SMS-t, vagy e-mailt küldeni a feladónak, ehhez csak rá kell kattintani a megfelelő gombra. Ha valamelyik elérhetősége nincs megadva, akkor az nem jelenik meg.

Ha elfogadta a feladó a jelentkezésünket egy munkára, akkor megjelenik egy gomb, amivel elkezdhetjük a szállítást. Ilyenkor elindul egy service a háttérben, ami 30 másodpercenként elküldi az aktuális pozíciót a backend számára, így a feladó tudja követni, hogy merre jár a

terméke. Ha újraindul a telefon, akkor egy service ellenőrzi, hogy van-e folyamatban lévő fuvar és ennek megfelelően újra tudja indítani a helymeghatározást.



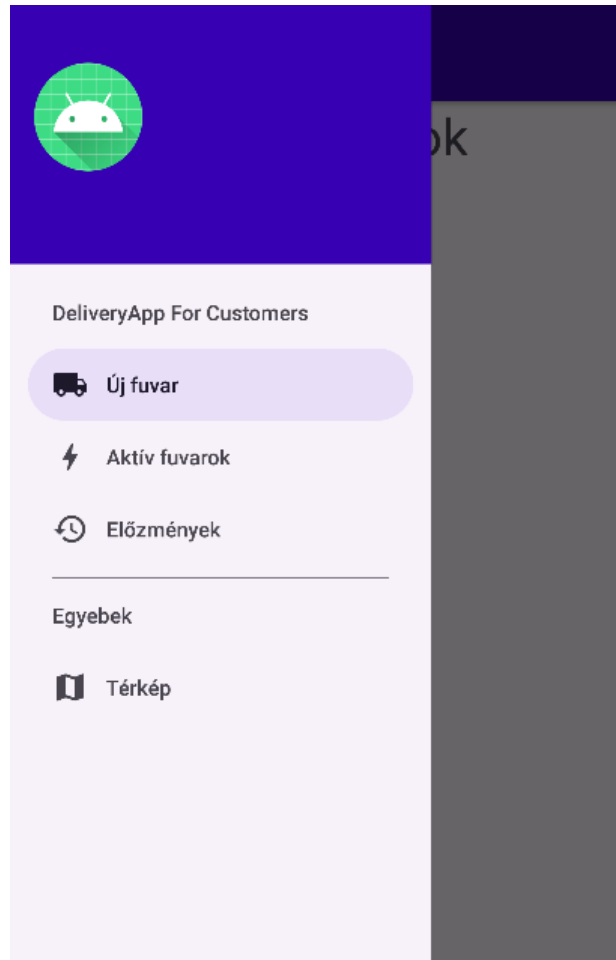
7. ábra: Értékelő és korábbi munkákat megjelenítő képernyők

Ha a szállító leszállította az árut, akkor ezt a gomb megnyomásával tudja jelezni. Ilyekor leáll a követés és a fuvar kész állapotba kerül. A fuvar végeztével lehetősége van a felhasználónak értékelni a klienst.

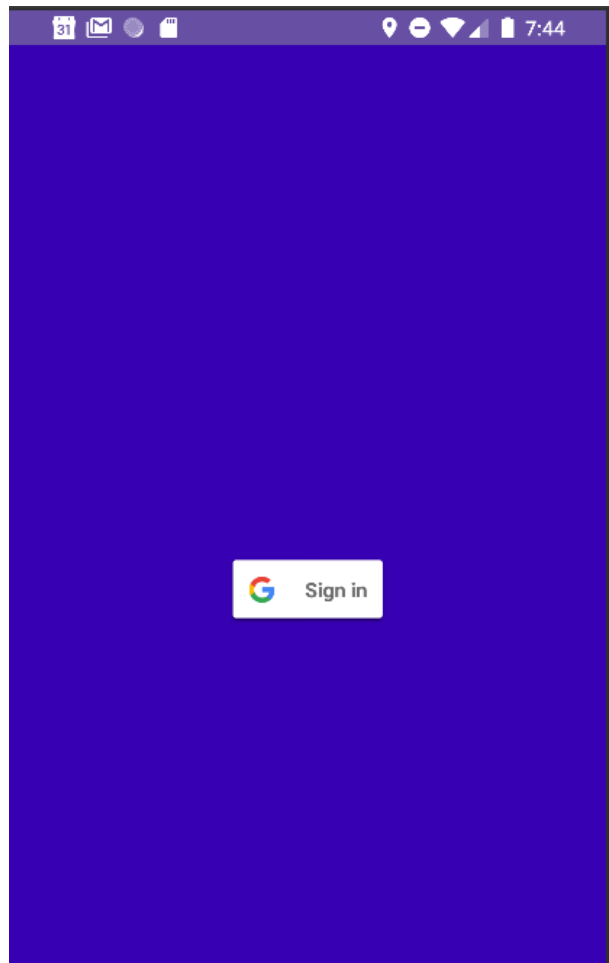
A *Jobs earlier* menüpontban lehet megnézni a korábbi, már befejezett fuvarokat. Ezeket csak meg lehet nézni, de nem lehet semmilyen további műveleteket végezni.

## Kliens alkalmazás

Az appba Google fiók használatával lehet bejelentkezni. Bármilyen fiók használható. Ha a felhasználó már be van jelentkezve, akkor a bejelentkező oldalról automatikusan továbblép az app az alkalmazásba. Az app funkciói között navigationdrawerrel (oldalsó menü) lehet navigálni.



8. ábra: Navigation Drawer









9. ábra: Bejelentkezés gomb

Az Új fuvar menüpontra nyomva egy form nyílik meg, ahol egy fuvar adatait lehet megadni, majd a backendre feltölteni. A sikeres feltöltést toast üzenet jelzi. Ha nem vagy rosszul adtuk meg egy mezőt (pl. a felvétel dátuma későbbi, mint a kézbesítés dátuma), akkor a form beküldés helyett a hibát toast üzenetben jelzi.



### Aktív fuvarok

	Gianna Wheeler 20 000 Ft	Elfogad Elutasít
	Cameron Shah 20 000 Ft	Elfogad Elutasít
	Ariel Garner 20 000 Ft	Elfogad Elutasít
	Michael Villegas 20 000 Ft	Elfogad Elutasít
	Tristan Dickson 20 000 Ft	Elfogad Elutasít
	Joseph Pineda 20 000 Ft	Elfogad Elutasít

### Szállítók

Termék neve

Maximum ár

A termék leírása

Súly

Szélesség

Magasság

Hossz

Felvétel dátuma

Kiszállítás dátuma

10. ábra: Aktív fuvarok képernyő és termék felvétele form

Az Aktív fuvarok menüpontra nyomva az egy felhasználóhoz tartozó aktív, azaz elfogadásra vagy elutasításra kész fuvarjelentkezőket látjuk egy listában. Amint elfogadunk egy adott fuvarhoz tartozó jelentkezőt, a többi automatikusan elutasításra kerül.

A Térkép nézet gombra nyomva pedig egy térkép jelenik meg, ahol nyomon tudjuk követni a fuvar pozícióját.

## Backend

<https://a-szallitok-api.azurewebsites.net/api-docs/>

user Informations about the users		
GET	/login	Log in the user identified in the JWT token
GET	/user/{ID}	Find user by ID
PUT	/user/{ID}	Update an existing users' profile
DELETE	/user/{ID}	Delete user by ID
PUT	/user/{ID}/promote	Promote user by id
GET	/user/toplist	Find user by ID
GET	/user/requestedJobs	All the jobs that the logged in user requested
GET	/user/jobRequests	All the users that requested a delivery from the logged in user

11. ábra : A felhasználóhoz kapcsolódó endpointok

vehicle Informations about supplier users' vehicles	
GET	/vehicle/{ID} Find a vehicle by ID
PUT	/vehicle/{ID} Update an existing vehicle
POST	/vehicle Add a new vehicle

12. ábra: A járműhöz kapcsolódó endpointok

delivery Informations about deliveries	
GET	/delivery Get all deliveries
POST	/delivery Add a new delivery
GET	/delivery/{ID} Find a delivery by ID
PUT	/delivery/{ID} Update an existing delivery
DELETE	/delivery/{ID} Delete a delivery by ID
POST	/delivery/{ID}/request Request to be the transporter of the delivery
PUT	/delivery/{ID}/reply Accept or decline the transport request
PUT	/delivery/{ID}/rateTransporter Rate the transporter of the delivery
PUT	/delivery/{ID}/rateClient Rate the client of the delivery
PUT	/delivery/{ID}/statusChange Update the status of the delivery

13. ábra: A fuvarokhoz kapcsolódó endpointok

## Web frontend

Sign in with Google

Bejelentkezés – Google-fiók - Google Chrome
accounts.google.com/o/oauth2/auth/identifier?response\_type=cod...
Bejelentkezés Google-fiókkal

### Bejelentkezés

Tovább ide: [the-transporters-fb.firebaseio.com](https://the-transporters-fb.firebaseio.com)

E-mail-cím vagy telefonszám

Nem tudja az e-mail-címét?

A folytatáshoz a Google megosztja az Ön nevét, e-mail-címét, nyelvi beállításait és profilképét a(z) the-transporters-fb.firebaseio.com alkalmazással.

Fiók létrehozása
Következő

magyar
Súgó
Adatvédelem
Feltételek

Created by: A szállítók

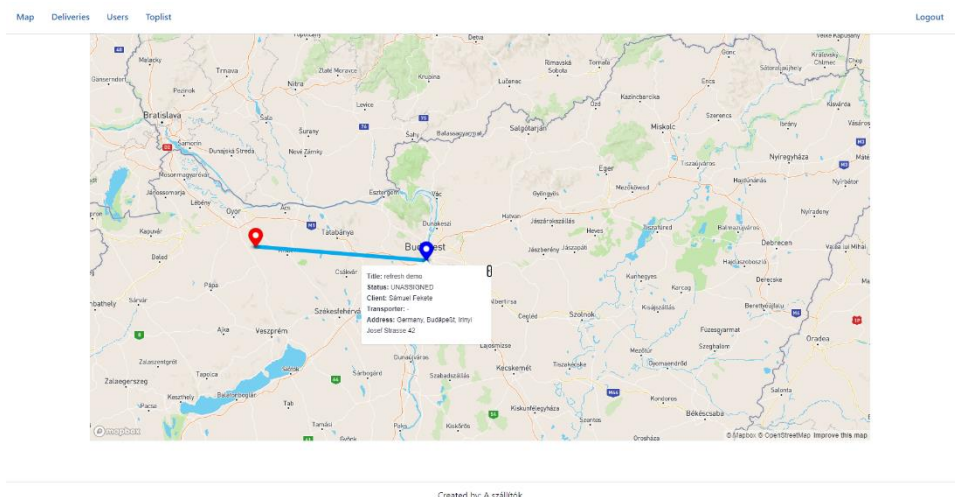
14. ábra: Bejelentkező képernyő

Az alkalmazásba egy gomb megnyomása után lehet bejelentkezni Google fiókkal. Az alkalmazás funkcióit csak az tudja igénybe venni, aki admin jogosultsággal rendelkezik. Ha valaki nem admin fiókkal lép be, akkor a következő oldalra kerül.

levente fa, you are not an admin!

15. ábra: Nincs jogosultság képernyő

Ezen az oldalon értesítjük a felhasználót a jogosultság hiányáról és itt ki tud lépni.



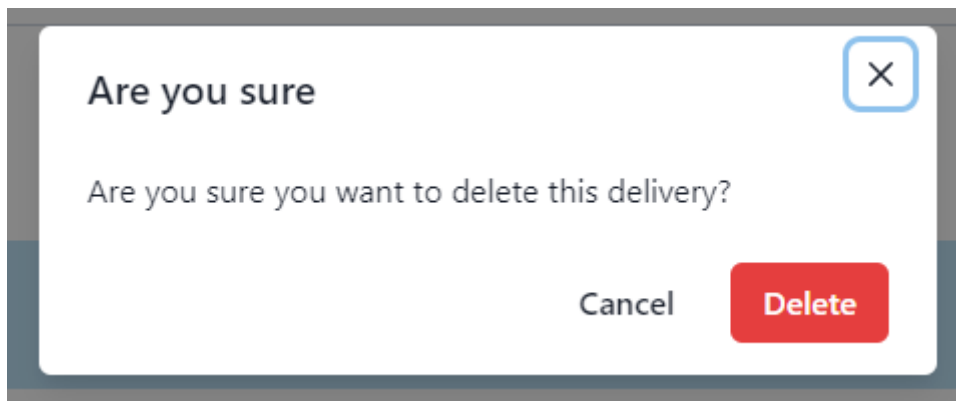
16. ábra: Térkép oldal

Az admin felhasználók bejelentkezés után a Map oldalra kerülnek, ahol megtekinthetik a fuvarokat egy térképen. A kék marker jelöli a fuvar kiinduló pontját, a piros pedig a végpontját. Az adott markerekre kattintva megjelennek a fuvar pontosabb információi és a marker által mutatott cím. A felhasználó az oldalak között egy navbar segítségével tud váltani. Illetve ki is tud lépni a fiókjából.

CLIENT	SOURCE	DESTINATION	STATUS	TRANSPORTER	ACTIONS
Sámuel Fekete	Hungary, Budapest	Germany, Budapest	UNASSIGNED	-	
Sámuel Fekete	Hungary, Budapest	Germany, Budapest	UNASSIGNED	-	
ádám adam	Hungary, Budapest	Hungary, Budapest	DELIVERED	Ádám Domonkos	

17. ábra: Fuvarok oldal

A deliveries oldalon megjelenik az összes fuvar, amiket igény esetén az admin törölni is tud. Törlés előtt az megerősítés szükséges az alább látható felugró ablakban, annak érdekében, hogy csökkentsük a véletlen törlés esélyét.



18. ábra: Felugró ablak destruktív művelet előtt

Map	Deliveries	Users	Toplist	Logout
Users				
NAME	EMAIL	PHONE NUMBER	ACTIONS	
Sámuel Fekete	feketesamu@gmail.com			
Olivér Ralsz	ralsozlover@gmail.com			
Ádám adam	zseniadam12345@gmail.com			
Ievente farkasházi	farkashazilevente1@gmail.com			
Ievente fa	farkashazilevente5@gmail.com			
Sámuel Fekete	fekete.samuelt@simonylbme.hu			
Ádám Domonkos	domonkosadam01@gmail.com			

User promoted!

19. ábra: Felhasználók oldal

A users oldalon megjelenik az összes felhasználó. A felhasználókat az admin tudja törölni, ilyenkor a felhasználók fuvarjai is törlődnek. Lehetőség van továbbá admin joggal felruházni felhasználókat.

Map	Deliveries	Users	Toplist	Logout
Toplist				
RANK	NAME	DELIVERIES COMPLETED	MONEY EARNED	
1	Ádám Domonkos	3	30000	
2	Ievente farkasházi	3	20000	
3	Sámuel Fekete	2	20000	

20. ábra: Felhasználói toplista

A toplist oldalon jelenik meg a top felhasználók listája. A felhasználók a fuvarokkal megszerzett pénzösszeg alapján vannak rangsorolva, ha ez egyezik, akkor a teljesített fuvarok száma alapján dől el, ki lesz előbb.

# Sprint eltérítés

## Feladat leírása

Amikor egy szállító megkezd a szállítást az átvett termékkel, lehessen valós időben nyomon követni, hogy merre tart.

## Feladat megvalósítása

A szállító alkalmazásban, amikor a felhasználó jelzi, hogy átvette az árut és megkezd a szállítást, akkor elindul a háttérben egy service, ami folyamatosan elküldi a szervernek egy újonnan létrehozott végpontra az aktuális pozíciót. A service a telefon újraindulása után is újraindul, ha van megkezdett szállítás. A kliens alkalmazásban egy térképen megjeleníthető az utolsó pozíció.

Hatékonyabb lett volna websocket alapon küldeni a folyamatosan érkező adatokat. Így, az új rest végponttal viszont kevés architektúrális módosításra volt szükség kliens és szerver oldalon is.

# Végső értékelés

## Feladatok megoszlása

### Csapatmunka megoszlása

Fekete Sámuel: 28%

Domonkos Ádám: 27%

Raisz Olivér: 20%

Farkasházi Levente: 25%

### Android szétválasztása

- Domonkos Ádám: Supplier alkalmazás
- Raisz Olivér: Kliens alkalmazás

### Backend szétválasztása

- Fekete Sámuel: Nekem már volt sok tapasztalatom Express és React alkalmazásokkal is, de korábban magamtól nem kezdtem még projektet nulláról, így általában én inicializáltam az alkalmazásokat, könyvtárakat, illetve én csináltam a Firebase-es bejelentkezést is, mert ilyet sem csináltam korábban. Én csináltam még az első middlewareket, majd az első frontend oldalt, és ez alapján segítettem Levinek, aki megcsinálta a többi middlewaret és oldalt.
- Farkasházi Levente: Én viszonylag kevés tapasztalattal rendelkezem. Minimális Express tapasztalatom volt az ebben a félévben teljesített vitmav42 (Szerver oldali JavaScript) tárgyam miatt, illetve TypeScriptben már fejlesztettem korábban is. A feladatok közösen megbeszéltük és Samu kezdte el csinálni őket, ami nagyon sokat segített a megértésükben. Általában a már kész részletekből merítve és kérdés esetén átbeszélve a dolgokat, már be tudtam fejezni a rám osztott részeket.

## Fekete Sámuel

Összeségében nagyon élveztem ezen a projekten dolgozni, érdekes volt egy ilyen sokszínű alkalmazást a nulláról megtervezni, majd lefejleszteni. Tetszett, hogy jól különválasztottuk az alkalmazás részeit, így tudtunk párhuzamosan haladni, de folyamatosan kommunikálnunk kellett egymással. Szívesen foglalkoztam volna többet is a projekttel, de sajnos a többi tárgy miatt nem maradt több idő rá, de úgy gondolom így is egy jól működő alkalmazást építettünk.

## Domonkos Ádám

Nagyon élveztem a közös munkát, nagyon jó csapatot alkottunk. Sikertől különálló egységekre osztani a projektet, így nagyon jó ütemben tudtunk haladni. Számomra a legérdekesebb része a fejlesztésnek az első szakasz volt, amikor együtt, közösen kellett kitalálni, hogy hogy tud a legjobban együttműködni a backend és a kliens alkalmazások. Sokszor újra kellett gondolnunk, és változtatni rajta, hogy valóban megfelelő legyen mindenki számára. Az Android alkalmazás írása közben nagyon sok újat tanulhattam. Érdekes volt látni, hogy egy ilyen nagyobb projektnél az architektúrát kellett először kitalálni és az alapján fejleszteni. Így, ha nagyobb módosítás is történt backend oldalon az alkalmazásban gyorsan le lehetett ezt követni. Örülök, hogy sikerült ilyen jól együtt dolgoznunk és egy működő projektet létrehozni.

## **Raisz Olivér**

A projekt elején én egy kicsit el voltam veszve, egyrészt nem volt még semmilyen projektem, ami architektúráisan ennyire komplex lett volna, másrészt nem értettem, mikor is fog megtérülni a rengeteg kezdeti tervezés és a rendszer összetettsége. Aztán ahogy haladtunk a projekttel, egyre inkább egyértelművé vált számomra, hogy mennyivel nehezebb lenne a munkánk most, ha nem fordítunk annyj időt és energiát a kezdeti tervezésre. Összességében tehát nekem nagyon tanulságos volt ez a tárgy.

## **Farkasházi Levente**

Alapvetően nagyon élveztem a közös munkát. A projekt elején nagy hangsúlyt fektettünk a részletes architektúrális tervezésre, ami később nagyon sokat segített és könnyen átszervezhetővé tette a projekt elemeinek működését. Én személy szerint kicsit féltem eleinte, hogy nem fogom tudni ennyi idő alatt elsajátítani a kellő tudást a feladatom elvégzésére, de hála Samu segítőkészségének és türelmének a feladat végére már egészen átláttam a projektet és a már működő komponensek alapján magabiztosan és gördülékenyen tudtam megvalósítani a még hiányzó részeket. A feladat nagyon érdekes volt és örültem volna, ha több időt tudtam volna vele foglalkozni. Örülök, hogy a félév során végig lelkesen és együttműködve tudtunk dolgozni rajta. Úgy érzem nagyon sokat tanultam belőle mind szakmailag, mind csapatos együttműködés terén.