



Rendszerfejlesztés

GKNB_INTM011

SZEAT-APP

Széchenyi Egyetem - Audi Telekocsi applikáció

Rendszerfelejtők

Győr, 2022

Tartalom

1. Projektszabályzat	1
1.1. Tagok listája	1
1.2. Kommunikáció és dokumentumkezelés	2
1.3. Jogok és kötelezettségek	2
1.4. Feladatkörök	6
1.5. Kompetencia mátrix	6
1.6. Feladatkörök bemutatása	8
1.7. Tevékenység-felelősség mátrix	9
1.8. Közös szótás (Glossary)	10
2. A kiválasztott probléma	13
2.1. Szöveges bemutatás	13
2.2. A probléma folyamatábrája	14
2.3. Halszálka diagram	15
2.4. SWOT analízis	16
3. Üzleti igény specifikáció	17
3.1. Célok	17
3.2. Stakeholderek	17
3.3. Rendszer tervezett működése folyamatábra	19
3.4. Rendszer tervezett működése - use case diagram	21
4. Követelményspecifikáció	21
4.1. Funkcionális követelmények	21
4.1.1. Jelentkezés az útra	22
4.1.2. Út visszamondása	23
4.1.3. Út meghirdetése	24
4.1.4. Út törlése	25
4.2. Nem funkcionális követelmények	26
4.2.1. Termék követelmények:	26
4.2.2. Szervezeti követelmények:	27
4.2.3. Külső követelmények:	27
4.3. Használhatóság	28
4.4. Megbízhatóság	28

4.5. Rendelkezésre állás.....	29
4.6. Teljesítmény.....	29
4.7. Támogatottság.....	30
5. Technikai specifikáció.....	31
5.1. Architektúra terv.....	31
5.2. Platform tulajdonságok és követelmények.....	32
5.3. Fejlesztői eszközök.....	33
6. Gazdasági számítások	34
6.1. Felmerülő költségek	34
6.2. TCO (Total Cost of Ownership).....	34
6.3. Tervezett bevételek	35
6.4. Fenntarthatóság növelése, költségek csökkentése.....	35
6.5. Jövőbeli bevételi lehetőségek	35
7. Adatbázis terv	36
7.1. Szöveges leírás	36
7.2. Táblák	36
7.3. Adatmodell	38
7.4. Egyed-kapcsolat diagram	39
7.5. Kulcsok és megszorítások	39
7.5.1. Elsődleges kulcsok	39
7.5.2. Másodlagos kulcsok	39
7.5.3. Megszorítások	40
7.6. Tárolt eljárások.....	40
7.7. Továbbfejlesztési ötletek	41
8. GUI terv	42

1. Projektszabályzat

Ezen projektszabályzat írásos keretek közt rögzíti a Rendszerfelejtők csapat féléves munkájának körülményeit ideértve:

- a csapattagok személyét, képességeit, jogait és kötelezettségeit,
- a csapattagok közötti kommunikáció és dokumentumkezelés módját és eszközeit,
- a feladatköröket és az ezekhez tartozó tevékenységeket,
- a csapat működési rendjét és az ehhez kapcsolódó általános szabályokat,
- a jutalmazás és büntetés lehetséges módjait és alkalmazásuk szabályait,
- és egy közös szótárat (Glossary), ami tartalmazza a projekt során felmerülő olyan kifejezéseket és szakszavakat, amelyek jelentése nem feltétlenül egyértelmű vagy ismert az informatikus társadalomban.

A projektszabályzat elkészítése és karbantartása a projektvezető kizárólagos felelőssége és feladata.

A projektszabályzat megtekintésére és módosítási javaslatok tételére minden csapattag jogosult.

1.1. Tagok listája

A csapat tagjai:

- Adorján András Bálint
- Bálint Ábel
- Hegedüs Róbert
- Jánoki Lilla
- Pintér Gábor
- Szakács Márk Milán

1.2. Kommunikáció és dokumentumkezelés

A csapat elsődleges kommunikációs felülete a **Messenger** alkalmazásban kifejezetten erre a célra létrehozott csoport. Itt történik a csapattagok közti alapvető szöveges kommunikáció.

A szóbeli kommunikáció alapvető platformja a **Discord** programban kifejezetten erre a célra létrehozott csoport.

Amennyiben a fent említett elsődleges kommunikációs platformok nem működnének, **Zoom** és **Teams** programok, illetve a személyes megbeszélés fogja ideiglenesen biztosítani a kommunikációt.

A csapattagok heti egy alkalommal, órarendjüktől és egyéb időbeosztásuktól függően, de általában hétfőn videókonferencián egyeztetnek, heti egy alkalommal pedig a gyakorlati óra keretein belül találkoznak személyesen is. Amennyiben a projekt megkívánja további videókonferenciákat és személyes találkozókat is egyeztethetnek.

Dokumentumok kezelése

A dokumentumok tárolási platformja a **GitHub**. A dokumentumokat a projektvezető és a titkár(ok) szerkeszti(k), viszont minden csapattag javasolhat változtatásokat.

1.3. Jogok és kötelezettségek

A projektvezető jogai és kötelezettségei

- Kommunikáció a gyakorlatvezető oktatóval.
- Kommunikáció az esetleges céges partnerrel.
- A csapattagok fegyelmezése.
- A csapattagok munkájának ellenőrzése.
- Jutalmak és büntetések kiszabása.
- A projektmunkáért kapott pontszám felosztása.
- Dokumentumok korlátozás nélküli szerkesztése.
- Projektszabályzat létrehozása, módosítása.
- A kommunikációs és dokumentumkezelési platformok rendben tartása.

Csapattagok jogai és kötelezettségei

- Személyes és videó megbeszéléseken való részvétel.
- Kiszabott feladatok teljesítése.
- Projektvezető és projektvezető-helyettes utasításainak teljesítése.
- Szabad véleménynyilvánítás és fellebbezés joga.

Működési rend, általános szabályok

- Fellebbezésre jogosult bármelyik tag, amennyiben teljesül a fellebbezés feltétele.
- Jutalmak és büntetések kiosztására csak a projektvezető jogosult.
- Bármelyik tag vállalhat és kaphat is a feladatkörén kívül eső feladatot.
- Minden megbeszélésről jegyzőkönyv készül.

Fellebbezés szabályai

- Fellebbezés feltételei:
 - A csapattag nem ért egyet a projektvezető döntésével.
 - A fellebbezőn kívül legalább 1 másik tag is egyetért a fellebbezéssel.
- Fellebbezés módja: A csapat szavazással dönti el, hogy a fellebbezés jogos-e. A szavazásban a projektvezető szavazata 2-őt ér, minden más tag szavazata 1-et.
- Jogos fellebbezés következménye: Ha a szavazás eredménye az lett, hogy a fellebbezés jogos, akkor a csapatnak ki kell találnia, egy mindenki számára előnyös megoldást. Amennyiben nem tudnak megegyezni a csapattagok, úgy a gyakorlatvezető tanár bevonásával kell végleges döntésre jutni és megoldani a problémát. A megoldás részleteit írásban rögzíteni kell, függetlenül attól, hogy a gyakorlatvezető tanár bevonásra került-e.

Jutalmazás szabályai

- A projektvezető bármely másik csapattagot részesítheti jutalomban.
- Jutalmazás lehetséges formái:
 - Szóbeli, írásbeli dicséret.
 - Egy üveg lekvár.
 - Egy doboz/üveg/korsó sör.
 - Plusz pont a projekt értékelésénél.
- Jutalomban részesülhet, aki:
 - Önként vállal el a feladatköréhez nem tartozó feladatot.
 - Konstruktív hozzászólásokat, javaslatokat tesz a projekttel kapcsolatban.
- Jutalmazás módja:
 - A projektvezető a csapattag munkássága alapján szóbeli vagy írásbeli dicséretben részesíti a csapattagot.
 - 2 szóbeli dicséret után írásbeli dicséretben részesül az adott csapattag.
 - 2 írásbeli dicséret után az adott csapattag egy üveg lekvár jutalomban részesül.
 - 4 írásbeli dicséret után az adott csapattag egy doboz/üveg/korsó sör jutalomban részesül.
 - 6 írásbeli dicséret után az adott csapattag 1 ponttal többet fog kapni a projekt értékelésénél.

1. táblázat - Jutalmak

Név	Adorján András Bálint	Bálint Ábel	Hegedüs Róbert	Jánoki Lilla	Pintér Gábor	Szakács Márk Milán
Jutalom típusa						
Szóbeli dicséret		X	X			
Írásbeli dicséret				X	X	
Lekvár						
Sör						
Plusz pont						

Büntetés szabályai

- A projektvezető bármely másik csapattagot részesítheti büntetésben.
- Büntetés lehetséges formái:
 - Szóbeli, írásbeli megrovás.
 - Egy doboz/üveg/korsó sör a projektvezetőnek.
 - Egy doboz/üveg/korsó sör a csapat összes tagjának.
 - Mínusz pont a projekt értékelésénél.
 - Kizárás a csapatból.
- Büntetésben részesülhet, aki:
 - Nem végzi el a megbeszélt határidőre a rá kiosztott feladatot.
 - Indok és előzetes egyeztetés nélkül nem jelenik meg a csapat megbeszélésén vagy a gyakorlati órán.
 - Trágár, becsmérő szavakkal illeti bármely csapattagot, hallgatótársát vagy a gyakorlatvezető tanárt.
- Büntetés módja:
 - A projektvezető a vétség súlyosságától függően szóbeli vagy írásbeli megrovásban részesíti a csapattagot.
 - 2 szóbeli megrovás után automatikusan írásbeli megrovásban részesül az adott csapattag.
 - 2 írásbeli megrovás után az adott csapattag köteles egy doboz/üveg/korsó sört köteles fizetni a projektvezetőnek.
 - 4 írásbeli megrovás után az adott csapattag köteles egy doboz/üveg/korsó sört köteles fizetni az összes csapattagnak.
 - 6 írásbeli megrovás után az adott csapattag 1 ponttal kevesebbet fog kapni a projekt értékelésénél.
 - 8 írásbeli megrovás után az adott csapattag 3 ponttal kevesebbet fog kapni a projekt értékelésénél.
 - 10 írásbeli megrovás után az adott csapattag azonnali kizárásra kerül a csapatból.

2. táblázat - Büntetések

Név	Adorján András Bálint	Bálint Ábel	Hegedüs Róbert	Jánoki Lilla	Pintér Gábor	Szakács Márk Milán
Jutalom típusa						
Szóbeli megrovás						
Írásbeli megrovás						
Sőr a projektvezetőnek						
Sőr az összes csapattagnak						
Mínusz pont						
Kizárás a csapatból						

1.4. Feladatkörök

A tagok az alábbi feladatköröket töltik be:

- Projektvezető
- Projektvezető-helyettes
- Titkár
- Programozó
- Adatbáziskezelő
- Grafikus

A feladatköröket a csapat egy közös megbeszélés során alakította ki a tantárggyal kapcsolatos addigi ismereteik és a projekt kapcsán várhatóan felmerülő feladatok alapján.

1.5. Kompetencia mátrix

A kompetencia mátrix alapvető tulajdonságokat, képességeket tartalmaz, amelyekre valószínűsíthetően szüksége lesz a csapatnak a projekt során. Minden csapattag esetében leolvasható, hogy rendelkezik-e az adott kompetenciával. A kompetencia meglétét az 'X' karakterek jelölik.

3. táblázat - Kompetencia mátrix a személyekkel

Név	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12	K13	K14
Adorján András Bálint	X	X	X	X		X	X	X	X	X	X		X	X
Bálint Ábel	X	X		X			X		X	X	X	X	X	
Hegedüs Róbert	X	X		X				X	X	X	X	X		
Jánoki Lilla	X	X	X	X			X	X	X	X	X		X	X
Pintér Gábor	X	X		X			X	X	X	X	X		X	X
Szakács Márk Milán	X	X		X	X	X	X	X	X	X	X	X		

4. táblázat - Kompetenciák megnevezése

K1	Adatbáziskezelési ismeretek
K2	Angol nyelv ismerete
K3	Fogalmazási készség
K4	GitHub ismeretek
K5	Grafikai ismeretek
K6	Képszerkesztési ismeretek
K7	Kommunikáció
K8	Kreativitás
K9	MS Office ismeretek
K10	Önfejlesztési készség
K11	Programozási ismeretek
K12	Rugalmasság
K13	Szervezési készség
K14	Vezetői képességek

5. táblázat - Feladatkörök megnevezéssel és a személyekkel

Projektvezető	Adorján András Bálint
Projektvezető-helyettes	Jánoki Lilla
Titkár	Jánoki Lilla, Bálint Ábel
Programozó	Bálint Ábel, Hegedüs Róbert, Pintér Gábor
Adatbáziskezelő	Pintér Gábor, Szakács Márk Milán
Grafikus	Adorján András Bálint, Szakács Márk Milán

1.6. Feladatkörök bemutatása

Projektvezető

- Kommunikáció a gyakorlatvezető oktatóval
- Kommunikáció az esetleges céges partnerrel
- A csapattagok fegyelmezése
- Jutalmak és büntetések kiszabása
- A projektmunkáért kapott pontszám felosztása
- A kommunikációs és dokumentumkezelési platformok rendben tartása
- Megbeszélések levezénylése

Projektvezető-helyettes

- A projektvezető feladatainak elvégzése, annak távolléte, esetleges munkaképtelensége esetén

Titkár

- A kommunikációs és dokumentumkezelési platformok rendben tartása
- A projekt dokumentálása
- Csapattagok munkáinak begyűjtése, rendszerezése és továbbítása a projektvezetőnek

Programozó

- A projekt során felmerülő programozási feladatok ellátása

Adatbáziskezelő

- A projekt során felmerülő adatbáziskezelési feladatok ellátása

Grafikus

- Látványtervek, folyamatábrák és GUI-k készítése

1.7. Tevékenység-felelősség mátrix

6. táblázat - Tevékenység-felelősség mátrix

Név	Adorján András Bálint	Bálint Ábel	Hegedüs Róbert	Jánoki Lilla	Pintér Gábor	Szakács Márk Milán
Tevékenység						
Kommunikáció	J			J		
Dokumentáció	V, J	V		V		
Prezentáció	V, J	V	V	V	V	V
Megbeszélések koordinálása	V					
Kapcsolattartás	V	I	I	I	I	I
Látványtervek készítése	V, J	I	I	I	I	V
Folyamatábrák készítése	V, J	I	I	I	I	V
GUI-k készítése	V, J	I	I	I	I	V
Adatbáziskezelés	J	I	I	I	V	V
Programozás	J	V	V	I	V	I
Projektszabályzat módosítása	V	I	I	I	I	I
Jegyzőkönyv írás	I			V, B		

V - Végrehajtási felelősség. Az érdekelt felelőssége a munka elvégzése. Nem feltétlenül hoz döntéseket, de a csoportot arra sarkallja, hogy mindig időben hozzák meg a döntéseket.

J - Jóváhagyási jogkör. Végleges hozzájárulás a tevékenység kimenetelének elfogadásához. Döntéseket hoz.

B - Meg kell beszélni. A munkavégzés során az érdekelt ad információt. Nem hoz döntéseket, de a döntések előtt tanácsot kérnek tőle.

I - Informálni kell, ha döntés született. Mindig naprakész szeretne lenni ennek a tevékenységnek az előrehaladásáról.

1.8. Közös szótár (Glossary)

Stakeholder

Az üzleti igényspecifikáció a résztvevője.

Appsofőr

A SZEAT-APP sofőrje.

Apputas

A SZEAT-APP utasa.

Use-Case diagram

Magyarul használati eset diagramként is ismert, a felhasználó és a rendszer közti lehetséges interakciók grafikus ábrázolására használt módszer. A Use-Case diagram különböző használati eseteket és különböző típusú felhasználókat mutat be, gyakran más típusú diagramok is kísérik. Az egyes eseteket körök vagy ellipszisek jelölik. Az aktort (azaz a felhasználót) pálcikafigurákként ábrázolja.

Aktor

Use-Case diagramban egy szerepkört, a rendszer egy felhasználóját reprezentálja.

SWOT analízis

A SWOT analízis (vagy más néven mátrix) egy stratégiai tervezési és stratégiai irányítási elemzéshez használt technika, mely segít egy személynek vagy szervezetnek abban, hogy azonosítani tudják a versenytársakkal szembeni vagy a projekttervezéssel kapcsolatos erősségeket, gyengeségeket, lehetőségeket vagy veszélyeket.

TCO

A TCO (Total Cost of Ownership) egy pénzügyi becslés, melynek célja, hogy segítse a vásárlók és a tulajdonosok számára meghatározni egy termék vagy szolgáltatás közvetlen és közvetett költségeit. Ez egy vezetői számviteli eszköz, mely a teljes költségszámításhoz vagy akár az ökológiai, gazdasági számításokhoz is alkalmazható, beleértve a társadalmi költségeket is.

Egyed-kapcsolat diagram

Az egyed-kapcsolati modell (vagy ER-modell) egy adott tudásterületen belüli, egymással összefüggő dolgokat ír le. Egy alapvető ER-modell egyed-típusokból áll (amelyek osztályozzák az érdekes dolgokat), és meghatározza az egyedek (ezen egyed-típusok példányai) között létező kapcsolatokat. A szoftverfejlesztésben az ER-modellt általában azért alkotják meg, hogy ábrázolja azokat a dolgokat, amelyekre az üzleti folyamatok végrehajtásához egy vállalkozásnak emlékeznie kell. Következésképpen az ER-modell egy absztrakt adatmodellé válik, amely egy olyan adat- vagy információs struktúrát határoz meg, amely egy adatbázisban, jellemzően egy relációs adatbázisban implementálható.

Tárolt eljárás

A tárolt eljárás a strukturált lekérdezési nyelv (SQL) utasítások gyűjteménye a hozzá rendelt névvel. Ezeket a tárolt eljárásokat egy relációs adatbázis-kezelő rendszerben (RDBMS) tárolják. Tehát lehetséges az eljárás többszörös meghívása, amely csökkenti a végrehajtási időt. Az adatok eljárásokkal módosíthatók, és a fő előnye az, hogy nem kapcsolódnak egy adott alkalmazáshoz.

GUI

A grafikus felhasználói felület vagy grafikus felhasználói interfész (angolul graphical user interface, röviden GUI) a számítástechnikában olyan, a számítógép és ember közti kapcsolatot megvalósító elemek összessége, melyek a monitor képernyőjén szöveges és grafikai elemek együtteseként jelennek meg. A grafikus felhasználói felületeken alapvető szerepe van a mutatóeszközök, például az egér használatának, amelyekkel a grafikus felület elemei intuitív módon, a fizikai világ egyfajta modelljeként kezelhetők.

Elsődleges kulcs

Egy relációs adatbázis használatakor minden táblában meg kell jelölni, hogy melyik mező, vagy melyik mezők együttesen az elsődleges kulcsok. Az elsődleges kulcs minden rekordban egyedi.

Idegen (másodlagos) kulcs

Az idegen kulcs olyan oszlop, amelyet a másik táblázatban elsődleges kulcsnak neveznek, azaz a táblázat elsődleges kulcsa idegen kulcsnak nevezhető egy másik táblázatban. Lehet, hogy az idegen kulcs duplikált & NULL értékeket tartalmaz, ha a NULL értékeket elfogadják.

2. A kiválasztott probléma

2.1. Szöveges bemutatás

A Széchenyi István Egyetem hallgatói közül sokan dolgoznak egyetemi tanulmányik mellett. Mindenki szeretne térben és időben pontosan, minél kevesebb plusz idő és energia ráfordításával eljutni munkahelyére. A pontos lokációtól függően gyalogosan, biciklivel, tömegközlekedéssel, taxival vagy saját autóval is megközelíthetik munkahelyüket a hallgatók, de ezen közlekedési módoknak számos hátránya és veszélye van.

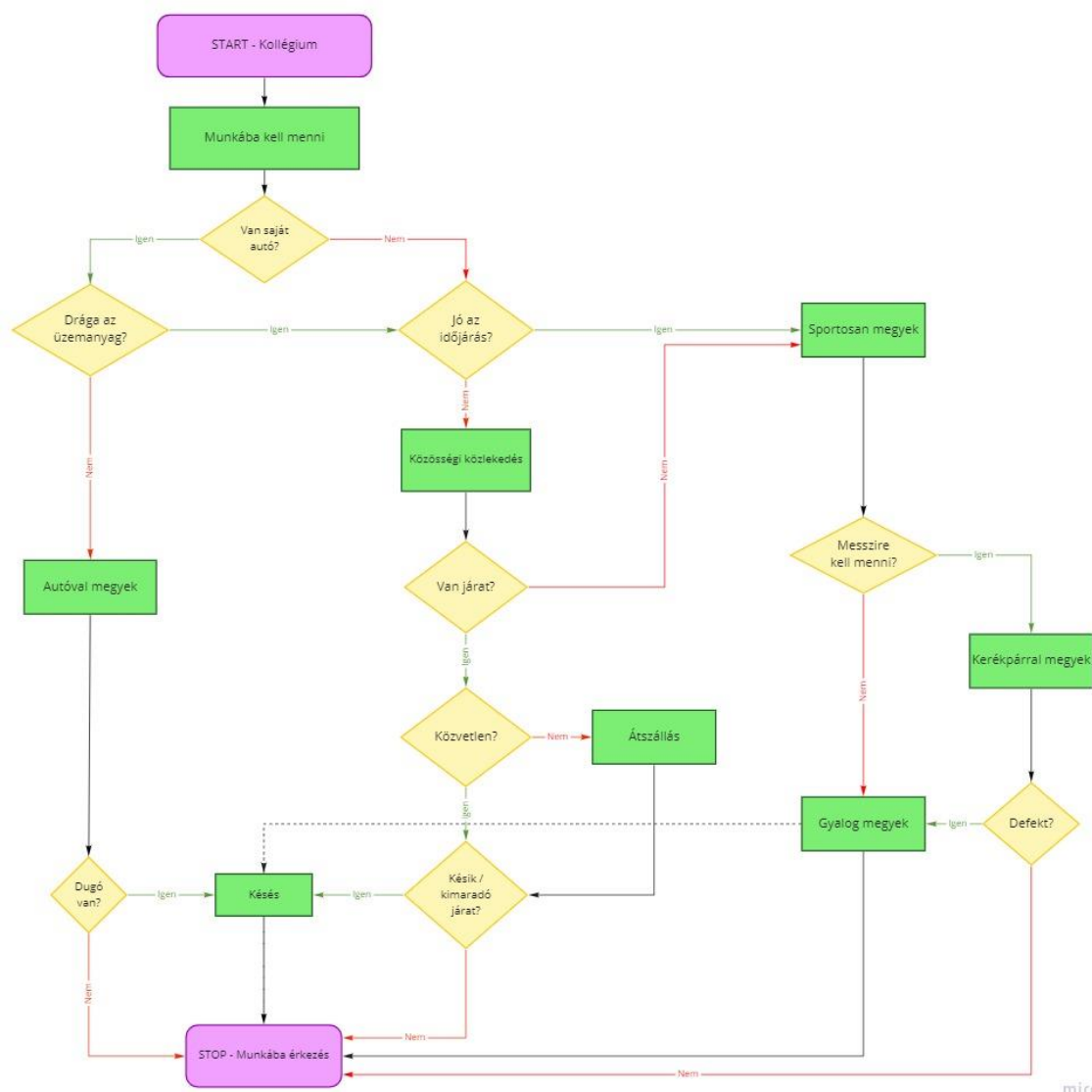
A gyaloglás és biciklizés nagyon időigényes, megterhelheti testünket és közben ki vagyunk téve az időjárási viszonyoknak, továbbá a közlekedés résztvevői és a gyárak által kibocsátott, levegőt szennyező káros anyagoknak.

A tömegközlekedés nem ér el közvetlenül Győr minden pontjára. Vannak helyek, amik egyáltalán nem lehet eljutni és vannak, amelyek az egyetemtől kiindulva csak egy vagy több átszállással érhetőek el. Sok esetben a buszok nem is közlekednek megfelelő sűrűséggel és akár órákat is várunk kell a következő járatra. Sajnos abban sem lehetünk 100%-ig biztosak, hogy a következő járat meg fog érkezni, ugyanis sok hallgatótársunk tapasztal nap mint nap járatkimaradásokat.

A taxi árak borzasztóan magasak, főleg egy egyetemi hallgató munkalehetőségeihez és fizetéséhez képest.

A saját autó használata sok korábban említett problémára megoldást tud nyújtani, viszont legalább annyi újat is vet fel. A benzin fosszilis üzemanyag, tehát korlátozott mértékben van csak jelen a bolygón és per pillanat jóval nagyobb mértékben termelik ki, mint ahogy termelődik. Emiatt sajnos az ára is egyre magasabb. Ráadásul használat közben a környezetet szennyező anyagok jönnek létre, ezzel lokális emissziót okozva. Az autók használata nem csak a levegő, de az utak minőségét is rontja. Minél több autó közlekedik, annál nagyobb mértékben romlanak az utak és annál nagyobbak lesznek a közlekedési dugók is. A munkahelyek parkolókapacitása is korlátozott, minél több autó érkezik hozzájuk, annál hamarabb telik meg a parkoló és annál messzebb tudunk csak parkolni.

2.2. A probléma folyamatábrája

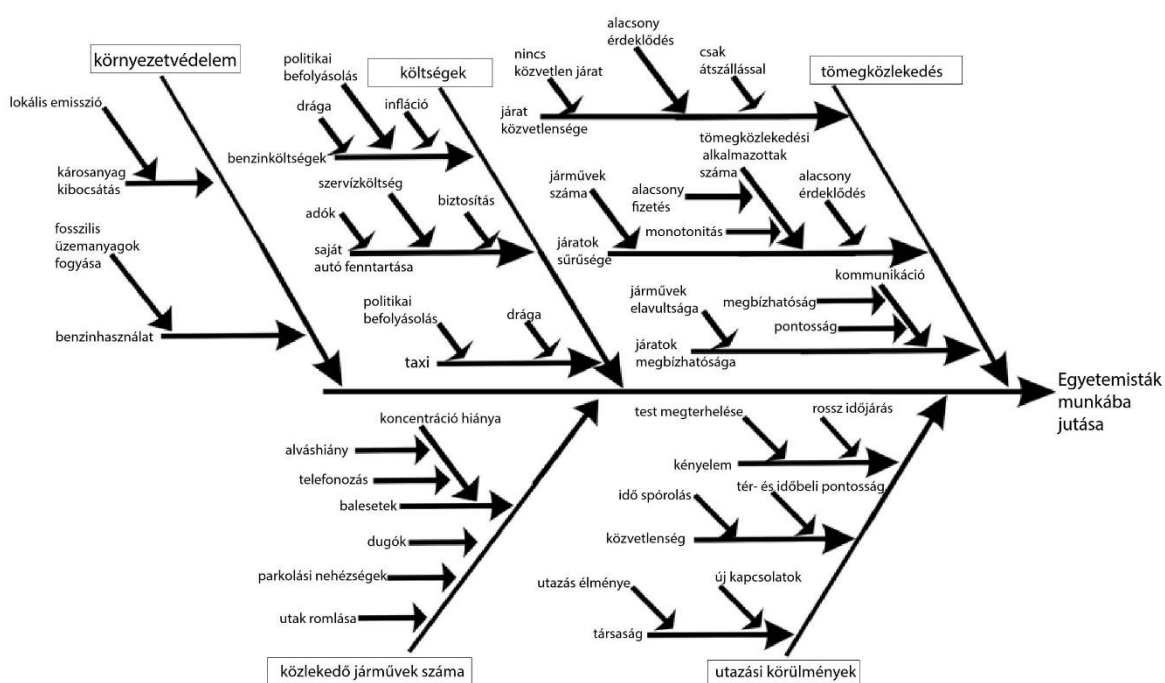


1. ábra - A probléma folyamatábrája

Az 1. ábrán a kollégiumtól (START) a munkába (STOP) való eljutás lehetséges kimenetelei láthatóak. A kiindulópont minden esetben a kollégium lesz és a végcél pedig az, hogy elérjünk a munkába olyan módon, hogy az esetleges késés a lehető legminimálisabb legyen. Kezdetben megvizsgáljuk, hogy rendelkezünk-e saját autóval és amennyiben igen akkor tovább tudunk lépni arra, hogy az üzemanyag ára mennyibe kerül. Ha ezt nem találjuk soknak, akkor választhatjuk azt az opciót, hogy kocsival megyünk munkába. Azonban itt még nem ér véget a folyamat, hiszen előfordulhat az, hogy a sok autó miatt dugó keletkezik az utakon és ezért késve érkezünk meg a munkahelyünkre. Ellenkező esetben nem fogunk késni, feltéve, ha időben indultunk el otthonról. Hogyha rendelkezünk autóval, de az

üzemanyag árát drágának találjuk vagy nincsen autónk, akkor érdemes lehet megvizsgálni az időjárási viszonyokat. Ha ezek kedvezőtlenek és rossz idő van, akkor rá vagyunk arra kényszerülve, hogy a tömegközlekedést használjuk. Ebben az esetben a következő kérdés az, hogy van-e járat és amennyiben van, a járat közvetlen-e, ugyanis ilyen módon nagy valószínűséggel el tudjuk kerülni a késést, viszont erre nincsen garancia, még így sem. Hogyha nincsen közvetlen járat akkor kénytelenek vagyunk átszállni, ami már jó eséllyel késést fog eredményezni, de természetesen ez esetben is előfordulhat az, hogy pontosan érkezünk meg. Amennyiben kedvezőek az időjárási feltételek vagy egyáltalán nincsen olyan járat, amivel el tudnánk jutni a munkahelyünkre, gyalogosan vagy kerékpárral kell közlekednünk annak függvényében, hogy mennyire kell messzire mennünk. Ezeknek előnye, hogy sportolunk, ami hozzájárulhat egészségünk megőrzéséhez és fel is frissülünk. Ha nem kell nagyon távolra mennünk, akkor a gyaloglás egy jó választás lehet, ha már kicsit nagyobb távot kell megtennünk, akkor érdemes lehet igénybe venni kerékpárunkat. Ekkor azonban kaphatunk defektet és ilyenkor kénytelenek vagyunk a mégis csak gyalog menni (magunk mellett tolni a biciklit), ami nagy valószínűséggel szintén késést fog eredményezni.

2.3. Halszáлка diagram



2. ábra - A probléma halszáлка diagramja

2.4. SWOT analízis

<p><u>Strengths - Erősségek</u></p> <ul style="list-style-type: none">• függetlenség• helyettesíthetőség• jól bejáratott rendszer• offline• olcsóbb az autófenntartásnál• hallgatói kedvezmények	<p><u>Weaknesses - Gyengeségek</u></p> <ul style="list-style-type: none">• busszal kevesebb helyen lehet megállni• ki vagyunk téve az időjárásnak• alacsony járatszám• nincs mindig elég férőhely• kimaradó járatok• hétvégén kevesebb járat
<p><u>Opportunities - Lehetőségek</u></p> <ul style="list-style-type: none">• társalgás, kapcsolatépítés• kötött menetrend• friss levegő• mozgás	<p><u>Threats - Veszélyek</u></p> <ul style="list-style-type: none">• késés• jármű meghibásodása• utasok közti konfliktus• utasok egészségügyi állapota• testiépség veszélye

3. ábra - A probléma SWOT analízise

3. Üzleti igény specifikáció

3.1. Célok

A SZEAT-APP elsődleges célja, hogy a Széchenyi István Egyetem hallgatóinak munkába jutását segítse, egyszerűbbé, gazdaságossá és környezetbaráttá tegye. A hallgató megnézheti az appon, hogy mikor és honnan indulnak sofőrök. Az appon keresztül lehetősége van jelentkezni a fuvarra. Ha az autó megtelik, akkor a sofőr benzinköltsége megtérül, az utasok pedig sokkal olcsóbban utaznak, mintha taxival mennének. Megoldásunk segítené a dugó kialakulásának elkerülését is és lokális emisszió csökkentését. Minél többen utaznak saját autó helyett közösen, annál kevesebb jármű fog közlekedni és szennyezni a környezetet. Az utasok nem lennének kitéve az időjárás viszontagságainak és a tömegközlekedés bizonytalan mivoltának sem, továbbá a hasonló korosztályú és azonos munkahelyen dolgozó utasok könnyebben tudnának ismerkedni, barátságokat kötni. Ez a közös munkavégzésre és a kollégiumi életre is pozitív hatásokkal lehet.

3.2. Stakeholderek

Projektünk elsődleges belső érdekeltjei az applikáció fejlesztői, akik elkészítik és a továbbiakban is karban tudják majd tartani, fejleszteni az applikációt. Mellettük belső érdekeltnek lehetnek még az esetleges befektetők. A külső érdekeltnek körébe tartoznak a hallgatók, akik igénybe szeretnék majd venni a szolgáltatást és az appsofőrök, akik az autókat fogják vezetni. A külső érdekeltnek közé tartozhatnak konkurenciaként még az ismert, személyszállítással foglalkozó sofőrök, például busz- és taxisofőrök.

Belső érdekelték

A fejlesztők nélkül nem jöhetne létre és nem is működhetne az applikáció. Feladatuk megtervezni a célnak megfelelően működő alkalmazást. Későbbiekben is szükség lesz a munkájukra a program karbantartása és esetleges fejlesztések kivitelezése kapcsán.

A befektetők kulcsszerepet játszhatnak az applikáció létrehozásának, fenntartásának és továbbfejlesztésének finanszírozásában. Minél népszerűbb lesz az alkalmazás, annál több emberhez juthat el és annál több utas veheti igénybe a szolgáltatásokat. Később az alkalmazást ki is lehet bővíteni és további szolgáltatásokat biztosítani.

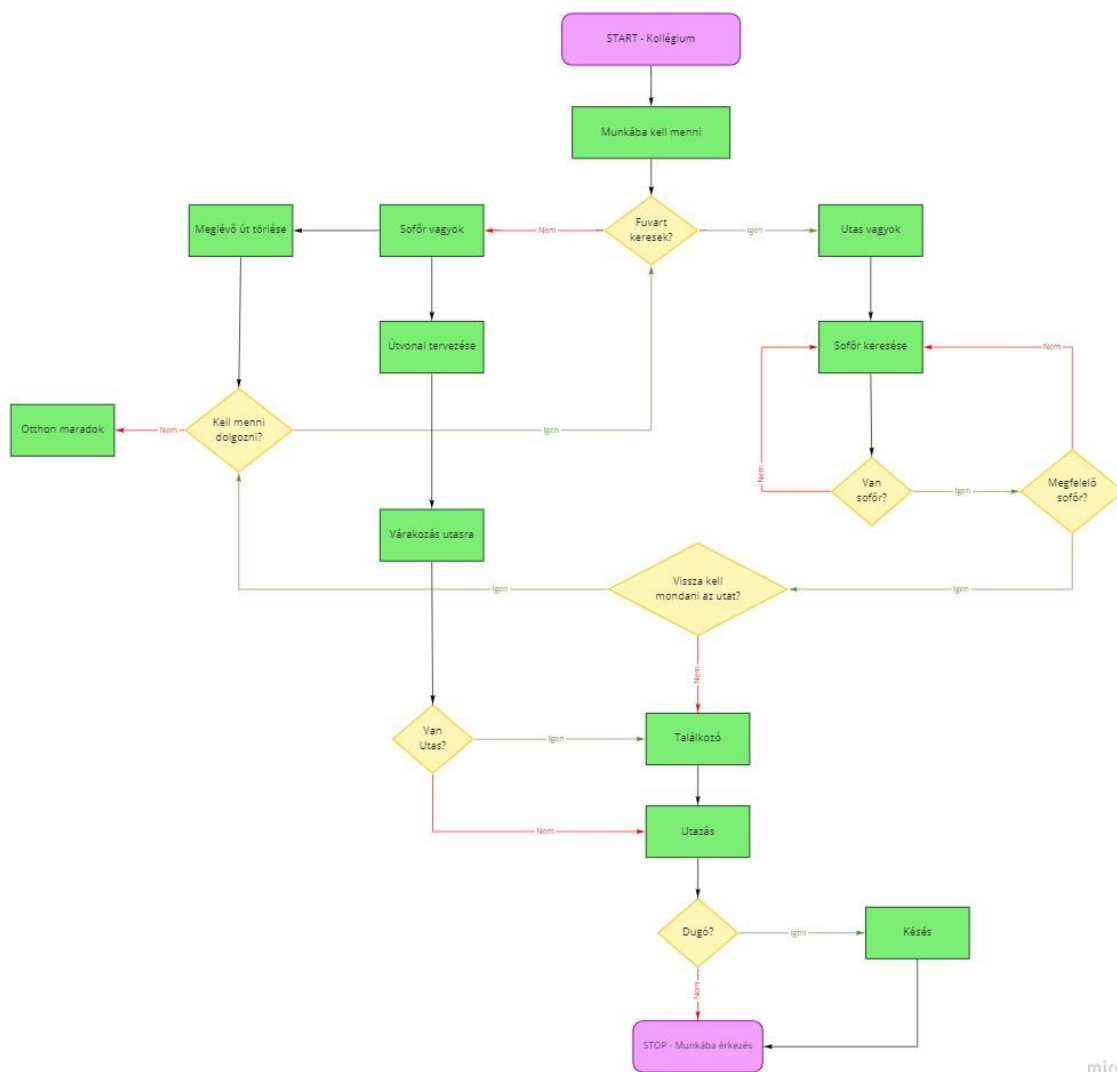
Külső érdekelték

A hallgatók érdeke, hogy munkahelyükre minél gyorsabban és pontosabban érkezzenek, utazásuk pedig minél olcsóbb és kényelmesebb legyen. A SZEAT-APP használatával a hallgatóknak nem kell alkalmazkodni a kötött tömegközlekedési menetrendekhez és az időjárás viszontagságaihoz, továbbá sokkal olcsóbban is élvezheti a személyautók kényelmét, mintha taxit hívna. Az utazás közben akár új barátokra is lelhet.

Az appsofőrök érdeke, hogy minél több utast sikerüljön összegyűjteni egy útra. Cél, hogy az összegyűlt összeg minél nagyobb mértékben fedezze az autófenntartási és benzinköltségeket.

A busz- és taxisofőrök inkább vetélytársként lehetnek érdekelték, mivel minél többen használják a SZEAT-APP-ot, annál kevesebben fognak igénybe venni más városi személyszállítási szolgáltatásokat.

3.3. Rendszer tervezett működése folyamatábrára

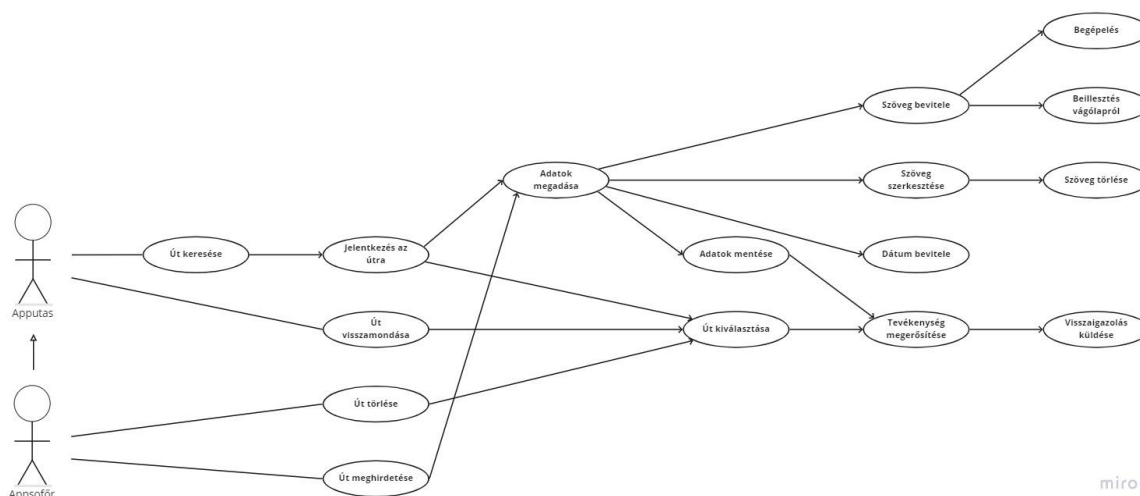


4. ábra - A megoldás folyamatábrája

A rendszerünk tervezett működésének folyamatát mutatja be. A kiinduló pont, azaz a START a kollégium lesz és a STOP pedig az, hogy megérkezzünk a munkába. Először is a fuvar keresésének a kérdését kell megválaszolni ugyanis azért indultunk el, mert munkába kell mennünk. Ha fuvart keresünk, akkor feltételezhetően utasok vagyunk (hacsak nem adódott valami probléma az autónkkal és ezért van szükségünk fuvarra) ezért találnunk kell egy sofőrt. Addig vagyunk kénytelenek keresni ameddig nem találunk egyet és ezután azt kell átgondolnunk hogy ő lesz-e a számunkra megfelelő sofőr. Hogyha úgy döntünk valamilyen oknál fogva, hogy nem akkor a keresés újból indul addig, amíg nem találunk egy megfelelő személyt. Ha sikerült megtalálni az alkalmas sofőrt, akkor meg kell vizsgálni azt, hogy valamilyen oknál fogva vissza kell-e mondanunk az utat. Az út visszamondása

történhet azért, mert nem kell mennünk mégsem dolgozni és így otthon tudunk maradni, de ha lemondtuk az utat és továbbra is kell menni dolgozni, akkor a fuvar keresés indul ismét a legelejéről. Abban az esetben, ha semmilyen zavaró körülmény nem jön közbe, akkor nem mondjuk vissza az utat és tudunk találkozni a sofőrünkkel és megtörténik az utazás. Ekkor már csak az nehezítheti meg utunkat, ha dugóba keveredünk és ez miatt a munkába, azaz a STOP-ba érkezésünk késve történik. Amennyiben megfelelő tempóban tudunk haladni akkor késés nélkül meg tudunk érkezni a munkahelyünkre feltéve, ha időben indultunk el. A másik opció lehet az, ha nem keresünk fuvart, mert sofőrök vagyunk. Ekkor meg kell terveznünk azt az útvonalat, amivel a lehető leghatékonyabb módon minden utast el tudunk vinni a munkába és mi is időben be tudunk érni dolgozni. Ha már megterveztük, hogy melyik útvonalon kívánunk menni, akkor is előfordulhat, hogy azt valamilyen oknál fogva törölnünk kell például azért, mert nem kell menni dolgozni és otthon tudunk maradni. Amennyiben ez az opció nem áll fent akkor új útvonalat kell megterveznünk és meg kell várnunk, illetve találkozunk kell utasainkkal, feltéve, hogy vannak. Hogyha nincsenek vagy már felvettük őket, akkor elkezdődhet az utazás a munkába. Ilyenkor már csak a korábban is említett dugó problémája léphet fel, és ha ez fenn áll akkor valószínűleg késni fogunk a munkából viszont ha nem akkor pontosan meg tudunk érkezni.

3.4. Rendszer tervezett működése - use case diagram



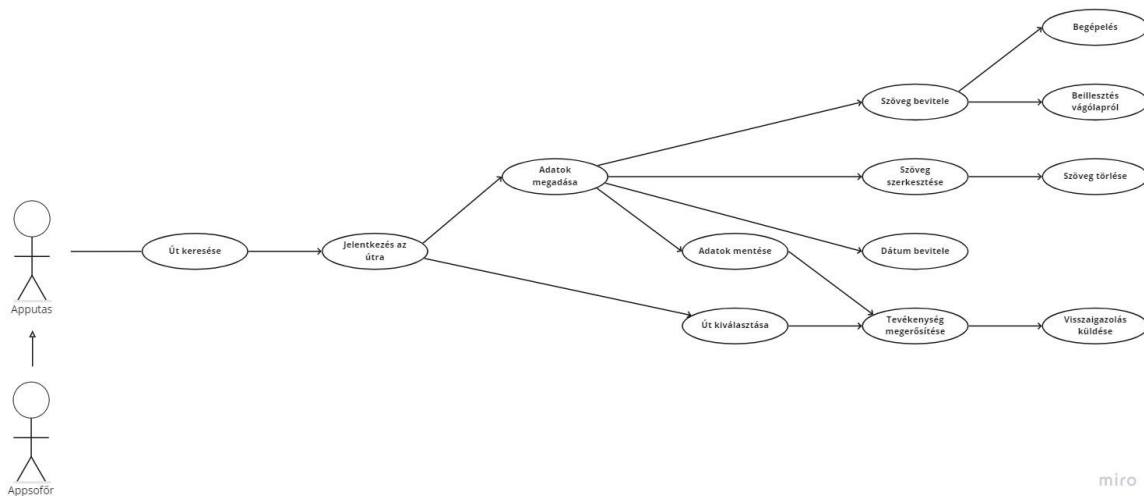
5. ábra - SZEAT-APP működése - Üzleti use case diagram

Az 5. ábra a SZEAT-APP működését szemlélteti, melyben szerepel utas és sofőr. A sofőröknek két fő szerepkörük van, az utak meghirdetése és esetleges törlése. Az utas számára az út keresés a legfontosabb, melyre utána jelentkezhet, s ezt követően, ha kívánja, le is mondhatja a fuvart. Mind az út meghirdetéshez, mind a jelentkezéshez szükség van adatok megadására, s ezen adatok a megadás és mentés után is szerkeszthetők, törölhetők, további információkkal kiegészíthetők. Az utak megtervezéséhez, törléséhez, valamint az ezekre való jelentkezéshez, vagy annak törléséhez szükséges, hogy kiválasszuk a tetszőleges útvonalat, s ezután a kívánt tevékenységet meg is erősítsük, s erről visszaigazolást kapjunk az alkalmazástól.

4. Követelményspecifikáció

4.1. Funkcionális követelmények

4.1.1. Jelentkezés az útra



6. ábra - SZEAT-APP - Use case - Jelentkezés az útra

A használati esetet az Apputas indítja el!

Előfeltétel:

- SZEAT-APP alkalmazás, regisztrált fiók, alkalmazás ismerete a kezeléshez.
- Internet elérés.
- Appsofor által meghirdetett út.

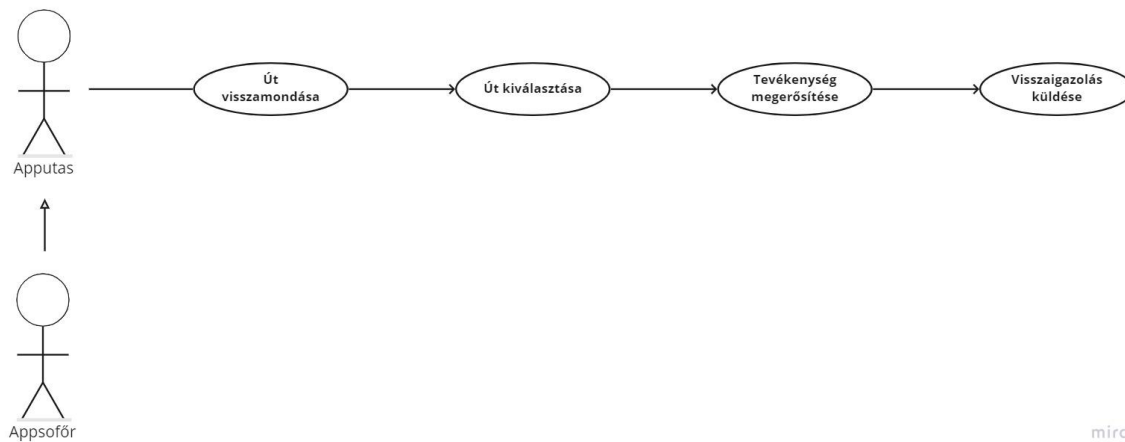
Folyamat:

A folyamat akkor indul el, amikor az Apputas munkába kíván menni. Megnyitja az Applikációt és a felületén kiválasztja az út keresése funkciót. Ilyenkor egy kereső felületen meglehet adni az utas által kiválasztott induláspontot és végcél továbbá az utazás tervezett időpontját. Ezt követően az útvonal keresése gomb megnyomásával a rendszer lekérdezi az adott paraméterekkel rendelkező Appsoforokat, majd listázza. Amennyiben ilyen útvonal létezik nincs más dolga az utasnak, mint jelentkezni az utazásra. Innentől kezdve pedig már csak az Appsofornek kell elfogadni az utast. Ha nem létezik ilyen útvonal, akkor az Apputasnak más alternatív utazási mód után kell néznie.

Hibák:

- Nincsen Appsofor a szükséges útvonalra.
- Nem működik az Applikáció által használt szerver/adatbázis.

4.1.2. Út visszamondása



7. ábra - SZEAT-APP - Use case - Út visszamondása

A használati esetet az Apputas vagy az Appsofor indítja el!

Előfeltétel:

- SZEAT-APP alkalmazás, regisztrált fiók, alkalmazás ismerete a kezeléshez.
- Internet elérés.
- Appsofor részéről: utassal való rendelkezés egy útvonalra.
- Apputas részéről: sofőrrel való rendelkezés egy útvonalra.

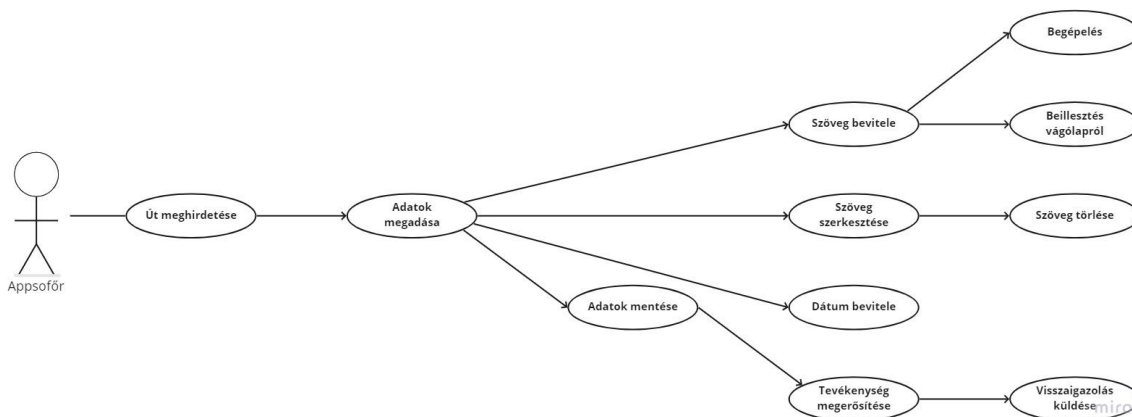
Folyamat:

A folyamat akkor indul el, amikor az Appsofőrnek vagy az Apputasnak bármely nyomós indoknál fogva le kell mondania a már előre elfogadott útját. Ebben az esetben mindkét felhasználónak a felületén a menüsorban a következő funkciót kell kiválasztani: saját útjaim, majd ezt követően a rendszer kilistázza a jövőben tervezett útjait a felhasználónak. Ebből a listából kell kiválasztani azt az útvonalat melyet a felhasználó törölni kíván. A törlés gomb megnyomásával egyidőben a rendszer figyelmezteti az aktuális másik felet (Apputast, ha az Appsofőr mondja le vagy fordítva), hogy az előzőleg már elfogadott útvonal törlésre került.

Hibák:

- Nem működik az Applikáció által használt szerver/adatbázis.
- Nem kapja meg az értesítést a másik fél.

4.1.3. Út meghirdetése



8. ábra - SZEAT-APP - Use case - Út meghirdetése

A használati esetet az Appsofőr indítja el!

Előfeltétel:

- SZEAT-APP alkalmazás, regisztrált fiók, alkalmazás ismerete a kezeléshez.
- Internet elérés.
- Igazolt Appsofőri státusz.

Folyamat:

A folyamat akkor indul el, amikor az Appsofőr egy útvonalat kíván létrehozni. Megnyitja az Applikációt és a felületén kiválasztja az út meghirdetése funkciót. Ilyenkor egy tervező felületen a sofőr meg tud adni egy útvonalat, indulási időpontot, helyszínt, az utazás várható időtartamát, és a szabad helyek számát is. Ezt követően az út meghirdetése gomb megnyomásával a rendszer nyilvánosságra hozza az adott paraméterekkel az útvonalat az applikáción belül. Ezek után a sofőrnek az utasok jelentkezéseit kell elfogadnia. Amennyiben legalább egy utas jelentkezik, a sofőr végre tudja hajtani a szolgáltatást. Ha nem jelentkezik egy utas sem, akkor a sofőr egyedül utazik.

Hibák:

- Nincsen Apputas az útvonalra.
- Nem működik az Applikáció által használt server/adatbázis.

4.1.4. Út törlése



9. ábra - SZEAT-APP - Use case - Út törlése

A használati esetet az Appsofőr indítja el!

Előfeltétel:

- SZEAT-APP alkalmazás, regisztrált fiók, alkalmazás ismerete a kezeléshez.
- Internet elérés.
- Sofőr által meghirdetett út.
- Igazolt Appsofőri státusz.

Folyamat:

A folyamat akkor indul el, amikor az Appsofőrnek bármely nyomós indoknál fogva le kell mondania a már előre elfogadott útját. Ebben az esetben az Appsofőrnek a felületén a menüsorban a következő funkciót kell kiválasztani: saját útjaim, majd ezt követően a rendszer kilistázza a jövőben tervezett útjait a sofőrnek. Ebből a listából kell kiválasztani azt az útvonalat melyet a sofőr törölni kíván. A törlés gomb megnyomásával egyidőben a rendszer figyelmezteti az utast, hogy az előzőleg már elfogadott útvonal törlésre került.

Hibák:

- Nem működik az Applikáció által használt szerver/adatbázis.

4.2. Nem funkcionális követelmények

4.2.1. Termék követelmények:

Hatékonyági követelmények:

- Karbantartott autó.
- Letisztult átlátható applikációs felület

Hordozhatósági követelmények:

- Utas nem hozhat 20kg-nál nagyobb csomagot.

Méret követelmények:

- Utasok számával arányos autó méret.
- Elegendő tárhellyel rendelkező eszköz (10 MB).

Teljesítmény követelmények:

- Applikációt futtatni képes eszköz (mobiltelefon).
- Megfelelően optimalizált alkalmazás

Használhatósági követelmények:

- Internetkapcsolat.
- Applikáció előre feltelepítése.
- 18 éves kor betöltése.
- Érvényes műszakival rendelkező jármű
- Évszaknak megfelelő gumiszett.

4.2.2. Szervezeti követelmények:

Szabványügyi követelmények:

- Megfelelő teljesítményű motor.

Implementációs követelmények:

- Applikáció univerzális használhatósága.
- Jelenlegi operációsrendszerekkel kompatibilis applikáció tervezése.
- Elfogadható válaszidő.
- Alkalmazásban a sofőrök átlátható véleményezése.

4.2.3. Külső követelmények:

Etikai követelmények:

- Utasok és sofőr közötti közös megbecsülés, kölcsönös tisztelet.
- Sofőr ne legyen priuszos.

Együttműködési követelmények:

- Pontos érkezés az utas és sofőr részéről is.

Jogi követelmények:

- Személyes adatok megfelelő kezelése.

Biztonsági követelmények:

- Jogosítvány megléte.
- Utazás közben biztonsági öv kötelező használata.
- Megfelelően funkcionáló légzsákok.

Titokvédelmi követelmények:

- Útvonal bizalmas kezelése (Ne tudjanak róla illetéktelen személyek).

4.3. Használhatóság

Egy gyakorlott felhasználó könnyedén felfedezheti az alkalmazást, figyelembe véve a hasonló szoftverek felhasználói felületét. Vannak felhasználók, akik a menüket és vannak, akik az ikonokat részesítik előnyben. A két grafikai kialakítást egybeolvasztva lenne megvalósítva az app.

Egy kezdő felhasználó számára az eszközrendszere több helyről elérhető és segítségükre van a súgó rendszer is, ami a megfelelő ikonra (egy kérdőjelre) kattintva megtalálható.

Egy, a felhasználók kérdései alapján felépített online súgó és egy fórum, amely megteremtheti a folyamatos kapcsolatot a készítőkkal, nem csak a használhatóságot, hanem a későbbi továbbfejlesztést is megkönnyítheti.

A felület letisztult, könnyen használható és átlátható legyen. A rendszer legyen látványos és figyelemfelkeltő a kellő minimalizmussal.

Az alkalmazás jól optimalizált, nem foglal a kelleténél több helyet, és nem igényel a kelleténél több nyersanyagforrást (CPU, RAM, GPU).

Felesleges adatokat ne tároljunk.

4.4. Megbízhatóság

Biztonság: Fontos, hogy minden felhasználó csak ahhoz a részhez férhessen hozzá, amihez jogosultsága van. Hiszen nem engedhetjük meg, hogy egy egyszerű felhasználó járatokat töröljön, vagy egy alkalmazás karbantartó hozzáférjen az utasok adataihoz. Ezért kulcskérdés a felhasználók megfelelő szerepkörbe való osztása mely az adminisztrátor feladata. Valamint egy olyan azonosítási procedúrával kell rendelkeznie a rendszernek mely megkerülhetetlen, és egyértelműen azonosítja a felhasználót.

Karbantarthatóság: Az esetleges hibák fellépésekor, akár a tesztelés, akár a felhasználói használat során (amennyiben ezt jelzik a fejlesztők felé) a hiba detektálása és javítása azonnal megkezdődik. A hiba súlyosságától függően elképzelhető, hogy a rendszert le kell kapcsolni a hibajavítás idejére. Kisebb hibák esetén elég egy csak a hibás fájlok kicserélésének idejére leállítani az alkalmazást futtató webszervert.

Hordozhatóság: A szoftvert (kliens oldali) alkalmazóknak lehetősége van operációs rendszert, vagy platformot váltani, ezt ugyanis a platformfüggetlen fejlesztés lehetővé teszi.

Hibakezelés: Az alkalmazás szabályos használata esetén nem fordul elő hibajelenség, nem jelenik meg hibaüzenet, minden szakszerűen működik. Amennyiben a felhasználó hibás adatot ad meg, a program azt megfelelően kezeli, és biztosítja annak javítását.

Jogszabályok: Jogszabályok betartása. A felhasználó adatainak biztonságának megőrzése

4.5. Rendelkezésre állás

A rendszer alapvetően a munkában elterjedt időszakokban vannak a legmagasabb kihasználtsági csúcsok. Office hour vagy akár a 2 vagy 3 műszakos időszakok előtt 1,5-2 órával és utána is szintén. Ezen időszakokban lehet a legnagyobb rendszer terhelésre számítani.

A rendszer felügyeletét reggel 5 órától este 11-ig biztosítani szükséges, az ezen kívül eső intervallumon csak a következő napra lehet a problémákat egy automatikus hangrögzítő segítségével leadni.

MTBF: A rendszer kritikus követelménye, hogy az esetleges hibák/leállások között a lehető legtöbb idő teljen el, azaz a lehetőleg minimálisnak kell lennie a felléphető hibák számának, melyek a működést befolyásolják. Statisztikák alapján egy leállás átlag ideje 32 perc. És ilyen mértékű leállás fél évente egyszer történik, tehát az MTBF együtthatónk egy évre számolva 16-ra számítható.

MTTR: Ebben az esetben is azt kell figyelembe venni, hogy a rendszer mennyi ideig van használhatatlan állapotban, de ebben az esetben fontos szempont a hogy a leállások alatt mennyi javítás volt szükség elvégezni annak érdekében, hogy a rendszer újra működőképes legyen. Ebben az esetben is , mint az MTBF esetében a rendszer érdeke, hogy a lehető legalacsonyabb hibával működjön. A rendszer karban tarthatósága egyszerű, mert az applikáción kívül a webszervert kell az adatbázissal karban tartani. A webszerverről a biztonság kedvéért naponta egyszer hajnali 3-kor biztonsági mentés készül, továbbá havonta egyszer újraindítják.

4.6. Teljesítmény

A rendszer biztonságának megőrzése érdekében folyamatosan figyeli biztonsági problémák szempontjából és állapotjelentést ad, amely megjelenik egy oldalon. Az állapotjelentés figyelmeztet a négy fontos területtel kapcsolatos általános problémákra és javaslatokat ad a kijavításukra. A rendszer különböző módon reagál alacsony és magas terhelésre.

A teljesítmény akkor optimális, amikor a felhasználó zökkenőmentesen és bármiféle lassulás nélkül tudja használni a rendszert.

Teljesítményvesztés oka lehet hardveres vagy szoftveres.

A rendszer folyamatosan számon tartja a háttérben futó feladatok erőforrástól elvett mennyiségét, ha valamelyik túl sokat használ, akkor elmenti a rendszer aktuális állapotát, majd megpróbálja a folyamatot leállítani, ezzel elérve az optimális teljesítményt. Erről a folyamatról a felhasználó értesítést kap.

A teljesítményvizsgálatot fontos elvégezni az adatbázisra és az operációs rendszerre egyaránt. A rendszert hatalmas mennyiségű munka terheli, annyira, hogy a rendszer már nem működik egy adott pillanatban. Ez lehetővé teszi számunkra a rendszer szűk keresztmetszeteinek azonosítását. Az adatbázis-tesztek során a leggyakoribb problémák a következők: A rendszerre gyakorolt nagy nyomás szerepet játszik az adatbázis-tranzakciók állapotának meghatározásában. Ha az adatbázis szoftver hibás elképzeléssel rendelkezik a tranzakciók állapotáról, akkor azonnal következetlenek lesznek. A vizsgálat elvégzése után a régi teszt adatokat meg kell tisztítani, és teret kell adni az új tesztadatokhoz. Az operációs rendszernél hasonlóan kell eljárni.

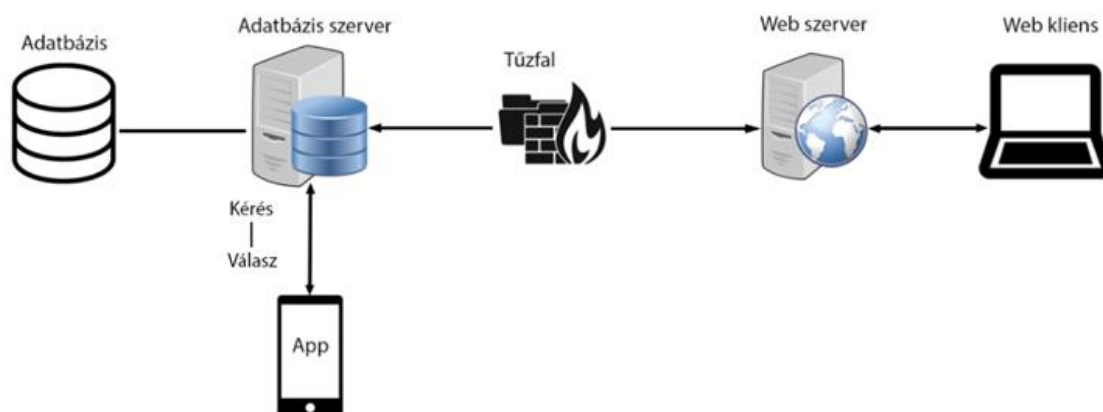
4.7. Támogatottság

Felhasználói támogatottság: Biztosítjuk a feladatok ellátása közben kialakult hardveres és szoftveres hibák kivizsgálását és azonnali elhárítását.

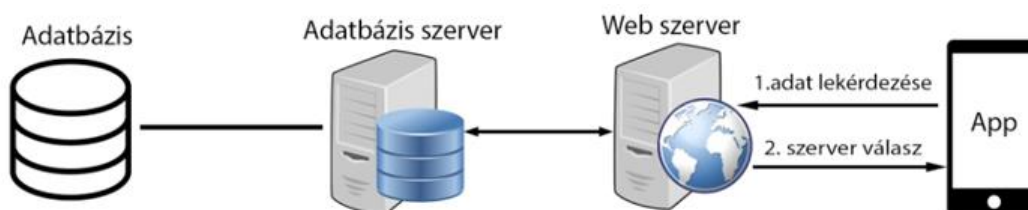
Az **információk** különböző módon jelennek meg. Súlyos hiba esetén egy felkiáltó jel mellett piros, nagy betűkkel írja ki, ha probléma merült fel a rendszerrel. Kisebb hiba esetén narancssárga felkiáltó jel mellett narancssárga, közepes méretű betűkkel írja ki a hiba okát. További információkat fekete színnel írat ki a képernyőre.

5. Technikai specifikáció

5.1. Architektúra terv



10. ábra - Fejlesztői architektúra terv



11. ábra - Felhasználói architektúra terv

A tervezett rendszer a következő összetevőkből áll:

- Adatbázis
- Adatbázis szerver
- Tűzfal
- Web szerver
- Web kliens
- Android/IOS készülék

Az architektúrában az adatforgalom szempontjából a következőképpen zajlik:

Az applikáció a web szerverrel kapcsolatba lép és lekéri a szükséges adatokat, innen kétféleképpen alakulhat. Az első eset, hogy a kérés eredményének meghatározására nem szükséges az adatbázis. Ebben az esetben a web szerver azonnal tud válaszolni. A másik eset, ha szükséges az adatbázis, ilyenkor a web szerver továbbítja a kérést az adatbázis szerver felé. Miután megkapta az adatbázisból a szükséges adatot, elküldi a web szervernek, végül pedig a szerver elküldi a választ az web kliensnek.

Gondolkodás után az Androidot és az IOS-t választottuk ki a megfelelő platformoknak. Azért erre esett a választás, mert az emberek így kényelmesen, bárhol, bármikor hozzáfér a szükséges információkhoz, csak egy telefonra vagy egy tabletre van szükségük.

A teljesítménytesztelés segít a rendszerek megfelelő karbantartásában és a hibák kijavításában, mielőtt a problémák elérnék a rendszerfelhasználókat. Segít fenntartani az alkalmazások hatékonyságát, válaszképességét, méretezhetőségét és sebességét az üzleti követelményekhez képest. Ha hatékonyan végez, a teljesítménytesztelésnek meg kell adni a szűk keresztmetszetek megszüntetéséhez szükséges diagnosztikai adatokat, amelyek gyenge teljesítményt eredményeznek. Szűk keresztmetszet akkor fordul elő, ha az adatfolyam megszakad vagy leáll, mert nincs elegendő kapacitás a számítási feladat kezeléséhez. A terheléstesztelés a rendszer teljesítményét méri a számítási feladatok növekedésével. Azonosítja, hogy az alkalmazás hol és mikor szakad meg, így az éles környezetbe való szállítás előtt kijavíthatja a problémát.

5.2. Platform tulajdonságok és követelmények

A rendszer futtatásához ajánlott hardverkörnyezet:

Az alkalmazás használatához szükség van 200.0 MB szabad tárhelyre a készüléken.

A rendszer futtatásához szükséges szoftverkörnyezet:

A szoftver Android rendszer esetén az Android 12.0 Lollipop vagy annál újabb verzióval képes működni, míg Apple készülékek esetében az iOS 15.4 a követelmény. A szoftver egyidejűleg több felhasználó kiszolgálására is alkalmas.

Egy platform független **böngészőből is elérhető változata** is lesz az applikációnak, amely további szabadságot nyújt a felhasználóknak.

5.3. Fejlesztői eszközök

Az alapvető platformunk a chat alapú kommunikációhoz az erre a célra létrehozott **Messenger** csoport. Itt folyik a mindennapos kommunikáció, és itt szoktuk összegyűjteni a kérdéseket és a megoldandó feladatokat.

A gyakorlati órák után tartott személyes megbeszéléseken, kívül minden vasárnap este tartunk online megbeszélést, amihez a **Discord** programot használjuk. Általában itt kerülnek elfogadásra az adott heti és kiosztásra a következő heti feladatok, továbbá itt vitatjuk meg a felmerülő kérdéseket is.

A megbeszélésekről szöveges dokumentáció készül a **MS Word** programmal. Ebben a programban történik a projekt dokumentum vezetése, szerkesztése is.

A dokumentumok kezelésére, tárolására az erre a célra létrehozott **GitHub** repository-t használjuk. Ide felkerülnek minden megbeszélés után a jegyzőkönyvek, illetve itt található a projekt dokumentum legfrissebb verziója is. A bevált gyakorlat szerint a projektagok elküldik a munkáikat a projektvezetőnek, aki ezeket elfogadja és beilleszti a megfelelő formátumban a közös projekt dokumentumba.

A folyamatábrák és use case diagrammok a **Miro** program ingyenes, online verziójával készültek, a halszálla diagrammhoz **Photoshop**-ot használtunk.

Szoftveres háttérnek a **Visual Studio Code**-ot és a C++ programozási nyelvet választottuk.

Adatbázis tervezéséhez pedig egy nyílt forráskódú, webes adatbázis kezelő felületet, a **MSSQL Server Management Studio** és a **MSSQL**-t fogjuk alkalmazni.

6. Gazdasági számítások

6.1. Felmerülő költségek

- Fejlesztési költségek
- Karbantartási költségek
- Üzemeltetési költségek
- Technikai követelmények
- Jogi költségek
- Domain név bérlete
- Amortizáció

6.2. TCO (Total Cost of Ownership)

A TCO-t magyarul egy termék valódi árának is nevezik. Röviden összefoglalva ez a szám egy rendszer teljes élettartamára vetített költségek összege, mely sok tényezőtől épül fel.

Munkabér:

6 fő van a csapatban, átlagosan heti 10 órát foglalkozik egy csapattal a feladattal, valamint a projekt 1 félévig, azaz tétélezzük fel, hogy 6 hónapig tart. A hónapokat átlagosan 4 hétnek vettük. A bruttó órabér legyen kb. 10.000 Ft. Ezen adatok alapján a teljes projektben a bérek költsége itt látható:

Bruttó munkabér = $6 * 10 * 4 * 10.000 \text{ Ft/hó} = 2.400.000 \text{ Ft/hó}$ → 6 hónapra: **14.400.000 Ft** bruttó költség

Technológiai háttér:

Mivel mind a VS Code mind a MSSQL Server Management Studio ingyenes szoftver, így ezek költsége **0 Ft**.

Amennyiben a vállalkozás növekedik szükség lehet új szerver vásárlásra, valamint egyéb váratlan költségek is felmerülhetnek, erre szánjunk br. **6.000.000 Ft**-ot a félév során.

Egyéb költségek:

Vannak bizonyos alapköltségek is, melyek a villamos energia, fűtés, víz, valamint iroda bérlet. Egy kisebb lakást, melyet irodaként be lehet rendezni kb. Győrben 150.000 Ft/hó + rezsiért lehet bérelni. Utóbbira számoljunk átlag 15.000 Ft-ot emberenként, összesen 90.000 Ft-ot. Így ez félévre vetítve:
 $(150.000 + 90.000) * 6 \text{ br. Ft} = \text{bruttó } \mathbf{1.440.000 \text{ Ft}}$

A fenti költségeket összesítve a CTO: $14.400.000 + 6.000.000 + 1.440.000 = \mathbf{21.840.000 \text{ Ft}}$

6.3. Tervezett bevételek

- Szponzorok szerzése, befektetők bevonása
- Reklámok megjelenítése az appon belül
- Terjeszkedés más településekre, más egyetemekre

6.4. Fenntarthatóság növelése, költségek csökkentése

- Saját szerverpark üzemeltetése
- Saját domain hostolása

6.5. Jövőbeli bevételi lehetőségek

- Prémium előfizetés lehetősége
- Szolgáltatások fizetősé tétele
- Egyszeri és havidíjas szolgáltatások bevezetése
- Program bérbeadása

7. Adatbázis terv

7.1. Szöveges leírás

A fél éves feladatunkban egy utasszállító applikációt készítettünk. Az adatbázisunkban eltároljuk, a sofőrök, utasok, és az útvonalak adatait. A fuvart követően az adatokat eltároljuk fél éven keresztül, ezek után pedig automatikusan törlésre kerülnek. Az adatok eltárolásába regisztrációnál a felhasználóknak bele kellett egyeznie. A törölt járatok esetén az adatokat el sem tároljuk az adatbázisba.

Alapvetően 5 tábla van: **User**, **Driver**, **Car**, **Passenger**, **Route**. A User tábla tárolni fogja a felhasználó ID-ját, a vezetéknév és keresztnévét külön-külön, a születési dátumát és a címét. A Driver tábla tárolni fogja a sofőr, az autó, és a felhasználó ID-ját külön. Továbbá a táblában a jogosítvány megléte és a sofőr esetleges büntetett előélete is el lesz tárolva. Emellett a sofőr jó, rossz, és összesített értékelése is külön el lesznek tárolva. A Car tábla tárolni fogja a kocsi ID-ját, márkáját, ezen belül a pontos modellt, a színét, a rendszámát és a rendelkezésre álló szabad helyek számát. A Route tábla tárolni fogja az útvonal és az autó ID-ját, az indulást és az érkezést, az út pontos dátumát és árát. A Passenger tábla tárolni fogja az utas, a felhasználó és az útvonal ID-jét.

Egyelőre ennyi adatot tárol az adatbázis, viszont az applikáció fejlődése során lehetőség lesz újabb típusú adatokat rögzíteni.

7.2. Táblák

Relációs séma
USER(UserID, FirstName, FamilyName, DateOfBirth, Address)
DRIVER(DriverID, UserID, CarID, HaveDrivingLicense, CriminalRecord, PlusRate, MinusRate, Rate)
CAR(CarID, Brand, Model, Colour, LicensePlateNumber, FreeSeats)
ROUTE(RouteID, CarID, Departure, Destination, DateOfRoute, Price)
PASSENGER(PassengerID, UserID, RouteID)

USER tábla
UserID [int] NOT NULL
FirstName [nvarchar](25) NOT NULL
FamilyName [nvarchar](25) NOT NULL
DateOfBirth [datetime] NOT NULL
Address [nvarchar](50) NOT NULL

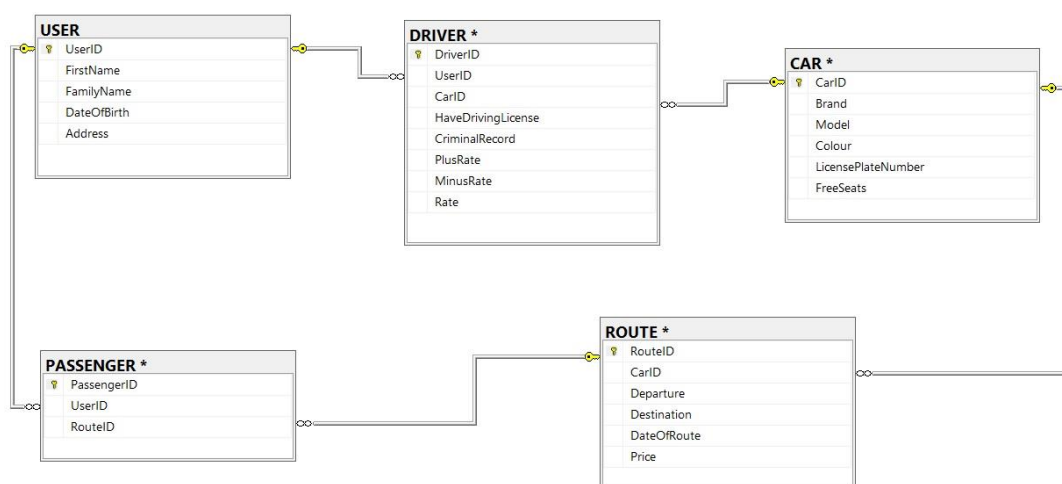
PASSENGER tábla
PassengerID [int] NOT NULL
UserID [int] NOT NULL
RouteID [int] NOT NULL

DRIVER tábla
DriverID [int] NOT NULL
UserID [int] NOT NULL
CarID [int] NOT NULL
HaveDrivingLicense [bit] NOT NULL
CriminalRecord [bit] NOT NULL
PlusRate [int] NOT NULL
MinusRate [int] NOT NULL
Rate [float] NOT NULL

CAR tábla
CarID [int] NOT NULL
Brand [nvarchar](15) NOT NULL
Model [nvarchar](15) NOT NULL
Colour [nvarchar](15) NOT NULL
LicensePlateNumber [nchar](6) NOT NULL
FreeSeats [nchar](1) NOT NULL

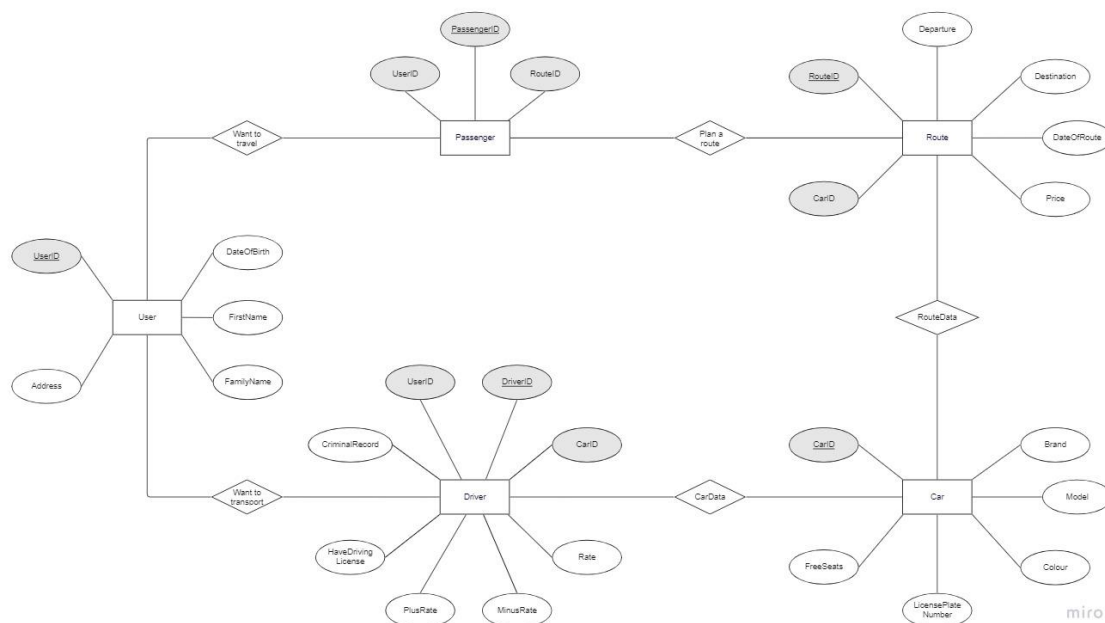
ROUTE tábla
RouteID [int] NOT NULL
CarID [int] NOT NULL
Departure [nvarchar](50) NOT NULL
Destination [nvarchar](50) NOT NULL
DateOfRoute [datetime] NOT NULL
Price [money] NOT NULL

7.3. Adatmodell



12. ábra - Adatmodell

7.4. Egyed-kapcsolat diagram



13. ábra - Egyed-kapcsolat diagram

7.5. Kulcsok és megszorítások

7.5.1. Elsődleges kulcsok

USER tábla: UserID

Driver tábla: DriverID

CAR tábla: CarID

ROUTE tábla: RouteID

PASSENGER tábla: PassengerID

7.5.2. Másodlagos kulcsok

Driver tábla: UserID, CarID

ROUTE tábla: CarID

PASSENGER tábla: UserID, RouteID

7.5.3. Megszorítások

USER tábla:

- Az alkalmazás használójának minimum 18 évesnek kell lennie (jelenlegi dátum - születési dátum ≥ 18)

DRIVER tábla:

- A jogosítvány megléte nem lehet régebbi, mint a születési dátum
- A bűnözési pontok száma nem lehet 10-nél nagyobb

CAR tábla:

- A szabad helyek száma nem lehet nagyobb, mint a férőhelyek száma (ha csak autók lesznek az adatbázisunkban akkor nem lehet pl. 5-nél nagyobb)
- A rendszám táblán a rendszámnak megfelelő formátumban kell szerepelnie, azaz 3 betű kötőjel és 3 szám

ROUTE tábla:

- A célállomás nem lehet országon kívüli
- A fuvar ára nem lehet 6 számjegynél nagyobb jegyű
- Az indulási dátum nem lehet az aktuális dátumnál nagyobb

7.6. Tárolt eljárások

Az adatbázisban tárolt eljárások a program üzemeltetését segítik.

- CreateUser: Az applikációban megadott adatok alapján automatikusan hoz létre egy új felhasználót. Az ID kiosztás is automatikusan történik.
- AddDriver: Első használatkor rákérdez az app, hogy utazni vagy utaztatni szeretnénk. A sofőr lehetőség választása esetén az applikáció felveszi a szükséges további adatokat és átadja az adatbázisnak, ami ezeket az adatokat automatikusan elmenti és ezek alapján létrehoz egy új sofőrt. Az ID kiosztás automatikusan történik.

- **AddPassenger:** Első használatkor rákérdez az app, hogy utazni vagy utaztatni szeretnénk. Az utas lehetőség választása esetén az applikáció felveszi a szükséges további adatokat és átadja az adatbázisnak, ami ezeket az adatokat automatikusan elmenti és ezek alapján létrehoz egy új utast. Az ID kiosztás automatikusan történik.
- **AddCar:** Amennyiben sofőrök vagyunk, akkor megadhatjuk az autónk adatait, a szín és a márka kötelező mezők, hogy beazonosítható legyen a jármű az utas számára találkozáskor, a többi adat megadása opcionális. Az applikációban megadott adatokat az adatbázis lementi és létrehoz egy új autót. Az ID kiosztás automatikusan történik.

7.7. Továbbfejlesztési ötletek

Továbbfejlesztési ötleteknek olyan lehetséges alkalmazás funkciókat találtunk ki, amik megkönnyíthetik a SZEAT-APP későbbi használatát.

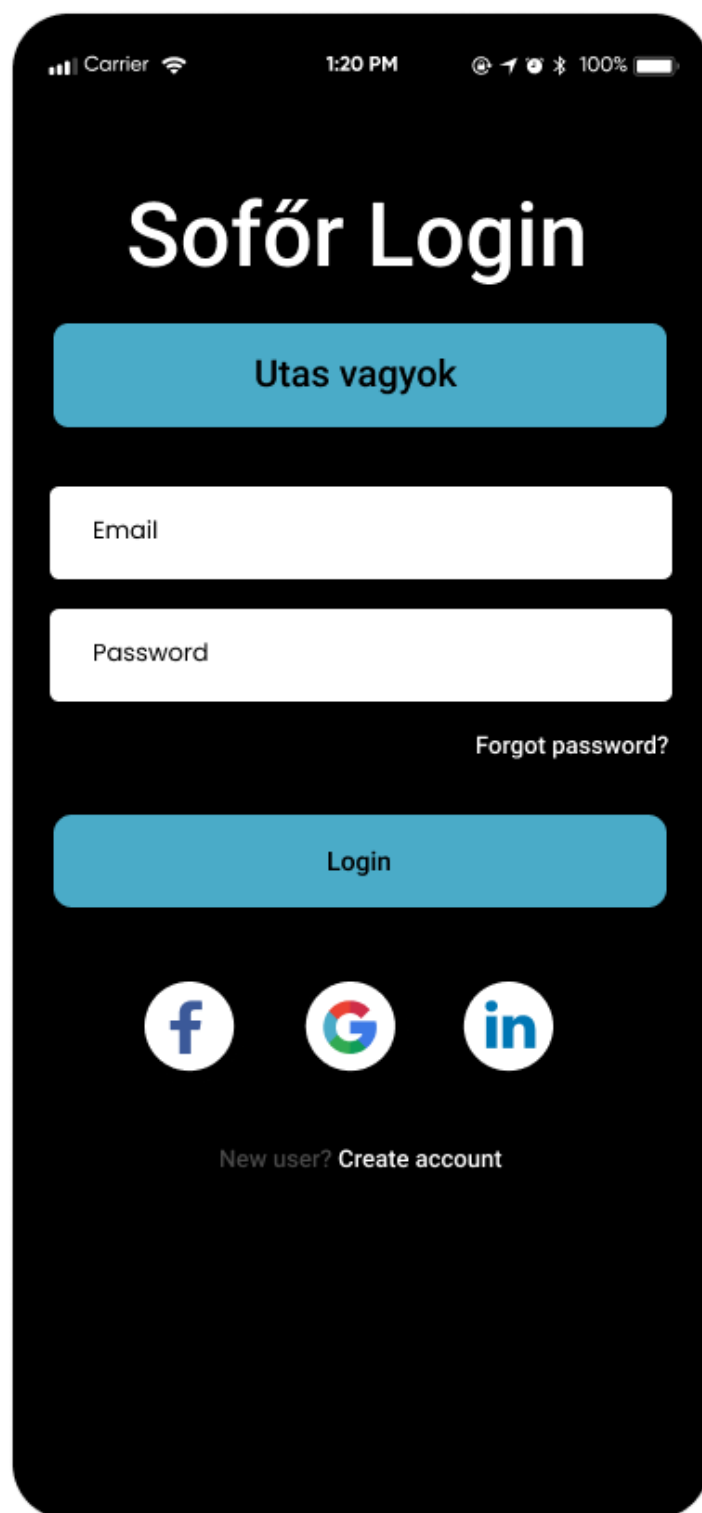
- **SelectCollege:** Mivel az app egy egyetemisták számára fejlesztett megoldás, így beépített funkcióként ki lehetne választani az indulási helyként, hogy melyik kollégiumi portától induljon a fuvar. Természetesen más opció is megadható, ez csak kényelmi megoldás.
- **CreateRoute:** Nem alapvető szükséglet, de beépíthető olyan funkció, mely grafikusán megjeleníti a start és a cél közti útvonalat a könnyebb átláthatóság kedvéért, becsült utazási időt is lehet vele számolni. A funkció megvalósításához lehetséges más térkép rendszerek bevonása.
- **CreateSchedule:** Hogyha valaki már előre tudja, hogy minden héten egy bizonyos időpontban utazna vagy tudna utasokat szállítani, akkor előre is felviheti a menetrendet és eszerint tud tervezni a többi sofőr vagy utas is.
- **RepeatRoute:** Mikor új útvonalat akarunk létrehozni, felajánlja a korábbi már felvitt utakat és akár azokból is választhatunk. Az adatokat automatikusan kitölti a rendszer, ezzel gyorsítva a folyamatot.

8. GUI terv

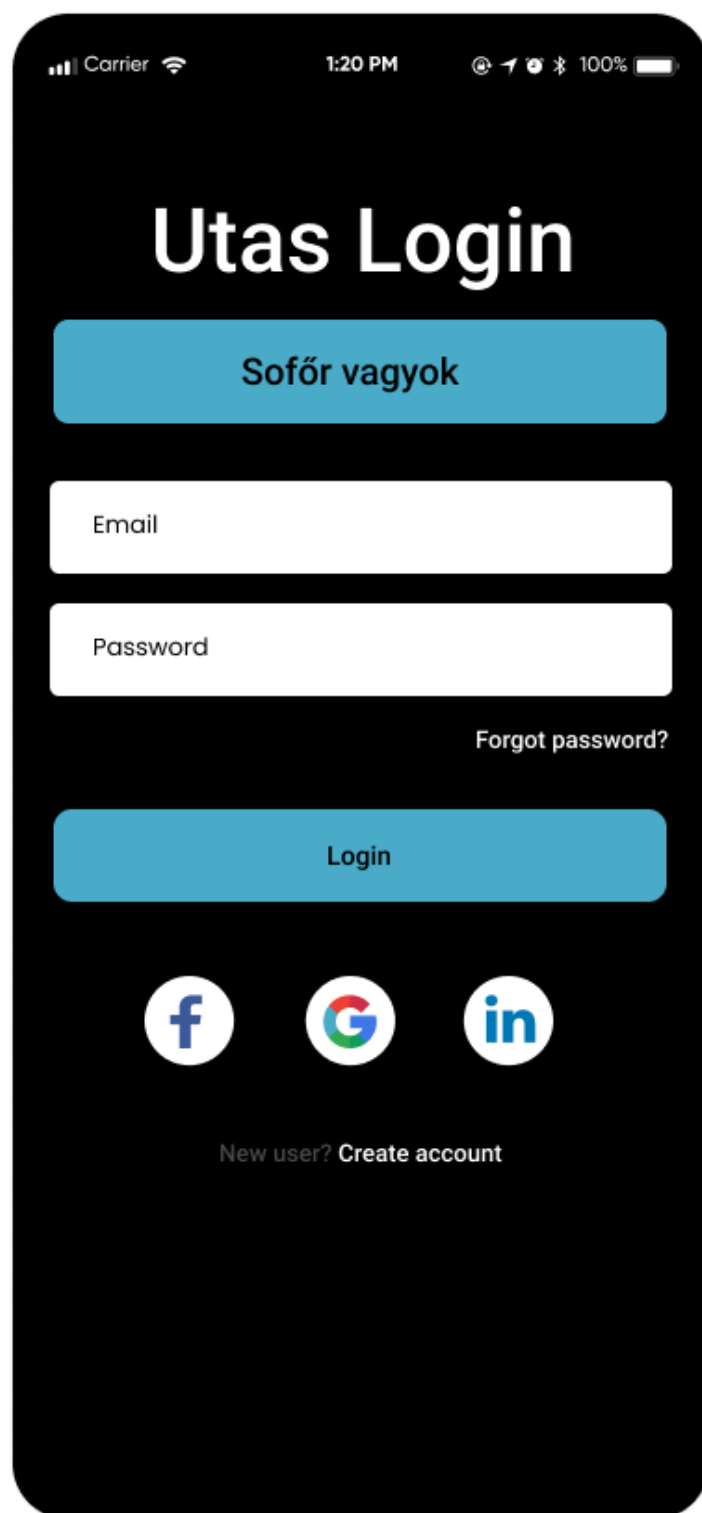
Csapatunk a GUI elkészítéséhez a **Figma** szoftvert használta, GUI tervünk ezen a [linken](#) elérhető.



14. ábra - Az applikáció nyitóképe, melyet belépéskor lát a felhasználó



15. ábra - Az applikáció sofőrök számára fenntartott belépési felülete



16. ábra - Az applikáció utasok számára fenntartott belépési felülete



17. ábra - A sofőrök számára fenntartott menü

Carrier 1:20 PM 100%

Fuvar meghirdetése

Honnan ?

Hova ?

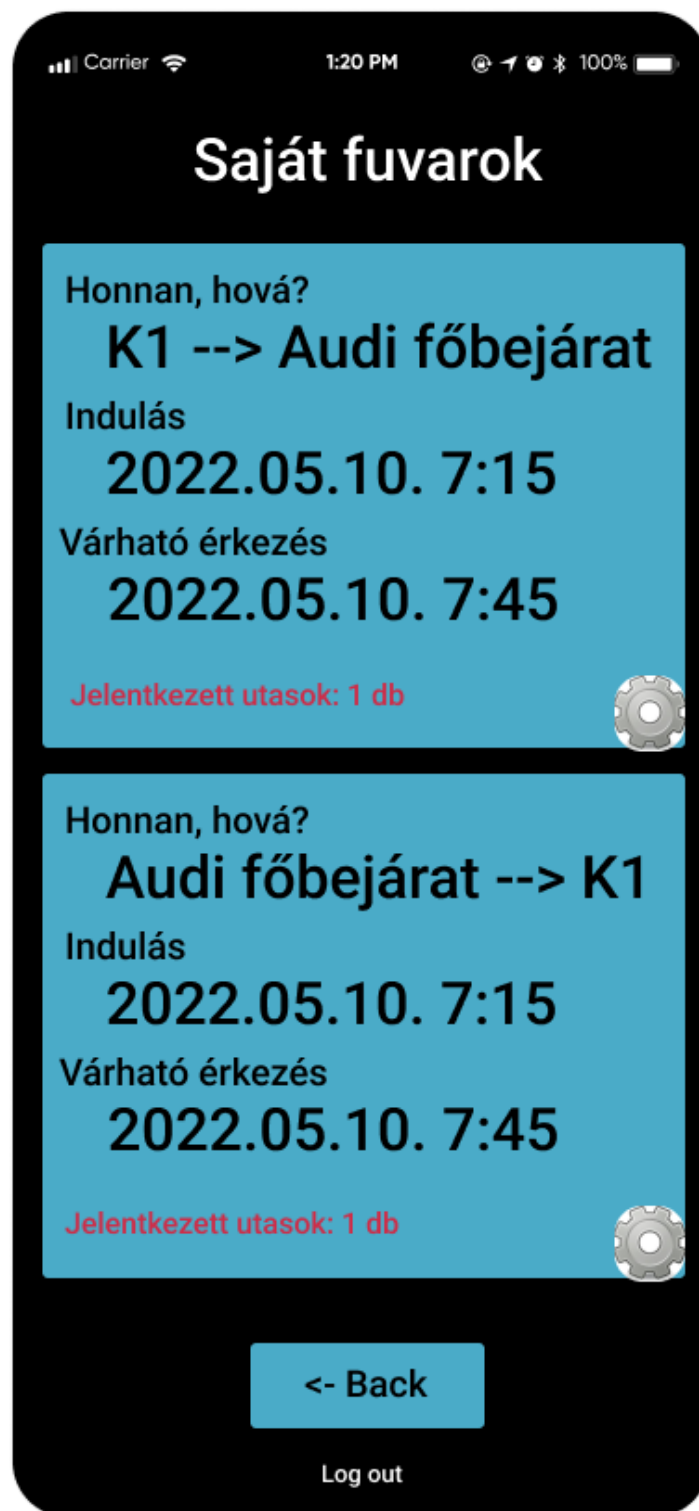
Mikor ? (yyyy.mm.dd; hh:mm)

Meghirdetés

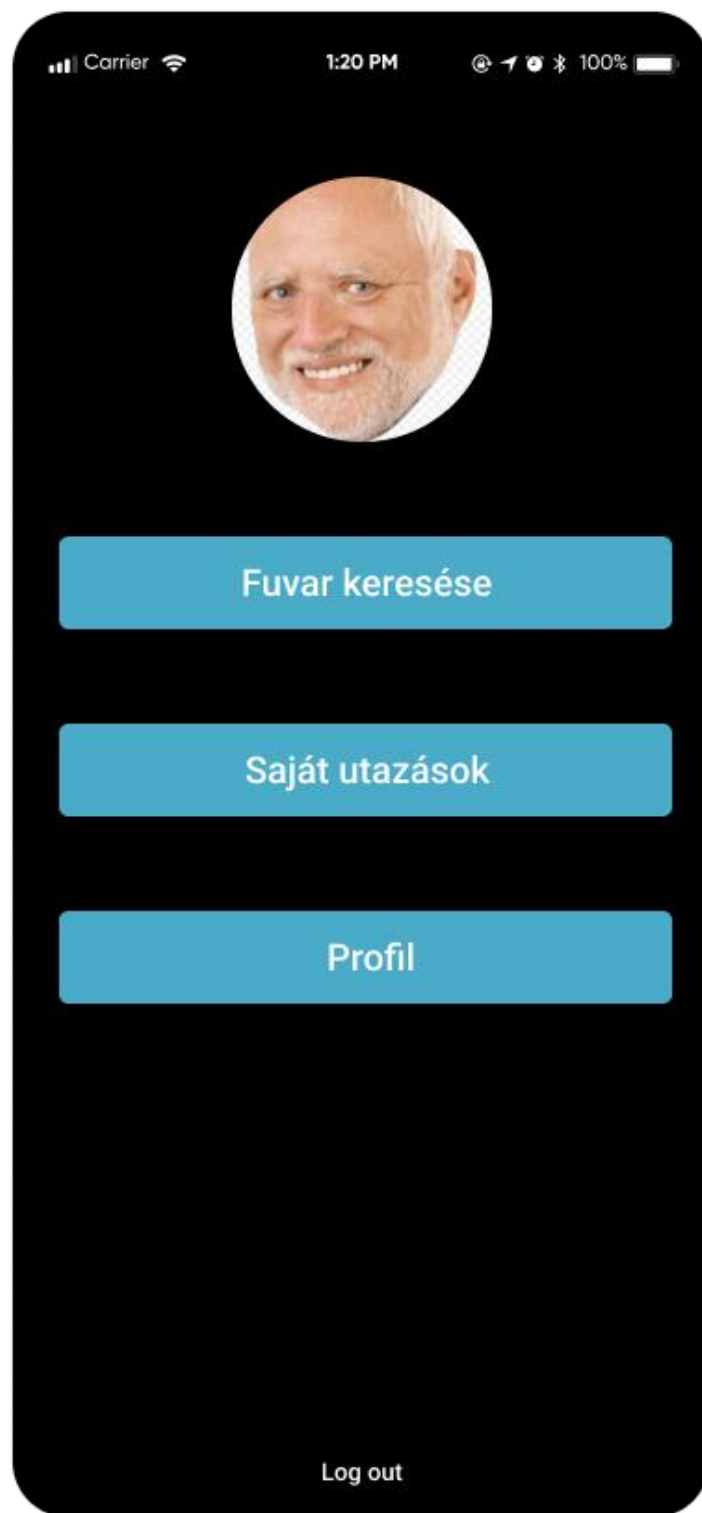
<- Back

Log out

18. ábra - Fuvarok meghirdetésének felülete



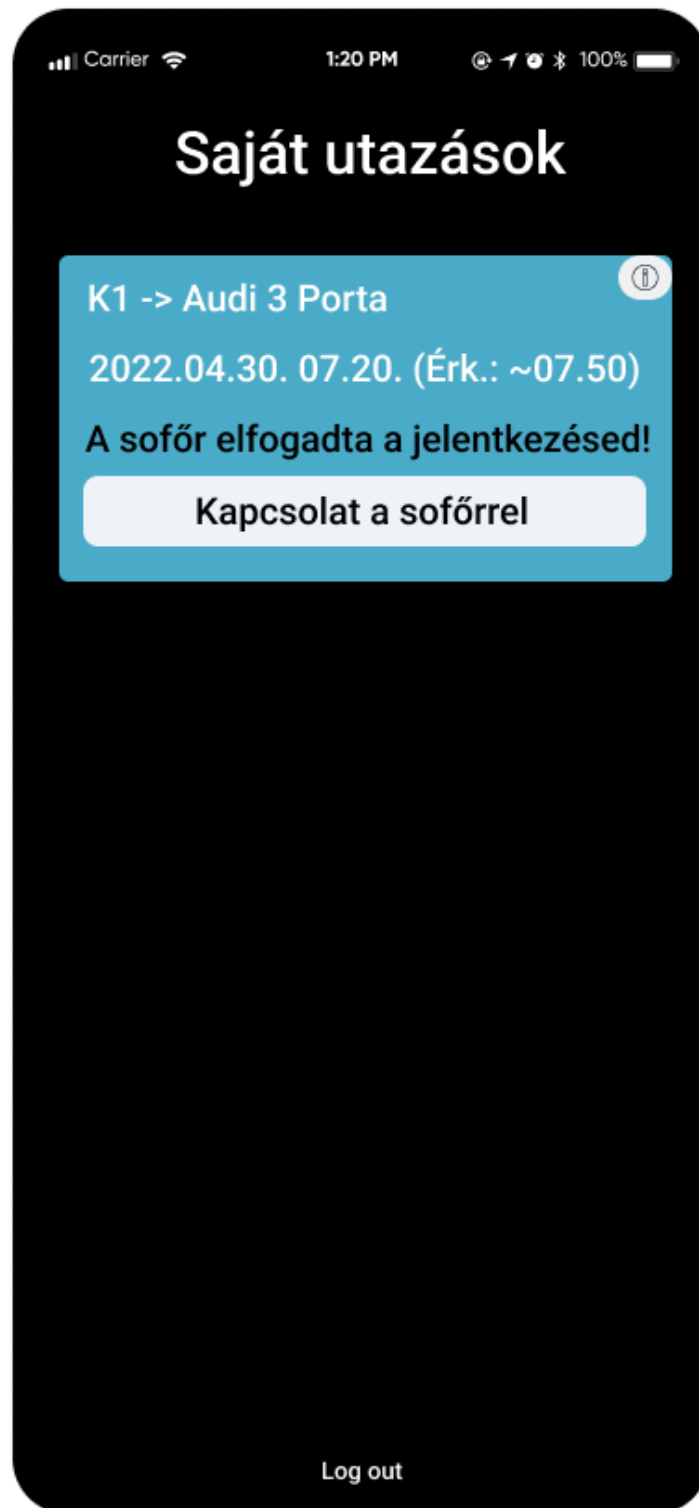
19. ábra - A sofőrök ezen a felületen áttekinthetik a saját, meghirdetett fuvarjaikat



20. ábra - Az utasok számára fenntartott menü



21. ábra - Az applikáció utasok számára fenntartott keresési felülete



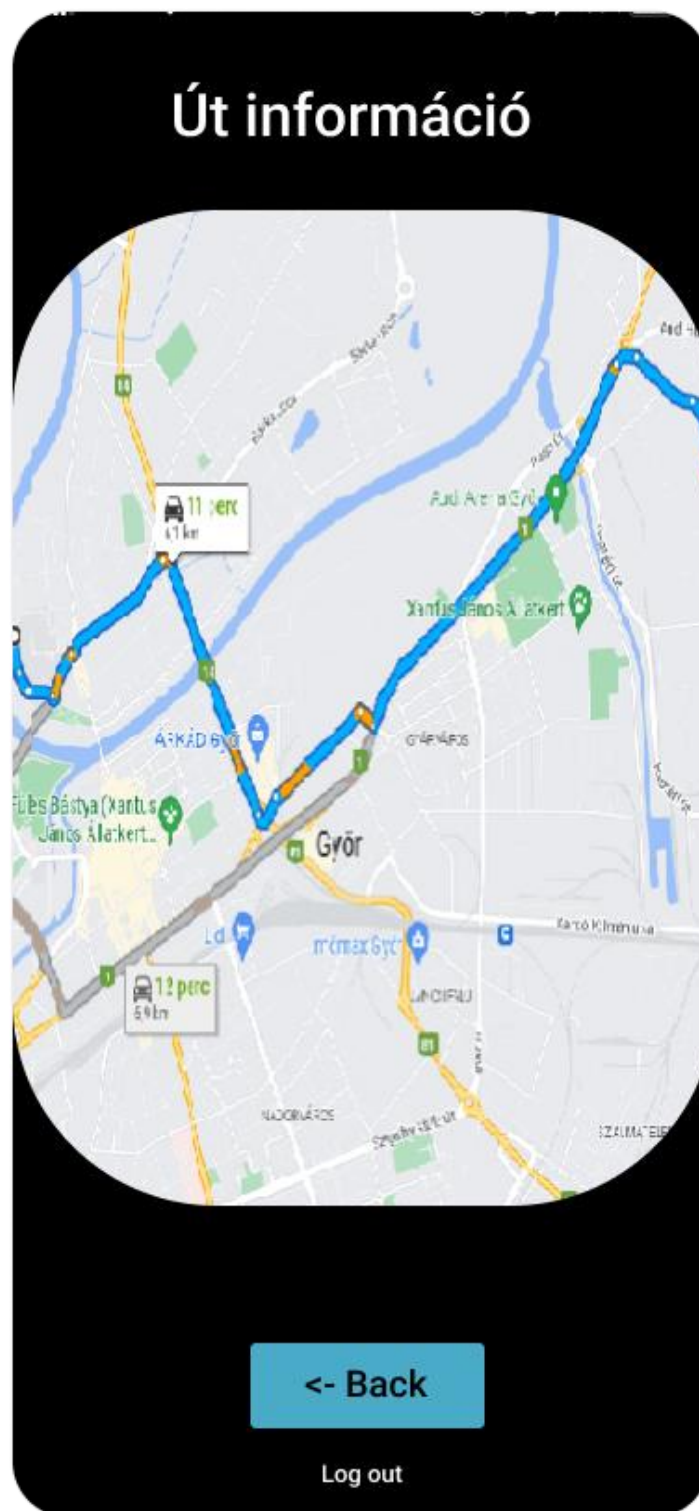
22. ábra - Az utasok ezen a felületen láthatják az aktuális utazásaikat



23. ábra - Az utas ezen a felületen tud jelentkezni az adott fuvarra



24. ábra - Az utas ezen a felületen látja az aktuális utazásával kapcsolatos információkat



25. ábra - Ezen a felületen tekinthető meg térképen az útvonal