

Szegedi Szakképzési Centrum Vasvári Pál Gazdasági és Informatikai Szakgimnáziuma

Az 54 213 05 számú Szoftverfejlesztő szakképesítés záródolgozata



**SMART TRUCK**

Készítette:

Szilvási Sándor

Szeged, 2020

## Tartalom

Bevezetés.....	3
1. A SZAKIRODALOM FELDOLGOZÁSA.....	4
1.1. Feladat meghatározás .....	4
1.2 Asztali alkalmazás .....	5
1.3 Webes alkalmazás .....	5
1.4 Adatbázis tervezés.....	6
1.4.1 Relációs adatbázis .....	6
1.4.2 Használati esetdiagramok, forgatókönyvek.....	8
1.4.2.1.Dolgozó .....	9
1.4.2.2 Főnök, admin.....	10
1.5 Megjelenítési tervek asztali és webes alkalmazásra .....	12
2. FEJLESZTŐI DOKUMENTÁCIÓ .....	13
2.1 Asztali alkalmazás .....	14
2.2 Az alkalmazás fájl térképe .....	14
2.3 Webes Alkalmazás.....	16
2.4 A program állományrendszere .....	18
2.5 Rendszerkövetelmény .....	19
3. FELHASZNÁLÓI DOKUMENTÁCIÓ .....	20
3.1 Asztali alkalmazás .....	20
3.2 Webes alkalmazás .....	21
3.3 Fejlesztési lehetőségek.....	23
3.3.1 Asztali alkalmazás Fejlesztési lehetőségek.....	23
3.3.2 Webes alkalmazás .....	24
Összegzés .....	25
Irodalmi jegyzék.....	26
KÖSZÖNETNYILVÁNÍTÁS .....	27

## Bevezetés

Záródolgozatom olyan cégeknek készült, akik az áru fuvarozásban vannak jelen. Azok számára, akik információkat szeretnének lekérni a dolgozóikról vagy kamionjaikról, azok műszaki adatairól<sup>1</sup> vagy az adott dolgozó útvonalairól. Annak érdekében, hogy ezt bármilyen nehézség nélkül meg tudják tenni, ezért létrehoztam a Smart Truck Kft alkalmazást.

A program legfontosabb célja, hogy pár pillanaton belül elérhessünk bármilyen információt a keresett kamionról és annak műszaki adatairól vagy a dolgozóról. Az alkalmazás sokkal könnyebbé teheti a mindennapokat. Gondoljunk csak bele, mennyi idő telik el azzal, mire egy kamionról kiderítünk minden információt, vagy épp a cég dolgozójáról vagy az esetleges útvonalakról, amelyeken fuvaroznak! Az alkalmazás segítségével, sokkal könnyebben hozzájuthatunk az információkhoz: merre tart épp egy kamionsofőr, a hónapban mennyit dolgozott vagy hány kilométert tett meg az utak során. A dolgozó adatai az alkalmazáson keresztül bármikor elérhetőek, ezáltal a cég telefonon vagy emailen értesíteni tudja a dolgozót a következő áruszállítással és annak esetleges változásaival kapcsolatban. Mivel nem csak az áruszállítók adatait látjuk, hanem a kamionok műszaki paramétereit is, ezért a cégek a későbbiek során költséghatékonyabban fuvarozhatnak, hiszen jobban át tudják gondolni, hogy adott útvonalra melyik kamiont indítsák, ezáltal csökkentve az üzemanyagfogyasztás költségeit. Tehát az alkalmazásom segítségével, sokkal kevesebb időt vesz igénybe a logisztikai feladatok megoldása.

A programban az “Esetleges új dolgozó” menüpontban lehet adatokat felvenni, vagy ha épp a dolgozó távozik a cégtől, akkor törölni. (Ebben az esetben a dolgozó eddigi útvonalai is törlődnek a programból.) Az “Esetleges új kamion” menüpont segítségével új járművet is hozzáadhatunk a már meglévő kamionok listájához és törölni is itt tudjuk őket.

Mivel a rendszert cégek számára terveztem, abban lehet az erőssége, hogy ezen a területen kevés ilyen szoftver található. Úgy gondolom, hogy a papír alapú rendszert le kell váltani egy gyorsabb és gördülékenyebb, modern technológiára. Véleményem szerint a jövő az olyan megoldásokban van, mint például az internetes alkalmazások és az asztali alkalmazások. Mivel

---

<sup>1</sup> Műszaki adat: Jármű adatait tartalmazó könyv mely kötelező, ha a járművet közúton használják

ezen manapság már sokak számára könnyen elérhetővé váltak, így a munkavégzést is nagyban megkönnyítik. Ezért is hoztam létre a Smart Truck Kft alkalmazást, mivel ez is a jövőt képviseli.

Mivel a rendszert cégeknek terveztem ezért én úgy gondolom, hogy a rendszernek, abba lehet az erőssége, hogy ez a terület még eléggé kiaknázatlan ezen a területen kevés ilyen szoftver található vagy nem is található és úgy gondolom, hogy a papír alapú rendszert le kell váltani a gyorsabb és gördülékenyebb modern technológiára. Vagyis a jövő ezekben az alkalmazásokban van, mint például az Internetes alkalmazásra és az asztali alkalmazás. Mivel manapság már mindenkinek van számítógépe és okostelefonja ezért bármikor elérheti az alkalmazást esetleg telefonról vagy számítógépről ezzel könnyítve a munkájukat az embereknek is. Ezért is hoztam létre a Smart Truck Kft alkalmazást mivel ez az alkalmazás a jövőt képviseli.

## 1. A SZAKIRODALOM FELDOLGOZÁSA

Jelen szakdolgozat a Microsoft Office 2016 irodai programcsomag MS Word 2016 kiegészítőjével készült. Az alkalmazásokhoz tartozó adatbázis adatok saját, kitalált rekordok. További, a szakdolgozat készítés során felhasznált programok, eszközök listája:

- <https://www.draw.io> - adatbázis tervezés
- <https://mockflow.com> - frame terv rajzolás
- GIMP 2.10.18 - logó készítés
- <https://www.pexels.com> - külső képek
- Visual Studio 2019 - asztali alkalmazás készítés
- Visual Studio Code – Weboldal készítése
- Google Drive - biztonsági mentések tárolása
- Git verziókövető rendszer - projekt követés
- Inkscape 0.92 - képek szerkesztése

### 1.1. Feladat meghatározás

A záródolgozatnak több része van. Egy asztali alkalmazás és egy böngészőben elérhető webes alkalmazás, melyek közös adatbázist használnak. Mivel a két alkalmazás eltér egymástól, ezért funkcióik is teljesen mások. Míg az asztali alkalmazáson tudunk adatokat

módosítani, törölni és hozzáadni, addig a webes részén csak a regisztrációnál történik hozzáadás, a többi részén pedig csak lekérdezni tudunk az adatbázisból, melyet megjelenít a webes felület. Mivel asztali alkalmazásomat szánom záródolgozatom fő részének, ezért arra fogok nagyobb hangsúlyt fektetni. A két alkalmazás a Smart Truck Kft nevet kapta.

## 1.2 Asztali alkalmazás

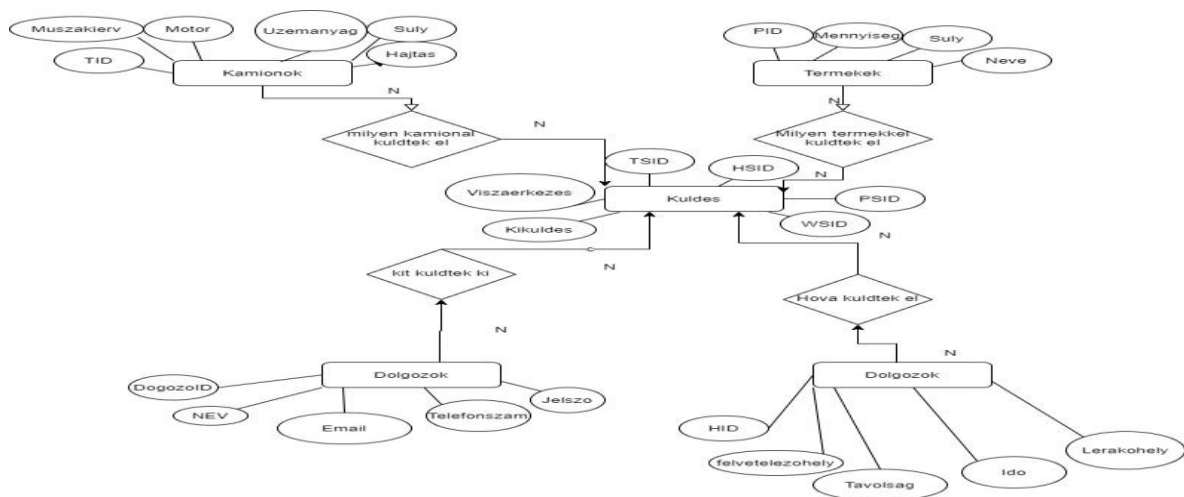
Az asztali alkalmazás funkcióját tekintve bizalmas információkat is tartalmaz, melyeket csak az admin vagy a cég vezetője láthat, ezért csak ez a két személy kap ehhez hozzáférést. Az alkalmazásban a dolgozók, kamionok adatait tudjuk módosítani, hozzáadni vagy törölni. Le tudjuk kérdezni még a dolgozók útvonalait, azt, hogy mikor, hová és mit szállított és ez mennyi időbe telt. Az adatok mindig a Datagridview-ban jelennek meg, egy grafikus felületű alkalmazásban. Az adatok kiválasztásával elérhető a módosítás, törlés, újadat felvitele funkció is. Ha ezeket módosítottuk, akkor már azonnal az új adatok lesznek elérhetőek a kamion adatbázisban.

## 1.3 Webes alkalmazás

Az adatbázisban már szereplő felhasználóknak a weboldalt megnyitva lehetőségük van belépni a lekérdezés fülre. Ez létrejöhetett az admin vagy a cégvezető által létrehozott adatokkal. Ha a dolgozó korábban már regisztrálta magát a weboldalon, akkor be tud jelentkezni ő is. Ellenkező esetben az oldalon lehetőség van regisztrációra, ahol a validációnak megfelelt adatok elküldése után tud belépni a dolgozó. Belépéskor a rendszerbe regisztrált összes dolgozót tartalmazó táblázatot látjuk, ahol adatokat tudunk lekérni a dolgozókról, a járművekről, a termékekről vagy útvonalakról. A webes felület az adatokat egy táblázatba gyűjti ki az adatbázisból. Ha az asztali alkalmazáson belül módosítunk valamilyen adatot és frissítjük a weboldalt, akkor a módosított adatokat látjuk. Vagyis ha asztalin belül módosítjuk például a felhasználó jelszavát, akkor már a módosított jelszóval tud majd belépni.

## 1.4 Adatbázis tervezés

A fejlesztést egy adatbázis megtervezése előzte meg. A következő ábra bemutatja a készített adatbázis működési sémáját:



1. ábra: Smart Truck Kft E-K diagramja.

Első feladatomban az alkalmazáshoz használt adatbázis megtervezése volt, vagyis egy E-K diagram készítése, melyet a tervezés folyamán többször is módosítanom kellett. Végül a tervezés után összeállt a végleges adatbázisom, melyet 5 táblásra terveztem. Főtábla, valamint ami összeköti mindent, a "Küldés" tábla. A hangsúly a "Küldés" táblán belül van, mivel abból bármilyen fontos adat elérhető. A megfelelő jogosultsággal rendelkező személyek módosíthatják az adatbázist. A főtábla és többi tábla között I:N kapcsolat van.

### 1.4.1 Relációs adatbázis

Az adatbázisom a 'Kamion' nevet kapta, melyben az egyedek és kapcsolatok szerkezeti felépítése és a tulajdonságok a következők szerint alakultak:

Egyedek:

- Dolgozók
- Hely
- Kamion
- Küldés
- Termékek
- 

Tulajdonságok

- Dolgozók: **id**, nev, telefonszam, email, jelszo
- Hely: **híd**, Felvetelezohely, Tavolsag, ido, Lerakohely
- Kamionok: **tid**, Muszakierv, Rendszam, Motor, Uzemanyag, Suly, Hajtas
- Küldés: **Sid**, kid, heid, teid, did, Kikuldes, Visszaerkezes --KapcsolóTábla
- Termékek: **teid**, Mennyiseg, TermekSuly, TermekNeve

A Dolgozók tábla az összes, a rendszerben regisztrált dolgozót tárolja vagy az asztali alkalmazásban létrehozott dolgozók azon adatait mutatja, amivel be tudnak lépni az alkalmazásba.

A Hely táblában az indulás és az érkezés helye, a szállítási idő és a megtett távolság (kilométerben számolva) láthatók.

A Kamionok tábla a kamionok műszaki adatait tárolja el, például, hogy mikor jár le a műszaki vizsgálja a kamionnak. Ezeket az információkat le tudja kérdezni az alkalmazott és a főnök is.

A Küldés tábla A kapcsoló tábla ez köt, mindet össze innen lehet, az útvonal adatai majd le kérdezni dolgozók alapján.

A Termékek táblában található a szállított termékekről tárolt adatok: nevük, súlyuk, mennyiségük.

Kapcsolatok:

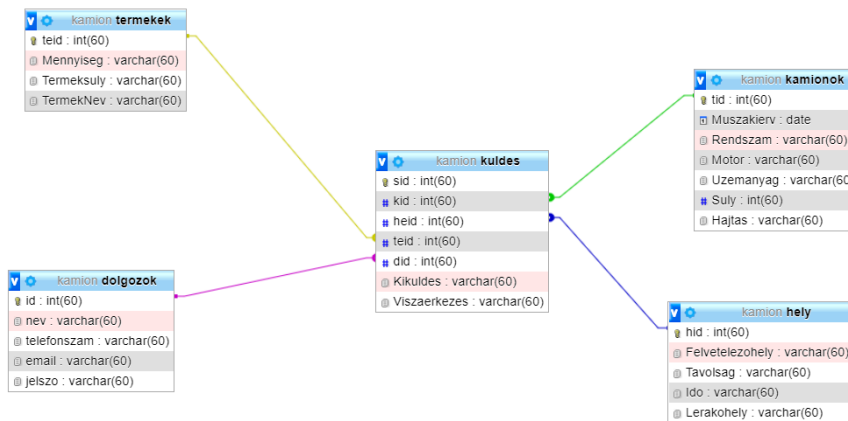
Kuldes\_Dolgozok(Id,did)

Kuldes\_Termkek(teid,teid)

Kuldes\_Hely(hid,heid)

Kuldes\_Kamionok(tid,kid)

Az adatbázis felépítésén látható (2. ábra), hogy az elsődleges kulcs (PRIMARY\_KEY) mely tábla idegen kulcsával (FOREIGN\_KEY) van kapcsolatban. Az adatbázis felépítése során a "Küldés" tábla lett a kapcsoló tábla, amely a többi táblához az idegen kulcsok segítségével kapcsolódik. Minden tábla külön tárolja az adatokat és ezeket az adatokat jeleníti meg a küldés tábla. A táblák a "Küldés" kapcsoló táblán kívül egymással nem állnak közvetlen kapcsolatban. A táblázatok egy a többhöz kapcsolatban állnak egymással, vagyis több adat kapcsolódik az egy táblához, vagyis a kapcsoló táblához, amelyet a Küldés táblaként láthatunk az ábrán amely így néz ki:



2. ábra: Relációs adatbázis séma.

#### 1.4.2 Használati esetdiagramok, forgatókönyvek

Az alkalmazást használó személyeket tudjuk kategorizálni és jogosultságokat tudunk adni számukra. Vannak az egyszerű felhasználók, az alkalmazásban ez a dolgozót jelenti. Vannak a főnöki jogosultsággal rendelkezők. Ők mindent adathoz hozzáférnek és tudják is azokat módosítani. Admin jogosultsággal rendelkeznek, adatokat tudnak törölni, módosítani, hozzáadni, de adatbázist és táblázatokat nem tudnak törölni. Végül vannak a rendszergazda jogosultsággal rendelkező személyek, akik tudnak adatbázist és táblát is törölni.



#### 1.4.2.1.Dolgozó

##### **Belépés:**

A dolgozó be tud jelentkezni az asztali alkalmazásba és a webes alkalmazásba is, ahol ezek a forgatókönyvek lehetségesek:

Rendelkezik a belépéshez szükséges adatokkal.

##### *Hibalehetőség:*

Elfelejtí vagy rossz adatokat ad meg (email cím és jelszó) és ezáltal nem tud belépni a rendszerbe.

##### *Megoldás:*

Felveszi a kapcsolatot a rendszergazdával vagy az adminnal személyesen, emailen keresztül vagy telefonon. Ha a rendszergazda esetleges leterheltség esetén nem áll rendelkezésre, akkor engedélyezi az adminnak az új jelszó generálását az asztali és webes alkalmazásban, majd eljuttatja az illetékes személynek, aki az új jelszóval be tud lépni az alkalmazásba.

##### **Adatlekérdezés**

Bejelentkezés után az asztali alkalmazásban információt tud lekérni az egyes dolgozókról, útvonalakról vagy kamionokról.

Jó esetben a dolgozó megfelelő adatokat kér le és valós információhoz jut az alkalmazás segítségével.

##### *Hibalehetőség:*

Nem megfelelő információt kér le vagy kap vissza, esetleg hibába ütközik a lekérdezés során.

##### *Megoldás:*

Felveszi a kapcsolatot a rendszergazdával vagy az adminnal személyesen, emailen keresztül vagy telefonon. A rendszergazda felveszi a hibát és elkezd dolgozni rajta. Ha leterheltség esetén nem áll rendelkezésre, akkor az adminnak tudja továbbítani a hibát, aki ha nincs leterhelve, elkezd a hibát kijavítani.

#### 1.4.2.2 Főnök, admin

### **Belépés**

A főnök és az admin be tud jelentkezni az asztali alkalmazásba és a webes alkalmazásba is, ahol ezek a forgatókönyvek lehetségesek:

Rendelkezik a belépéshez szükséges adatokkal és be tud jelentkezni hiba nélkül.

#### *Hibalehetőség:*

Elfelejtí vagy rossz adatokat ad meg (email cím és jelszó), ezáltal nem tud belépni a rendszerbe.

#### *Megoldás:*

Felveszi a kapcsolatot a rendszergazdával vagy az adminnal személyesen, emailen keresztül vagy telefonon. Ha a rendszergazda esetleges leterheltség esetén nem áll rendelkezésre, akkor engedélyezi az adminnak az új jelszó generálását az asztali és webes alkalmazásban, majd eljuttatja az illetékes személynek, aki az új jelszóval be tud lépni az alkalmazásba.

### **Szerver kapcsolódási hiba**

#### *Hibalehetőség:*

Helyes belépési adatok esetén sem tud az asztali alkalmazásba belépni a főnök vagy az admin.

#### *Megoldás:*

Lehetséges hálózati hiba a rendszerben vagy az adott személynek nincs internet elérése.

## Adatfelviteli hiba

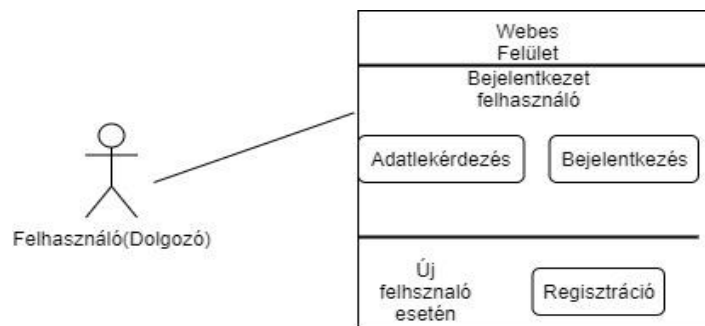
### *Hibalehetőségek:*

Üres mezőket szeretne felvenni az adatbázisba vagy esetlegesen hibás adatot szeretne feltölteni.

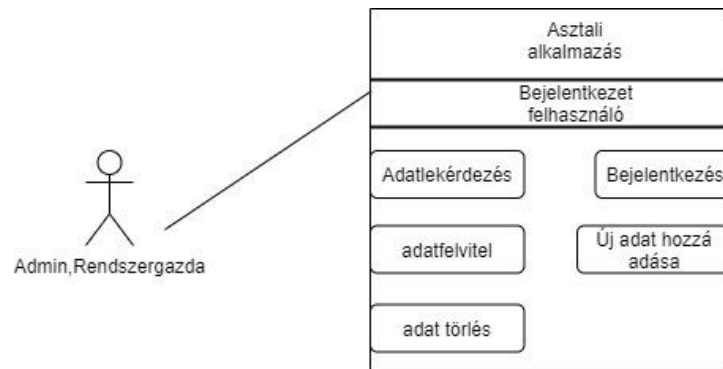
### *Megoldás:*

Egy grafikus megjelenítő ablak fogja ezt jelezni: a rendszer nem engedi az üres rekord létrehozását és hibás adat felvitelét sem az adatbázisba.

A 3. és 4. ábra szemlélteti, hogy a jogoknak megfelelően milyen hozzáférésük van a dolgozóknak, vagyis a felhasználóknak és az admin rendszergazda jogosultsággal rendelkező személyeknek a rendszerhez. A felhasználói szinten lévő személyeknek ugyanannyi jogosultságuk van az asztali alkalmazásban, mint a webes felületen és természetesen az admin jogosultsággal rendelkező személyeknek is.



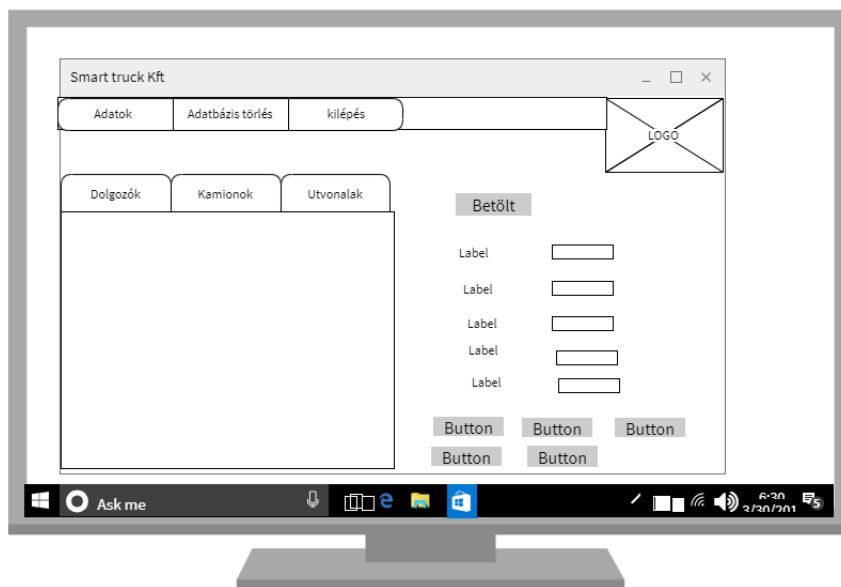
3. ábra: Use case diagram felhasználói szinten.



4. ábra: Use case diagram admin vagy rendszergazdai szinten.

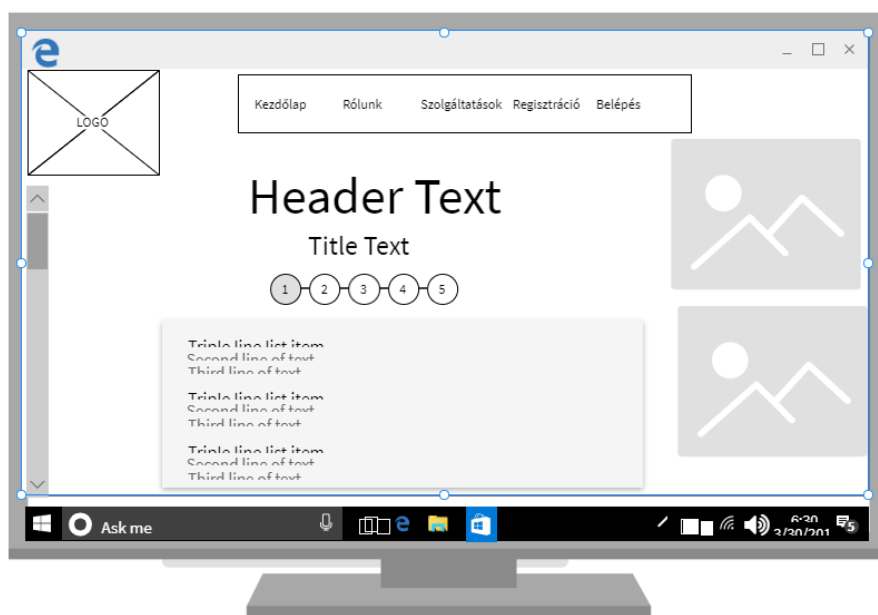
## 1.5 Megjelenítési tervek asztali és webes alkalmazásra

Az asztali alkalmazásom (5. ábra) és webes alkalmazásom (6. ábra) létrehozása során is törekedtem arra, hogy az mindkét felületen felhasználóbarát legyen, vagyis könnyen kezelhető és kényelmes legyen azt használni. Az asztali alkalmazásnál törekedtem a letisztultságra, hogy a felhasználó ne bonyolódjon bele a program használatába, vagyis egyszerűen kezelhető felületet lásson, és azt könnyen tudja kezelni.



5 ábra: Grafikus terv az asztali alkalmazáshoz.

A webes alkalmazásomnál nagyon figyeltem, hogy az oldal reszponzív legyen, hisz manapság ez alapkövetelmény. A weblapomat próbáltam a modern elvárásoknak megfelelően megírni és fejleszteni: bármilyen eszközről elérhető legyen és minden népszerű böngésző használatát támogatja. Megjelenését tekintve template-eket és animációkat használ, ezáltal színesebbé téve a weboldalt, de törekedtem a túl sok animáció kerülésére a letisztultság megtartása érdekében. Alapértelmezetten magyarul jelenik meg az oldal és külön fűleken érhetőek el a főbb funkciók.



6 ábra: Grafikus terv az asztali alkalmazáshoz.

## 2. FEJLESZTŐI DOKUMENTÁCIÓ

A helyi hálózathoz való csatlakozást és tesztelést a XAMPP Control Panel, az Apache/3.2.4 (Win64) webkiszolgáló és a PHP 7.2.28 verziója segítette. Az adatbázisomat a phpmyadmin SQL felületén hoztam létre, amit a 10.4.11-MariaDB adatbázis-kiszolgáló jelenít meg.

## 2.1 Asztali alkalmazás

Az alkalmazás Windows 10 operációs rendszeren lett fejlesztve, Visual Studio 2019-ben. A program fejlesztése során az OOP (objektumorientált) programozás szabályait betartottam. Az alkalmazást úgy fejlesztettem, hogy a felhasználó igényeit kielégítse. Fontosnak tartottam, hogy felhasználóbarát program legyen, ezért az egész egy grafikus felületű szoftver lett, amelynek a kezelése nagyon egyszerű. A betanulási ideje nem vesz többet igénybe 10-15 percnél. Ez az idő rövid, amely manapság elvárás. Igen megbízható, mivel minden biztonsági elvárásnak eleget tesz. Az adatbázisokban tárolt adatok külön szerveren vannak eltárolva egymástól. Ha adatlopás történik, nem minden adat szivárog ki. A szoftvernek szüksége van stabil internetkapcsolatra, hogy tudjon az adatbázishoz csatlakozni és elvégezni a kért műveleteket. A szoftver esetleges hiba esetén jelzi a meghibásodást a felhasználónak, aki továbbítani tudja azt az adminisztrátornak. Karbantartások esetén a szoftver üzenetet küld a felhasználóknak, hogy karbantartás alatt van a szoftver és ezért nem elérhető.

## 2.2 Az alkalmazás fájl térképe

Az alkalmazás betartja az objektum orientált programozás szabályait, ezért a fájlokat különböző mappákba csoportosítottam, melyek a következők:

### **Modell**

A mappán belül található egy connectinSting.cs nevű osztály, melynek a feladata—az adatbáziskapcsolat létrehozása. Továbbá itt hozom létre a programhoz szükséges többi osztályt is, amelyet partial classsal oldottam meg. A mappán belül megtalálható az osztályok database része is, amelyek mindegyik osztálynak le van gyártva. Itt történik az adatbázisban a sorok listázása és törlése is.

### **MyException**

A mappában az összes validálás execeptionje található.

## Repository

A mappában több almappa is található, amelyek mindegyikében van egy Repository.cs fájl. Ez partial class-szal kapcsolódik a Repository1.cs nevű fájlhoz. A Repository.cs a fájl, ami azért felelős, hogy létrehozza a listát és egy táblát készítsen a listából. Van-minden mappában egy RepositoryDatabaseTable.cs nevű mappa, aminek a fő feladata, hogy beállítsa a konstruktorok kezdő értékét, és a táblát létrehozza a teszt adatoknak. Itt hívódik meg az adatok mappa tábla törlése és teszt adatok törlése is. Található ezen kívül még a RepositoryTableSql.cs nevű fájl, ami partial class-szal kapcsolódik a RepositoryDatabase.cs fájlba. A RepositoryTableSql.cs fájl feladata, hogy visszaadja az adott mappa értékeit, törölje az adott értéket vagy új adatot szűrjön be, esetlegesen módosítsa azt. Található még a mappákban egy RepositoryTableTestData.cs nevű fájl, ami partial class-szal kapcsolódik a RepositoryDatabase.cs fájlba. A RepositoryTableTestData.cs fő feladata, hogy az adott mappában lévő tesztadatokat feltöltse az adott mappa táblájába. A mappában található egy Repository1.cs nevű fájl, amihez minden almappa Repository.cs nevű fájlja kapcsolódik. A fő feladata a Repository1.cs nevű fájlnak, hogy létrehozza a listákat a Repositoryban. - Itt található még egy RepositoryDatabase.cs fájl, amely azért felelős, hogy létrehozza a kamion adatbázist vagy törölje azt.

## Sql

Itt található a Kamion.sql adatbázis.

## View

Ebben a mappában találhatóak grafikus felületek. Ahol létrehoztam a tartalmat adó formokat, hogy könnyen áltassam program felépítését megjelenésileg. Legelőször is elhelyeztem a login Formon A buttonokat a labeleket és textboxokat később a jelszó tesztboxát tikosítottam hogy ne legyen látható a jelszó. A Fő Formom a következő elemekből áll össze:

- menuStrip ami a menü kezelését szolgáltatja
- tabControl, amely a fő menürendszert szolgáltatja
- ezen textBox-ok, dataGridView-k, button-ok elhelyezve

- panel (az adok felvételének elfedése)
- Picture box (logó helye)
- background image (Háttér feltöltése képpel)

A fő formon elhelyeztem ErrorProvidereket, melyek hibás adat megadásakor jelzik a hibát. Megtalálható még egy toll stripLabel is, ami adatbázis hiba esetén jelez.

A fő formon a tabpage segítségével tudunk navigálni az egyes részek között. Több lehetőség áll rendelkezésünkre: dolgozói adatokat lekérni, módosítani vagy új adatokat felvinni, meglévő kamiont törölni vagy új járművet hozzáadni a céghez. A dolgozó eddigi útvonalait is le tudjuk kérdezni (7. ábra).

7 ábra: Dolgozói utvonala lekérdezés.

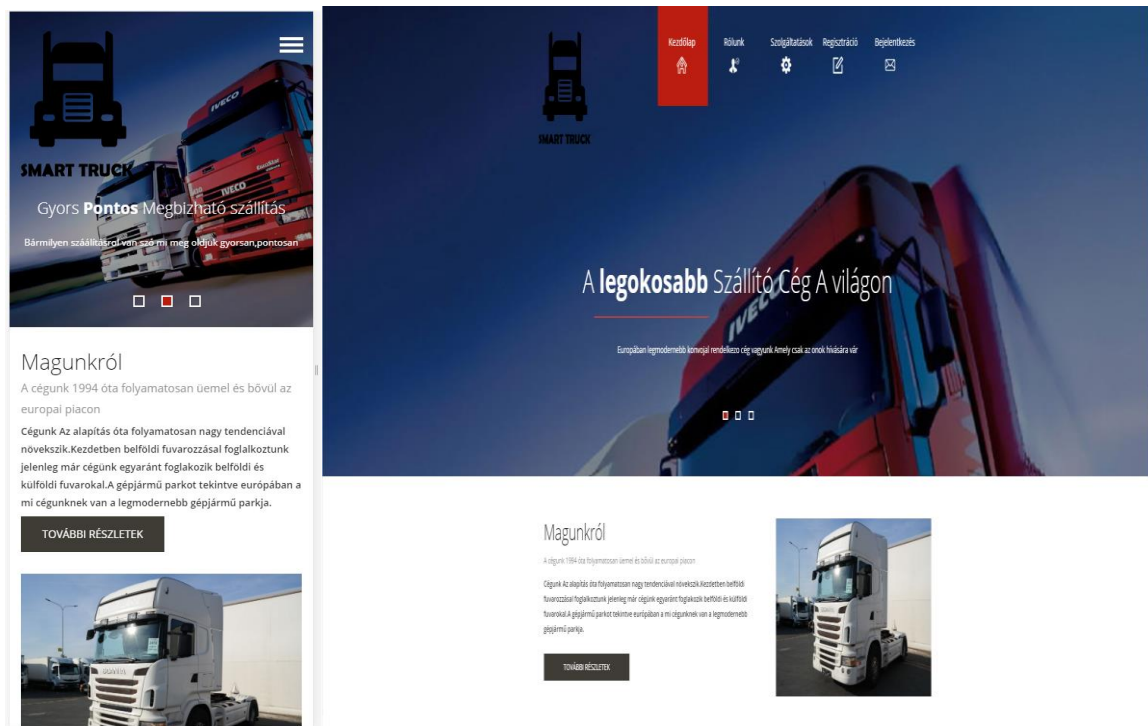
## 2.3 Webes Alkalmazás

A szakdolgozatomban web böngészőben hozzáférhető részét Visual Studio Code keretrendszerrel valósítottam meg. Munkám során egy template segítségével vettem alapul melyet



át szerkesztettem a saját weboldalamnak meg felelően a Html fájl kiterjesztést át írtam phpra ezek kívül még hozzá csatolt saját tudásom alapján Javascript, jQuery és PHP programozási nyelveket. Azért Döntöttem egy Template mellet mivel a fő arca a projektemnek az asztali alkalmazás és a webes csak a plusz melléklet mellé. A templatet saját tudásom alapján hozzá csatoltam egy login rendszert mely az adatbázishoz csatlakozik, és abból kéri ki az adatokat bejelentkezéskor.

A weblaphoz készítettem még egy Regisztrációs fület is, amely sikeres regisztráció esetén hozzá adja az adatbázishoz a regisztrációt. A bejelentkezés után a weboldal át dob egy adatlekérés fültre ahol adatokat tudunk le kérdezni. A Weboldal készítése során mé segítségül vettem a Bootstrapet. Manapság már elvárás hogy a felhasználók vagy, akik a weboldra tévednek azoknak olvasható formátum weboldal biztosítsunk, vagyis bármilyen eszközről látható és olvasható legyen, vagyis ha reszponzivnak kell lennie, amit én Bootstrap segítségével oldottam, meg ami azt jelenti, bármely eszközről használjuk az oldalt az olvashatóságából nem fog veszíteni, mert alkalmazkodik a használt eszközhöz.



8 ábra: Telefonos nézet asztali nézet.

A weboldal készítése során még törekedtem arra, hogy minél egyszerűbben tudjon tájékozódni a felhasználó vagy a látogató. Fontos volt számomra, hogy a program nyelve magyar legyen. Ügyeltem arra, hogy a weboldal bármilyen böngészőből elindítható és bármilyen eszközön használható legyen (például: okostelefon, tablet).

## 2.4 A program állományrendszere

A programozás során próbáltam a tiszta kód szabályait követni. Igyekeztem megvalósítani egy átlátható állományrendszert, amiben a fájlok funkcionálisan vannak szétválasztva a megfelelő navigációs utakkal felcímkézve. A fájlrendszerem a következő:

### Css

A weboldal stílusjegyeiért felelős állományokat tartalmazza, mely saját és integrált kódokat is használ. Ebben a mappában található találhatók még az animációkért felelős kódok. Itt található a bootstrap is, amely a kinézetért és a reszponzivitásért felelős. Itt van a TouchTouch fájl is, amely azért felelős, hogy a kiválasztott menüt kiemelje.

### Img

Ebben a mappában található a weboldalon látható képek, mint például a háttérkép. A logó kép is itt található, amelyet én készítettem. A weboldalon található olyan képek, amelyeket a weboldal alapból használt, azokat én is használatba vettem és nem változtattam rajtuk. Vannak olyan képek, melyek külső hivatkozásból származnak (lásd: Irodalomjegyzék Hivatkozások c. fejezet).

## JS

Ebben a mappában található a JavaScript fájlok, amelyek különböző funkciókért felelősek (a képek méretezés vagy a kép reszponzivitása). A javascript legtöbb része nem saját készítésű, hanem a template-hez használt szkripteket használtam.

## PHP

Mivel az egész weboldalam php alapú, ezért elég a háttérben funkciókat kezel, közvetlen kapcsolatban áll a kapcsolódott adatbázissal és a szerveroldali ellenőrzések is itt íródtak. Az adatbázishoz való csatlakozást egy connect.php nevű fájlal oldottam meg, ami azért felelős, hogy kapcsolódjon az adatbázishoz és létrejöjjön a kapcsolata. Php-én belül kezelem le még a regisztrációt és a belépést is, amelyeket úgy oldottam meg, hogy ha regisztrál a felhasználó, akkor, ha minden adatot helyesen adott meg, egy register.php-vel beszúrja a helyes adatokat az adatbázisba. A belépést pedig egy adatbázisból való Select utasítással oldottam meg, mely belépéskor lekéri az adatokat és ellenőrzi azok helyességét.

## 2.5 Rendszerkövetelmény

Mivel a szoftver Microsoft Visual Studiot 2019-ben íródott nem igényel ennél az alkalmazásnál nagyobb gépigényt. Ezért úgy döntöttem, hogy a Microsoft Visual Studiot gépigényét fogom alapul venni, amelyek a következőképpen alakulnak:

Rendszerkövetelmény:

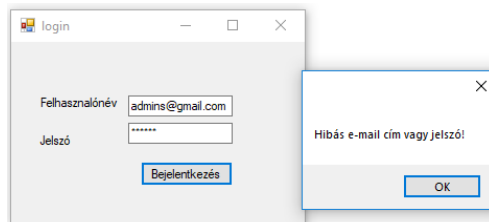
- Windows operációs rendszer (Vista, Windows 7, Windows 10)
- 1,8 GHz vagy gyorsabb processzor Dual-core vagy Jobb.
- 4 GB RAM vagy több.
- 130 GB szabad hely a merevlemezén.

### 3. FELHASZNÁLÓI DOKUMENTÁCIÓ

#### 3.1 Asztali alkalmazás

Az alkalmazás fejlesztése Visual Studio 2019-ben történ azon elül is A Net Frameworkben történt melyet az újabb generációs Windows operációs rendszerek tartalmaznak így az asztali alkalmazás futtatása lehetséges Vista, Windows 7, Windows 8 és Windows 10 operációs rendszereken egyaránt. Az alkalmazás elindításához szükségünk van, az adatbázisra Melyet az Adatbázis mappán belül egy kamin.sql állományban találunk, melyet be kell importálni az alkalmazás indításához. Az alkalmazás elindításához szükségünk lesz az indító fájlra, amit egy telepítő mappából indíthatunk el vagy kimásolhatjuk azt a mappából. után egy belépés ablak fog meg jeleni. Ha rendszerben megfelelően gépeltük be az adatokat, akkor az be fog engedni ás át dob a programra, ha hibás, akkor Hibát fog ki írni (8 ábra):

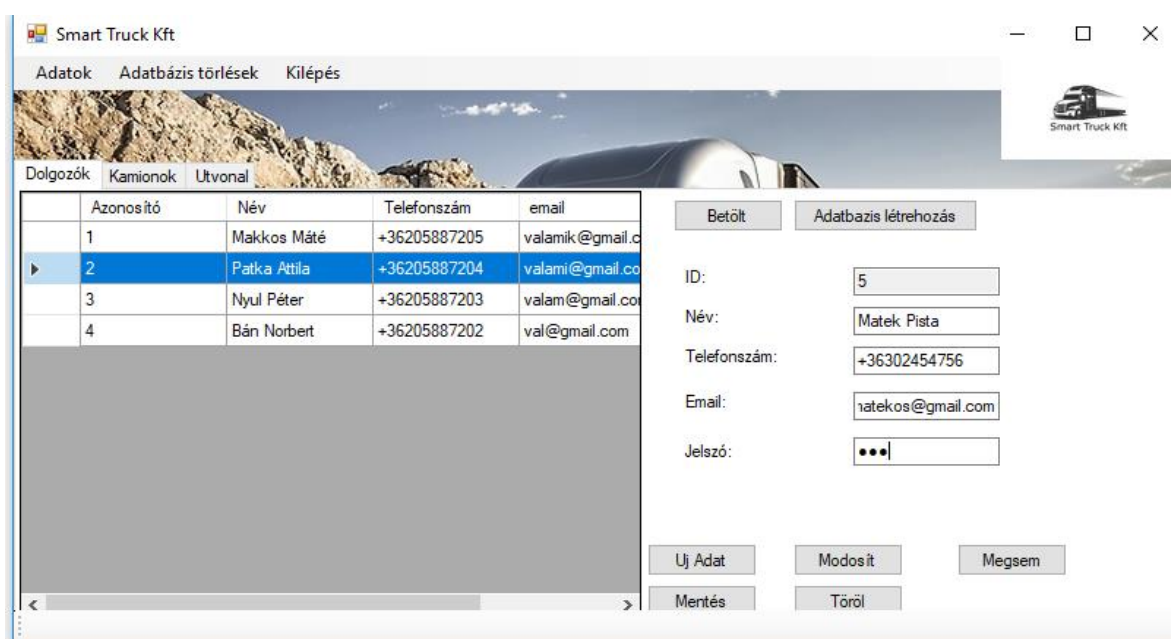
„Hibás Email-cím vagy jelszó!”



9 ábra: Hibás bejelentkezés

Sikeres belépés esetén megjelenik a fő modul. Itt egy vízszintes menüben helyezkednek el a menüpontok. A menüpontok alatt ki tudjuk választani hogy milyen cselekvést szeretne a felhasználó csinálni a tabpageken. A Tabpageken három rész közül tud választani a felhasználó van egy Dolgozó lap ahol a dolgozó adatait tudja módosítani törölni vagy új adatot felvinni. Található egy Kamion fül ahol egy kiválasztott Kamion adatait tudjuk módosítani és törölni vagy új kamiont hozzá adni az adatbázishoz. Az utolsó tab page pedig a dolgozó eddigi útvonalait láthatjuk ahol csak lekérdezés lehetséges dolgozók szerint. A három lap közül két

lap rendelkezik egy 'Új', 'Módosítás', 'Törlés' gombbal. Új adat felvitelkor megjelennek a textboxok ahová adatokat tudunk fel vinni . Az adatok beszúrás és adatmódosítás esetén is ellenőrzésre kerülnek, tehát, ha bármely mező üresen marad, a mentés nem engedélyezett és nem tölti fel az adatokat az adatbázisban. Ha szeretnénk az adatokat dolgozni először is szükséges az adatok betöltése amelyre van egy Betöltés gomb. Ha az adatok sikeresen betöltődtek akkor kiválaszthatjuk mely dattal szeretnénk dolgozni. Mindig elsőként szükséges jelölni a táblázatban módosítani vagy törölni kívánt elemet, új rekord felvitelkor értelemszerűen ez nem szükséges (10. ábra).



10 ábra: Új adatfelvitel, asztali alkalmazás.

### 3.2 Webes alkalmazás

Ezt az alkalmazást a számítógépre telepített webböngészőben nyitjuk meg az 'index.php' fájl segítségével. A meglévő MySQL kiszolgálón és a beimportált adatbázison kívül másra nincs szükség. Bármely operációs rendszeren, ami rendelkezik web böngészővel és internet hálózattal, elérjük a weboldalt. Érdekes minimum kettő fajta böngészőt használni, mert a biztonság érdekében egy webkeresőben egy IP címen belül egyszerre egy felhasználó tud csak

belépni. Tesztelés szempontjából két különböző webböngésző alkalmazása szükséges. A nyitóoldalt kezelő 'index.php' egy azonosító alapú hivatkozásokkal létrehozott landing page. A felső menüsor mindegyik eleme ugyanazon oldal különböző helyeire navigál. Az utolsó fül továbbít a bejelentkezés fülre, ezen keresztül lehet belépni az adatlekérdező felületre. Hibás belépési adatok esetén a rendszer kiírja, hogy helytelen jelszó vagy email cím.

A weboldalon információ található az alkalmazásról a 'Rólunk' hivatkozásnál. A 'Szolgáltatások' fül megmutatja, milyen szállításokkal foglalkozik a cég. A 'Regisztráció' fülön (11. ábra), ha korábban az oldalt látogató ügyfél vagy dolgozó még nem kapott felhasználói jogosultságot, illetve nem került be a rendszer adatbázisába, lehetőség van a regisztrálásra. Az adatok helyes megadása és belépés után a felhasználó egyből használni tudja az alkalmazás további szolgáltatásait. Minden mező kötelezően kitöltendő. Elírás, helytelen szövegformátum esetén a mezőből való kikattintás után egyből hibaüzenetet mutat a program, mely alapján korrigálni tudja a felhasználó a felvitt adatait. Ellenőrzés email megadásakor: meghatároztam a várt email címet, melynek tartalmaznia kell betűket, egy '@' jelet, majd az utána begépeltek szövegnek tartalmaznia kell egy pontot is. Érvényes jelszó esetén négy kikötést is szerepeltetek, mely szerint annak tartalmaznia kell kisbetűt, nagybetűt, számot, illetve legalább 8 karakter hosszúságúnak kell lennie. Illetve a névnél nagybetűvel kell, hogy kezdődjön a név. Ha ezek mind teljesülnek akkor sikeres regisztráció történik.

SMART TRUCK

Kezdőlap Rólunk Szolgáltatások **Regisztráció** Bejelentkezés

Név:

Telefonszám:

Email:

Jelszó:

Jelszó megismétlése:

Regisztráció

Kezdőlap Rólunk Szolgáltatások Regisztráció Bejelentkezés

**CégHelye**

Csurgód megei 6775  
Kisbörzsöny  
Kör. útja 124-128

+3620887203

SMART TRUCK  
A Világ legmegbízhatóbb szállítói cége  
Az egész Földön

© 2015 Truck. Design by W3layouts

10 ábra: Regisztrációs menü a weboldalon.

Ha a felhasználó sikeresen bejelentkezett, rögtön látható a lekérdezés fül, ahol kiválasztva a megfelelő táblázatot egy rendezett táblát kapunk. Egyszerre csak egy adat lekérdezést lehet megvalósítani a weboldalon. A belépett felhasználó által látott menü megváltozik a kijelentkezett állapothoz képest. Itt található meg a 'Lekérdez' hivatkozás. Ez már nem az oldalon belül, hanem a weboldal más helyén helyezkedik el.

Későbbiekben a jelszavakat egy külön adatbázisban szeretném tárolni. ezáltal is biztonságosabbá téve a weboldalamat az adatlopással szemben.

### 3.3 Fejlesztési lehetőségek

#### 3.3.1 Asztali alkalmazás Fejlesztési lehetőségek

Mivel a fejlesztés során sok mindenre nem jutott idő, ezért azokat a későbbiekben próbálom meg kivitelezni, ezáltal tovább fejleszteni az asztali alkalmazásomat. A legfőbb fejlesztési lehetőség úgy gondolom az, hogy az alkalmazást platform függetlenné lehetne tenni, mivel most csak Windows platformon-használható.

Funkciók terén is lehetne bővíteni például, szeretném majd belevinni, hogy a dolgozók is tudjanak majd adatokat módosítani vagy lekérni. Szeretném a bér lekérdezését is, így az egyes dolgozók meg tudják nézni a havi bérüket, hogy milyen levonások történtek vagy épp milyen plusz juttatást kaptak. Tervezem még a szabadságok felvételét, hogy a dolgozók tudják jelezni az alkalmazáson keresztül, hogy mikor szeretnének szabadságra menni és az alkalmazottaknak mutassa a meglévő szabadnapok számát.

Vagyis jelenleg még sok fejlesztés vár az asztali alkalmazásra, melyek összefoglalva a következők:

- platform függetlenné tenni-
- szabadságtervező hozzáadása

- bér lekérdezése
- jelszavak kezelése külön adatbázisban

### 3.3.2 Webes alkalmazás

Mivel a szakdolgozatom fő részének az asztali alkalmazást terveztem, ezért a webesen sok továbbfejlesztési lehetőséget látok még. A webes alkalmazásomhoz szeretnék még több fület -és több választási lehetőséget adni, mivel jelenleg csak információt lehet lekérdezni. A későbbiekben a weblapom fejlesztése során szeretnék hozzárendelni egy szabadságtervező felületet, hogy az alkalmazottak a belépést követően meg tudják tervezni előre a szabadságukat, amelyet a vezetőség el tudna majd bírálni. Szeretnék még egy bérlekérdezést is megvalósítani, ahonnan a dolgozó le tudná tölteni az adott havi bérezését.

Kiegészíteném a weboldalamat egy nyomon követés füllel, ahol láthatná a vezetőség, hogy merre jár a járművezető a szállítmányozás során. Ezáltal a főnök nyomon tudná követni, hogy milyen történik az áruszállítás.

A dolgozóknak készítenék egy munkabeosztás fület, ahol majd látná a következő havi vagy az aktuális havi beosztását a weboldalon és ki is tudná azt nyomtatni. Összefoglalva a főbb fejlesztési lehetőségek a webes felületen:

- bér lekérdezés
- nyomonkövetés térképen
- szabadságtervező
- beosztások kezelése, letöltése



## Összegzés

Záródolgozatom összegzéseként elmondhatom, hogy a projekt elején kitűzött fő cél elértem, az alkalmazásomban elkészültek az eltervezett funkciók.

Voltak olyan részek, amelyek fejlesztés során jutottak eszembe, ezeket is hozzáillesztettem vagy épp átalakítottam az alkalmazást miattuk. Az alkalmazásom szépen épült a hónapok során, melyhez kaptam tanári javaslatokat és segítséget. Az alkalmazás fő célja az adatok lekérdezése, módosítása és új adatok felvitele. Mivel fő alkalmazásként az asztali alkalmazást tekintettem a záró dolgozatom fő céljának, így ezzel foglalkoztam többet és építettem fel részletesebben. A fejlesztésben nagy nehézségeket okozott az alkalmazás megtervezése és kivitelezése, de ahogy elkezdett kialakulni bennem a kinézete és működése, egyre könnyebb lett tovább építeni és megvalósítani. A fejlesztés során 3 részre bontottam a projektem. Első része volt a kinézet megvalósítása, második része a backend kódolása és az adatbázishoz való csatlakozás megírása, harmadik része pedig a login rendszer felépítése. Az alkalmazás felépítése során sokszor futottam abba a hibába, hogy túl sokat szeretem volna vállalni, ezért mindig oda kellett figyelnem, hogy tartsam az előzetes tervekhez magam. Miután elkészítettem az asztali alkalmazásom, utána jött a webes része, melyhez egy template-t használtam, amelyet átírtam a saját weboldalamhoz. A weboldalhoz kellett építenem egy regisztrációs, egy belépési és egy adatlekérdezés fület. Mivel php tudásom hiányosabb volt, ezért ezzel sokkal többet kellett foglalkoznom. Az eddig tanultakat bele tudtam építeni a weblapba és alapul tudtam őket venni. Legtöbbet az adatbázisomban kellett módosítani, mert mindig volt valami, amivel jobban tudtam. A fejlesztés során egyértelművé vált számomra, hogy a legfontosabb lépés mindig a tervezés. Ahogy épült az alkalmazás és a webes felület, úgy kezdett egyre jobban összekapcsolódni ez a két rész.

Hálás vagyok, hogy szaktanáraink segítséget nyújtottak, amikor elakadtam és próbáltak rávezetni a hibákra. Úgy gondolom, a motivációm abba rejlett, hogy tudtam, ezzel a záródolgozattal fejleszthetem magam és még több tudást szerezhetek. A fejlesztés minden percét élveztem, mert tudtam, hogy ezzel a saját tudásom gyarapodik majd és egy olyan programot hozok létre, amely sok ember életét segíti és megkönnyítheti majd.

## Irodalmi jegyzék

1. <https://www.pexels.com>
2. <https://w3schools.com>
3. <https://www.iconsdb.com/white-icons/>
4. <https://w3layouts.com/truck-a-auto-mobile-category-flat-bootstrap-responsive-web-template/>
5. Órai munka során készített papíralapú jegyzet

## KÖSZÖNETNYILVÁNÍTÁS

Ezúton szeretnék köszönetet mondani Gyuris Csaba és Bálint Róbert témavezető szaktanárainak, aki szakértelmükkel, munkájukkal és hasznos magyarázataikkal a konzultációk során biztosították a segítség nyújtást és tanácsaikkal hatalmas segítséget nyújtott szakdolgozatom elkészüléséhez.