

ZÁRÓDOLGOZAT

Dániel Ádám Zoltán

2020

Járműkezelő szoftver

Dániel Ádám Zoltán

SZÁMALK-Szalézi Szakgimnázium

Szoftverfejlesztő

Konzulens: Kaczur Sándor

Nyilatkozat
a záródolgozat eredetiségéről

Alulírott Dániel Ádám Zoltán (név) {Borbély Melinda (anyja neve) 646764CE (szem. ig. szám)} büntetőjogi és fegyelmi felelősségem tudatában kijelentem és aláírásommal igazolom, hogy a záródolgozat saját munkám eredménye. A felhasznált irodalmi és egyéb információs forrásokat az előírásoknak megfelelően kezeltem, a záródolgozat készítésre vonatkozó szabályokat betartottam.

Kijelentem, hogy ahol mások eredményeit, szavait vagy gondolatait idéztem, azt a záródolgozatomban minden esetben, beazonosítható módon feltüntettem, a dolgozatban található fotók és ábrák közlésével pedig mások szerzői jogait nem sértem.

Kijelentem, hogy a záródolgozatom elektronikus változata teljes egészében megegyezik a nyomtatott formával.

Hozzájárulok ahhoz, hogy az érvényben lévő jogszabályok és a Számalk-Szalézi Szakgimnázium belső szabályzata alapján az iskola saját könyvtárában megtekinthető (olvasható) legyen a záródolgozatom.

A záródolgozat titkos/nem titkos.

Budapest, 2020. március 30.

.....
Tanuló aláírása

Tartalomjegyzék

1.	Bevezetés	3
2.	Feladatspecifikáció	4
2.1	Járműkereskedés specifikálása.....	4
2.2	Személyautó szerviz specifikálása	2
3.	Tervezés	3
3.1	A program funkciói	3
3.2	Adatbázis szerkezet leírása	3
3.2.1	Tábla szerkezetek.....	4
3.2.2	Táblák, és kapcsolataik	12
3.3	A felhasználói felület	13
3.3.1	Bejelentkezés	13
3.3.2	Főmenü	14
3.3.3	Járművek keresése	15
	16
3.3.4	Járművek hozzáadása.....	16
3.3.5	Szerviz	18
3.4	Felhasználó hozzáadása	23
3.5	Szerviz módosítása.....	24
3.6	MVC.....	26
3.6.1	UML ábra.....	28
4.	Megvalósítás	29
4.1	Járművek hozzáadásának megvalósítása.....	29
4.2	Járművek megjelenítésének megvalósítása.....	30
4.3	Szerviz működésének megvalósítása	30
4.4	Felhasználók kezelése	31

4.5	Osztályok leírása	31
4.5.1	Main	31
4.5.2	Adatbázis	31
4.5.3	Modell	35
4.5.4	Vezérlő	39
4.5.5	Nézet	44
5.	Tesztelés.....	48
5.1	Rendszerkövetelmények	48
5.2	Tesztelés menete	48
5.3	Teszteléskor felfedezett Hibák.....	53
6.	Továbbfejlesztési lehetőségek	54
6.1	Meglévő funkciók	54
6.2	Új funkciók.....	54
7.	Felhasználói dokumentáció.....	55
7.1	Rendszer követelmények	55
7.2	Telepítés és futtatás	55
7.3	Program leírása.....	55
7.4	Menürendszer	55
7.4.1	Belépés/Kilépés	56
7.4.2	Főmenü	56
7.4.3	Szerviz	57
7.4.4	Szerviz módosítása	63
7.4.5	Jármű keresése	65
7.4.6	Jármű hozzáadása	66
7.4.7	Felhasználó hozzáadása	69
8.	Összefoglalás	70
9.	Irodalomjegyzék	71

Ábrajegyzék

1. ábra Adatbázis Táblák és kapcsolataik	12
2. ábra Bejelentkezés ablak GUI terv	14
3. ábra Főmenü ablak GUI terv	15
4. ábra Járművek keresése GUI tervezés	16
5. ábra Jármű hozzáadása első ablak GUI terv	17
6. ábra Jármű hozzáadása második ablak GUI terv	17
7. ábra Jármű hozzáadása/módosítása/törlése GUI terv	17
8. ábra Szerviz menü ablak GUI terv	18
9. ábra Szerviz menü ügyfél hozzáadása GUI terv	19
10. ábra Szerviz márka hozzáadása GUI terv	19
11. ábra Szerviz autó hozzáadása GUI terv	20
12. ábra Szerviz feladat hozzáadása GUI terv	21
13. ábra Szerviz munkalap hozzáadása GUI terv	21
14. ábra Szerviz alkatrész hozzáadása GUI terv	22
15. ábra Szerviz szerelés hozzáadása GUI terv	22
16. ábra Szerviz beszerzés hozzáadása GUI terv	23
17. ábra Felhasználó hozzáadása ablak GUI terv	24
18. ábra Szerviz módosítása ablak GUI terv	24
19. ábra Szerviz ügyfél módosítása GUI terv	25
20. ábra Szerviz beszerzés módosítása GUI terv	26
21. ábra MVC modell (forrás: MVC_mintak.ppt)	27
22. ábra Áttekintő osztálydiagram	28
42. ábra Belépés ablak képernyőkép	56
23. ábra Főmenü képernyőkép	57
24. ábra Ügyfél hozzáadása képernyőkép	58
25. ábra Márka hozzáadása képernyőkép	58
26. ábra Autó hozzáadása képernyőkép	59
27. ábra Munkalap hozzáadása képernyőkép	60
28. ábra Feladat hozzáadása képernyőkép	61
29. ábra Szerelés hozzáadása képernyőkép	61
30. ábra Alkatrész hozzáadása képernyőkép	62
31. ábra Beszerzés hozzáadása képernyőkép	63

32. ábra Ügyfél adatainak módosítása képernyőkép	63
33. ábra Beszerzés adatainak módosítása képernyőkép.....	64
34. ábra Járművek keresése képernyőkép.....	65
35. ábra Járművek találati táblázata képernyőkép	65
36. ábra Jármű hozzáadása első oldal képernyőkép	66
37. ábra Jármű hozzáadása második oldal képernyőkép	67
38. ábra Jármű módosítása első oldal képernyőkép.....	67
39. ábra Jármű módosítása harmadik ablak képernyőkép	68
40. ábra Jármű módosítása második oldal képernyőkép.....	68
41. ábra Felhasználó hozzáadása képernyőkép.....	69

1. Bevezetés

A szakdolgozatom témája egy járműkereskedés járműveinek számon tartása és kezelése, minden mellett egy gumiabroncs és az autókkal kapcsolatos kisebb szervizelések feladatának ellátása. Fontos volt számomra, hogy olyan témát válasszak, mellyel segíteni tudom más munkáját, megtanuljam a program által használt programozási nyelveket és elsajátításam a szoftverfejlesztést, mely izgalmas, mivel folyton új dolgokat tanulok.

Szakdolgozatom célja tehát egy olyan program megalkotása melyet megrendelésre készítettem, hogy megkönnyítsem a megrendelő munkáját és azáltal, hogy ezt a programot használva kevesebb idő alatt több ügyfelet tudjon kezelní és nagyobb bevételre tudjon szert tenni.

Azért ezt a platformot választottam, mivel számomra mindig is a számítógépes platform marad a kedvencem. Ezt ismerem a legjobban, és a későbbiekbén is ezzel szeretnék foglalkozni. A fejlesztési nyelv nem volt kérdés, mikor elkezdtem a tanulmányaimat az iskolában a JAVA nyelvvel ismerkedtem meg először, és nagyon megtetszett. Az adatkezelés egy másik kedvenc témám, véleményem szerint az adatok ma is és mindig is fontos alapjai lesznek az informatikának, ezért érdemes ezzel a témakörrel foglalkozni.

Az elkészítés során Microsoft Windows 10 Profession magyar nyelvű operációs rendszert, NetBeans IDE 8.2 fejlesztői környezetet, JAVA 1.8.0_161 verziószámú JDK-t, MySQL 5.0.12 adatbáziskezelő szoftvert, XAMPP 3.2.4 webszervert használtam. A program kiüzetét a Java beépített Swinges GUI widget eszközökkel szeretném elkészíteni, tehát Desktop alkalmazás, amit számítógépen lehet futtatni.

2. Feladatspecifikáció

Az általam írt program elkészítésével két célom volt, az egyik cél egy járműkereskedés járművek adatainak megtekintése, felvétele, törlése és módosítása. A programom másik része az autószereléshez kötődik, ha egy adott ügyfél behozza a járművét azon a gumiabroncsaival kapcsolatos szervizelések, és az autóhoz köthető szervizelések, javítások, alkatrész cserék, a jármű műszaki vizsgára való felkészítése, és vizsgáztatását lehet ellátni. Mivel a járműkereskedés és a szerviz egy telephelyen van, így azonos nyitvatartási idő érvényes mind a kettőre. minden hétköznap reggel nyolc-tol, este öt óráig van nyitva a telephely, ezenkívül szombaton ként reggel nyolc-tol délután kettő-ig. A programot úgy terveztem, hogy több felhasználó is használni tudja, de csak egy felhasználó fogja használni. Az alkalmazás asztali rendszerként kerüljön fejlesztésre.

A funkciókkal szembeni általános elvárások

Új adat felvitelle esetén biztosítani kell, hogy csak olyan adat kerüljön mentésre, ahol a kötelező adatok ki vannak töltve, és az adat megfelel az elvárt adat típusának.

Állapot változás esetén biztosítani, hogy a változtatott adat korábbi állapota megmaradjon. Az új állapot, pedig megfeleljen az adott adat típusának. Az újadat típusának meg kell felelnie az elvárt adat típusának.

Adat javítás estén, biztosítani, hogy csak a hatályos adaton lehessen javítani, új állapot létrehozása nélkül.

Adat törlése esetén, biztosítani kell, hogy csak az adott adatott lehessen kitörlni.

2.1 Járműkereskedés specifikálása

A járműkereskedéshez tartozó részfunkciók a következőek, a telephelyen lévő járműveket az alkalmazással meg lehet tekinteni, módosítani, hozzáadni, és javítani a megadott jármű adatait, ezenkívül a járművekhez tartozó képeket lehet megtekinteni és hozzáadni, vagy eltávolítani. A járműkereskedés rendelkezik egy központi telephellyel, ahol a járműveket tároljuk, ha egy adott ügyfél vásárolni szeretne, akkor megtekintheti őket. Az ügyfelek csak a nyitvatartási időn belül tekinthetik meg a járműveket. Mivel három fajta járművet szeretnék tárolni a telephelyen, ezek el vannak egymástól szeparálva az átláthatóság miatt.

- személyautó

- motorkerékpár/quad
- haszonjármű

Mind a három fajta járműnek vannak tulajdonságaik melyek hasonlítanak egymásra, bizonysos szempontból. A közös tulajdonságok a következők:

- | | | |
|------------|----------------------|-----------------|
| • rendszám | • üzemanyag | • sebességváltó |
| • márka | • hengerűrtartalom | • fajtája |
| • modell | • kilométeróra állás | • jármű leírása |
| • típus | • teljesítmény | • jármű hibái |
| • évjárat | | • vételár |
| • hónap | | • engedmény |
| • állapot | | |

Minden egyes járművet a saját rendszáma határozz meg. A rendszám hét karakterből áll: három nagybetűből, melyek az angol ábécé betűiből állnak. A betűk után egy kötőjel van, amit három darab szám követ. A járművek maximális súlya 9999 kilogramm lehet, e fölötti súlyú járművel a kereskedés nem foglalkozik. Egyes járműtípusoknál vannak egyedi tulajdonságok, vagy esetleg hiányzik valamelyik tulajdonság amelyik a többi járműnél létezik. Mind a három járműtípusnál adott a jármű súlya. A járművek kivitele több típusú is lehet, ezek járművekként változóak. Személyautóknál a következők lehetnek: cabrio, coupe, egyterű, ferdehátú, kisbusz, kombi, lépcsőhátú, pickup, sedan, sport, terepjáró, városi terepjáró, egyéb. A motorkerékpároknál a következők lehetnek: chopper, classic(veterán), cross, enduro, gyerekmotor, gyorsasági(sport), oldalkocsis, quad, robogó, segédmotoros kerékpár, supermoto, trial, túra, versenymotor, egyéb. A haszonjárműveknél a következők lehetnek: billencs, darugémes, darus, dobozos, duplakabinos, harci jármű, hűtő, járműszállító, konténerszállító, mentő, nyerges szerelvény, platós, ponyvás, tartályos, tűzoltó, egyéb. A járművek állapotai már azonosak, ezek a következők: normál, kitűnő, megkímélt, újszerű, sérülésmentes, sérült, enyhén sérült, eleje sérült, hátulja sérült, baloldala sérült, jobboldala sérült, hiányos, hibás. Több fajta üzemanyaggal működnek a járművek. A motorkerékpár csak háromfajta üzemanyaggal kapható, ezek a következők: benzin, dízel, elektromos. A személyautónál, és a haszonjárműnél több fajta lehetőség adott: benzin, dízel, benzin/gáz, hibrid, hibrid(benzin), hibrid(dízel), elektromos, etanol, biodízel, gáz. A teljesítmény kilowattban van értelmezve, de egy beépített metódussal ezt át szeretném váltani lóerőre és mellé íratni. A jármű színeire pontos jogszabályt nem találtam csak a lényeget, ami 01-es számkóddal indul és 10-ig megy, tehát a járművek színei

összesen tíz féle lehet: fehér (01), sárga (02), narancs (03), piros (04), bíbor/lila (05), kék (06), zöld (07), szürke (08), barna (09), fekete (10). A sebességváltó értéke lehet: 0(mechanikus), 1(félautomata), 2(automata), vagy 3(szekvenciális). Mindegyik járműhöz tartozik egy leírás és egy rövid vélemény az adott jármű aktuális és típus hibáiról. A tulajdonos által megvásárolt járműveket, melyek még nincsenek a telephelyen, természetesen kezelni szeretném a programomban. Ha megérkezett az adott jármű a telephelyre, akkor átkerül a telephelyre a többi jármű közé.

2.2 Személyautó szerviz specifikálása

A szervizhez tartozó részfunkciók a következők, az ügyfél adatainak felvitele, az autó adatainak felvitele, munkalap létrehozása, az adott feladatokkal, és a szereléshez szükséges alkatrészekkel. Adott ügyfél adott személyautót a nyitvatartási időn belül hozhatja és viheti el a szervizből. Mikor az ügyfél behozza az autóját, meg kell adja az okmányazonosító számát, nevét, és elérhetőségét. Természetesen az autó információi a legfontosabbak a szervizelés szempontjából, ezért az autóról a következő információkat kell eltárolni. Az autó rendszáma határozza meg az autót, de ezenkívül a márkáját, évjáratát, típusát, színét, motorszámát, és alvázszámát kell megadni. minden egyes autó rendelkezik egy munkalappal, amin a szereléshez köthető információk találhatóak. A munkalapon a következő információk találhatók, a munkalap sorszáma, az adott autó rendszáma, a szerelekhez tartozó kolléga neve, a hibajegyzék, ami meghatározza mit kell csinálni az adott járművel, a munkadíj, anyagdíj, vállalási határidő. Egy autóhoz egy szerelő van hozzárendelve, az átláthatóság érdekében. A munkalap ezenkívül tartalmazza a feladatokat, amin az aktuális és már befejezett tennivalók találhatóak. Mivel a legtöbb szereléshez új alkatrész szükséges, ezért fel van tüntetve a munkalapon. Az alkatrészeket, amik szükségesek az aktuális szereléshez a munkalap sorsáma határozza meg. Tehát a munkalap tartalmazza az elvégzett feladatokat a munkadíjjakkal, és az elnevezésekkel, és az éppen szükséges alkatrészek kódját, megnevezését, és árát. Ezenkívül a megadott adatokból PDF-et szeretnék generálni munkalap néven, hogy igény szerint az ügyfél kapjon egy kézzelfogható dokumentumot.

3. Tervezés

3.1 A program funkciói

- felhasználó bejelentkeztetése, regisztrációja
- személyautó keresése, felvitele, törlése, módosítása
- motorkerékpár keresése, felvitele, törlése, módosítása
- haszonjármű keresése, felvitele, törlése, módosítása
- keresett jármű adatainak és képeinek megjelenítése
- ügyfél felvitele, módosítása
- munkalap létrehozása
- feladat felvitele
- szerelés felvitele
- alkatrész és beszerzés felvitele
- PDF generálása a megadott szerviz adatok alapján

3.2 Adatbázis szerkezet leírása

Az jármű kereskedés autóit külön táblákba tárolom az egyszerűbb lekérdezés érdekében, mivel mikor keresünk egy járművet egyszere csak egy fajta járműre keresünk rá. Ezenkívül különböző márkájú autók szervizmunkáit tartom nyilván. Az autókat az ügyfelek hozzák be, akiknek nyilvántartjuk a nevét, elérhetőségeit. Amikor behoznak egy autót, nyitunk neki egy új munkalapot egyedi munkalapszámmal, amin feltüntetjük, hogy mik a problémák az autóval. A választható feladat lehet valamely autórész javítása vagy cseréje, de akár vizsgára való felkészítés vagy műszaki vizsgáztatás is. minden feladatnak óradíja és/vagy anyagköltsége van. Amennyiben alkatrészcsérét igényel a feladat, akkor a kívánt alkatrészt meg kell rendelni az (egyik) alkatrészbolttól. A munkalap feladatainak elvégzését az egyes szerelők a ráfordított munkaórák bejegyzésével zájják. Ebben a fázisban már jelezhetjük az ügyfélnek, hogy kész az autó, akinek elvitelkor ki kell fizetnie az összeszeget a szervizelésért. Az ügyfél a személyi okmányszáma és típusa határozza meg. A szervizben lévő autók márkáit a márka nevű táblában tárolom. A bejelentkezéshez három táblát terveztem: a felhasználó, az engedélyek, és az felhasználóhoz tartozó engedély táblát.

3.2.1 Tábla szerkezetek

KEDVEZMENYSAV: bizonyos összegekhez tartozó kedvezmények törzse

KEDVEZMENYSAV: {kedvezmeny_azonosito, osszeg}

UGYFEL: az ügyfelek törzse

UGYFEL: {szemelyazon_okmany_szam, szemelyazon_okmany_tipus, nev, telszam, lakcim}

MARKA: tábla a márkkák nyilvántartásához

MARKA: {mkod, marka}

DOLGOZO: a dolgozók törzse

DOLGOZO: {dkod, nev, munkakor}

AUTO: az autók törzse

AUTO: {rendszam, ugyfel_azon, marka, evjarat, tipus, szin, motorszam, alvazszam}

MUNKALAP: a behozott autók adatait tárolja, hogy milyen hibákat kell javítani az autón, a munkadíjat, az anyagdíjat, és a vállalási határidőt

MUNKALAP: {munkalap_szam, rendszam, hibajegyzek, munkadij, anyagdij, val-lalasi_hatarido}

SZERELES: az autóhoz tartozó feladatokat tárolja a hozzárendelt szerelővel, a szerelés kezdésével és befejezésével, és a szereléssel eltöltött időt.

SZERELES: {munkalap_szam, szerelo, feladat, kezdes, befejezes, munkaora}

FELADAT: a konkrét feladat fajtákat tárolja a hozzá tartozó általános munkadíjjal, és el-nevezéssel, illetve a feladat típusait

FELADAT: {feladat, munkadij, elnevezes, tipus ('j', 'c' , 'f' , 'v')}

BESZERZES: a feladatokhoz tartozó anyagbeszerzés törzse

BESZERZES: {besz_kod, munkalap_szam, alkatesz, aron, megrendelve}

ALKATRESZ: alkatrészek törzse

ALKATRESZ: {kod, megnevezes}

KERESKEDES_AUTO: személyautók törzse

KERESKEDES_AUTO: {rendszam, marka, modell, tipus, evjarat, honap, kivitel, allapot, uzemanyag, hengerur_tart, kilometerora_all, sajat_tomeg, teljesitmeny, szin, seb_valto, jarmu_leirasa, jarmu_hibai, telephely, vetelar, engedmeny}

KERESKEDES_MOTOR: motorkerékpárok törzse

KERESKEDES_MOTOR: {rendszam, marka, modell, tipus, evjarat, honap, kivitel, allapot, uzemanyag, hengerur_tart, kilometerora_all, sajat_tomeg, teljesitmeny, szin, seb_valto, jarmu_leirasa, jarmu_hibai, telephely, vetelar, engedmeny}

KERESKEDES_KAMION: haszonjárművek törzse

KERESKEDES_KAMION: {rendszam, marka, modell, tipus, evjarat, honap, kivitel, allapot, uzemanyag, hengerur_tart, kilometerora_all, sajat_tomeg, teljesitmeny, szin, seb_valto, jarmu_leirasa, jarmu_hibai, telephely, vetelar, engedmeny}

FELHASZNALO: felhasználók törzse

FELHASZNALO: {felhasznalo_kod, vezetek_nev, kereszt_nev, felhasznalo_nev, jelszo}

FELH_ENGED: adott felhasználóhoz tartozó engedély szintje

FELH_ENGED: {felhasznalo, engedely}

ENGEDELYEK: engedély kódja és elnevezése

ENGEDELYEK: {engedely_kod, elnevezes}

KEDVEZMENYSAV

mező	adattípus	szerep	korlátozás	példa
kedvez-meny_azono-sito	int(3)	PK	AUTO_INC-REMENT	1
osszeg	mediumint(9)	not null		10000

UGYFEL

mező	adattípus	szerep	korlátozás	példa
szemelyazon_okmany_szam	varchar(30)	PK		653594DE
szemelyazon_okmany_tipus	varchar(30)	not null		személyazonosító iga-zolvány
nev	varchar(50)	not null		Kiss Pista
telszam	varchar(15)	not null		0613537895
lakcim	varchar(50)	null		1145 Budapest Lánc-híd utca 45.

MARKA

mező	adattípus	szerep	korlátozás	példa
mkod	char(6)	PK	ma/000-999	ma/001
marka	varchar(30)			Audi

DOLGOZO

mező	adattípus	szerep	korlátozás	példa
dkod	int(11)	PK	dkod>=0	50
nev	varchar(20)	not null		Nagy Béla
munkakor	varchar(15)	not null		autószerelő/de

AUTO

mező	adattípus	szerep	korlátozás	példa
rendszam	char(7)	PK		FCV-654 NA-1234
ugyfel_azon	varchar(30)	FK		653594DE
marka	char(6)	FK		ma/001
evjarat	char(4)		1. hely 1-2 2-4. hely 0-9	1999
tipus	varchar(20)			s3 coupe
szin	char(2)			01
motorszam	char(20)			G67GJ67787
alvazszam	char(30)			ZJHZGG6766567656765

MUNKALAP

mező	adattípus	szerep	korlátozás	példa
munkalap_szam	char(12)	PK	mu/év/ 0000-9999	mu/2020/0001
rendszam	char(7)	FK	1 autó 1 időben 1	FGN-421

			munkalapon szerepelhet	
hibajegyzek	varchar(50)	not null		törött tükör, kopott féktárcsa és fékbetét
munkadij	mediumint(9)	default 0	>0	40000
anyagdij	mediumint(9)	default 0	>0	15000
vallalasi_hatarido	datetime			2020-01-12 14:00:00

SZERELES

mező	adattípus	szerep	korlátozás	példa
munka-lap_szam	char(12)	PK része		mu/2020/0001
szerelo	int(11)	FK		50
feladat	char(6)	FK		fe/001
kezdes	date	not null	feladat felvitelekor automatikusan töltődik ki	2020-01-13
befejezes	date	null	befejezés > kezdés	2020-01-28
tenyleges_munkaora	tinyint(2)	default 0	01-99	16

FELADAT

mező	adattípus	szerep	korlátozás	példa
feladat	char(6)	PK	fe/000-999	fe/001
munkadij	mediumint(9)	not null	munkadij>0	30000
elnevezes	varchar(30)	not null		fékcsere
tipus	char(1)	not null	'j', 'c', 'f', 'v'	j

BESZERZES

mező	adattípus	szerep	korlátozás	példa
besz_kod	char(12)	PK	be/év/0000-9999	be/2020/0001
munkalap_szam	char(12)	FK része		mu/2020/0001
alkatresz	char(6)	FK része		al/001
aron	mediumint(9)	not null	aron>0	10000
megrendelve	date	not null	megrendelve<= mai dátum	2020-01-17

ALKATRESZ

mező	adattípus	szerep	korlátozás	példa
kod	char(6)	PK	al/000-999	al/001

megnevezes	varchar(30)	not null		féktárcsa
-------------------	-------------	----------	--	-----------

KERESKEDES_AUTO

mező	adattípus	szerep	korlátozás	példa
rendszam	char(7)	PK		FCV-654 NA-1234
marka	varchar(20)	not null		Audi
modell	varchar(20)	not null		A8
tipus	varchar(10)			B7
evjarat	char(4)	not null	1. hely 1-2 2-4. hely 0-9	2003
honap	char(2)	not null	1. hely 0-1 2. hely 0-9	02
kivitel	varchar(30)	not null		sedan
allapot	varchar(20)	not null		normál
uzemanyag	varchar(15)	not null		benzin
hengerur_tart	varchar(5)	not null		2967
kilometerora_all	varchar(7)	not null		210260
sajat_tomeg	varchar(4)	not null		1915
teljesitmeny	varchar(4)	not null		184
szin	char(2)	not null	01-10	03
seb_valto	char(1)	not null	0-3	1
jarmu_leirasa	varchar(1000)			valódi ritkáság, egyedi darab, kitűnő műszaki állapot
jarmu_hibai	varchar(1000)			törött tükrök, sérült szélvédő
telephely	tinyint(1)	not null		1
vetelar	int(8)	not null	vetelar>0	3000000

KERESKEDES_MOTOR

mező	adattípus	szerep	korlátozás	példa
rendszam	varchar(7)	PK		IMV-751 NM-0001
marka	varchar(20)	not null		BMW
modell	varchar(20)	not null		R 1200 RT
tipus	varchar(10)			LC
evjarat	char(4)	not null	1. hely 1-2 2-4. hely 0-9	2006
honap	char(2)	not null	1. hely 0-1 2. hely 0-9	06
kivitel	varchar(30)	not null		Túra
allapot	varchar(20)	not null		kitűnő
uzemanyag	varchar(15)	not null		benzin
hengerur_tart	varchar(5)	not null		1170
kilometerora_all	varchar(7)	not null		16000
sajat_tomeg	varchar(4)	not null		360
teljesitmeny	varchar(4)	not null		92
szin	char(2)	not null	1. hely 0-1 2. hely 0-9	01
seb_valto	char(1)	not null	0-3	3
jarmu_leirasa	varchar(1000)			Full extrás kivitel, kevés kilométerrel, kifogástalan műszaki és esztétikai állapotban. Hazai BMW márkkakereskedésben vásárolt.
jármu_hibai	varchar(1000)			jobb oldali index nem működik, első fényszóró nem működik
telephely	tinyint(1)	not null		1
vetelar	int(8)	not null	vetelar>0	4699000

KERESKEDES_KAMION

mező	adattípus	szerep	korlátozás	példa
rendszam	char(7)	PK		GBH-543 NK-0001
marka	varchar(20)	not null		DAF
modell	varchar(20)	not null		45
tipus	varchar(10)	null		Turbo 150
evjarat	char(4)	not null	1. hely 1-2 2-4. hely 0-9	1996
honap	char(2)	not null	1. hely 0-1 2. hely 0-9	06
kivitel	varchar(30)	not null		konténerszállító
allapot	varchar(20)	not null		Normál
uzemanyag	varchar(15)	not null		Dízel
hengerur_tart	varchar(5)	not null		12580
kilometerora_all	varchar(7)	not null		450000
sajat_tomeg	varchar(4)	not null		7340
teljesitmeny	varchar(4)	not null		315
szín	char(2)	not null	1. hely 0-1 2. hely 0-9	01
seb_valto	char(1)	not null	0-3	0
jarmu_leirasa	varchar(1000)	null		Műszaki lejárt, motor váltó hibás, két vandonatúj akkumulátor, lábon vihető, csörlő, vonóhorog!
jarmu_hibai	varchar(1000)	null		gumik előregedtek cserések
telephely	tinyint(1)	not null		1
vetelar	int(8)	not null	vetelar>0	999000

FELHASZNALO

mező	adattípus	szerep	korlátozás	példa
felhasznalo_kod	int(11)	PK	felhasznalo_kod>0	1
vezetek_nev	varchar(30)	not null		Kiss
kereszt_nev	varchar(30)	not null		Sándor
felhasznalo_nev	varchar(20)	not null		kiss.sandor
jelszo	varchar(128)	not null		jelszó

jel-

szó=3ece34947b362b2a4dee099c9b5a77be85ed912f41e3448d1f40e38a9bbae51b85351
e06b657f83f8515502ffbe4e0318ac2df2a4a61d2dec737588d780089dc

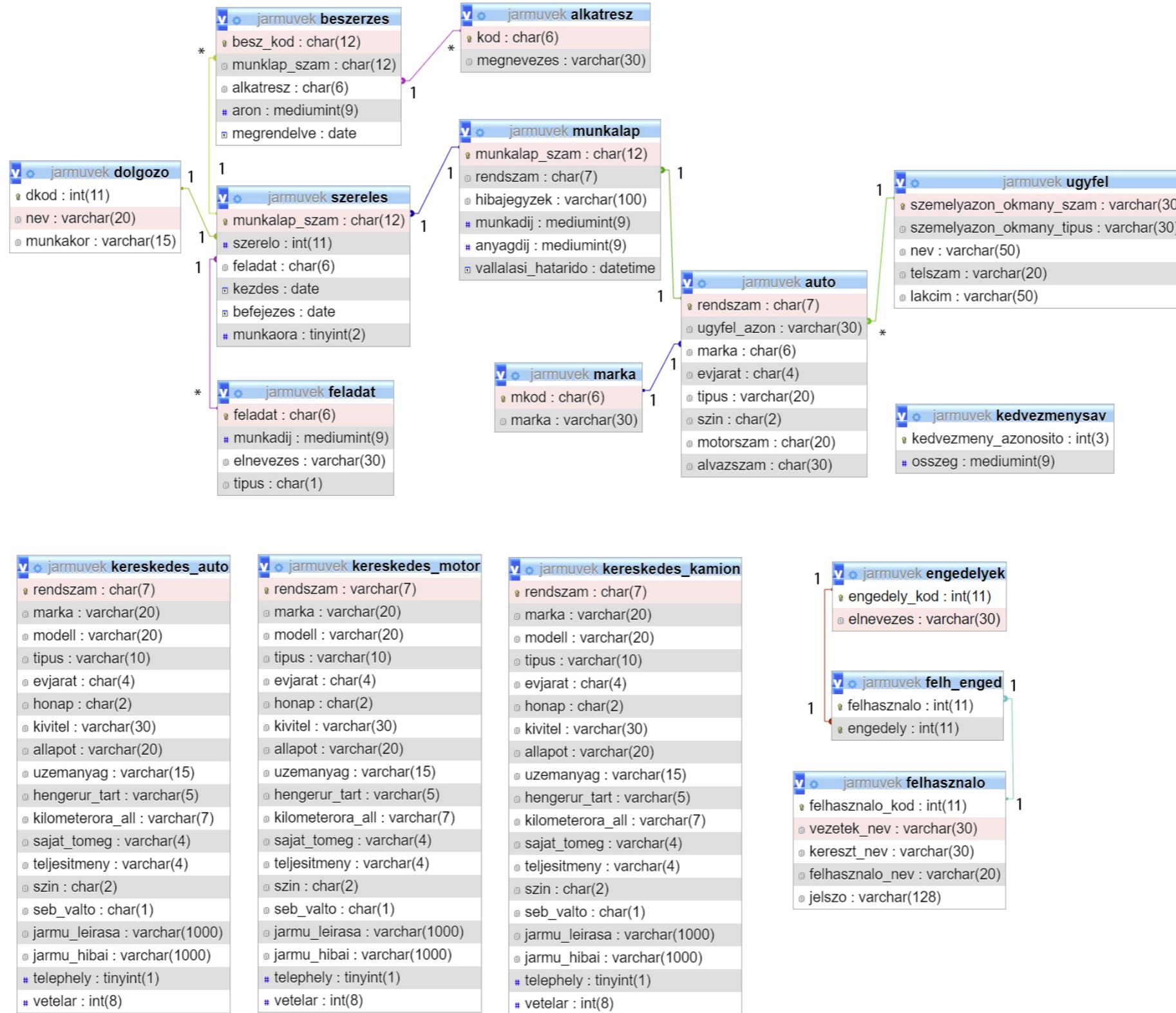
FELH_ENGED

mező	adattípus	szerep	korlátozás	példa
felhasznalo	int(11)	PK	felhasznalo>0	1
engedely	int(11)	PK	engedelyek>0	1

ENGEDELYEK

mező	adattípus	szerep	korlátozás	példa
engedely_kod	int(11)	PK	enge-dely_kod>0	2
elnevezes	varchar(30)	not null		Menedzser

3.2.2 Táblák, és kapcsolataik



1. ábra Adatbázis Táblák és kapcsolataik

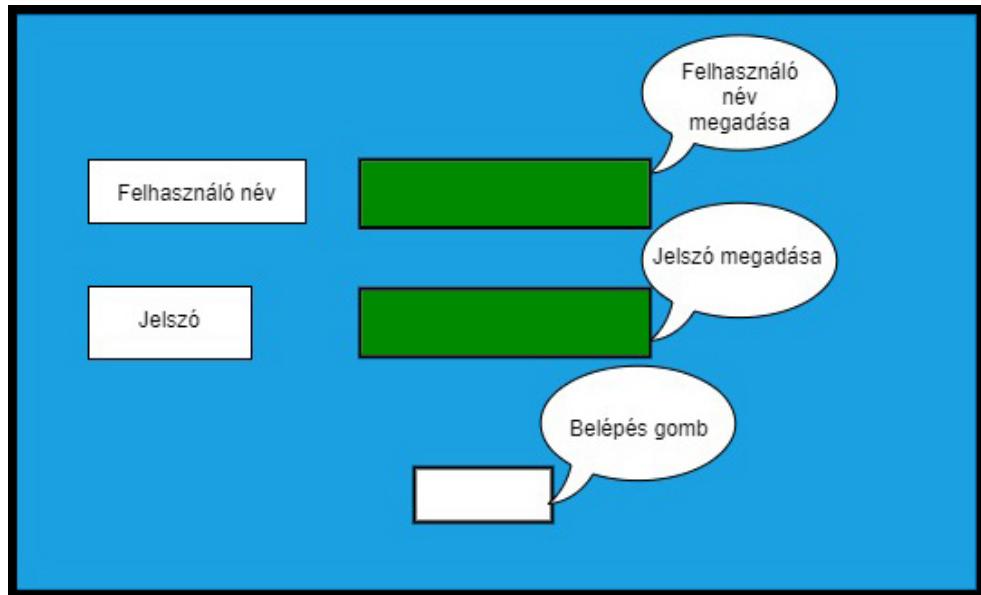
3.3 A felhasználói felület

Törekedtem az egyszerűségre, hogy a program kezelését ne kelljen megtanulni és az átlag felhasználók is könnyen használhassák a programot. Továbbá a kezelőfelület egyértelmű, a feliratok nem félrevezetőek, így könnyen értelmezhető, hogy mi mit működtet, minek a hatására mi történik. Próbáltam egy logikus, átlátható felületet megtervezni, ahol az egyes vezérlőelemek jól elkülönülnek egymástól. A program minden csak annyi információt szolgáltat, ami feltétlenül szükséges, így nem válik bonyolulttá a működés. Az átláthatóság és a logikus felhasználói felület megalkotása érdekében, ahol szükséges volt a felület nagysága miatt a JTabbedPane komponenst szeretném használni, mivel az ügyfél az 1024*768 pixeles felbontást igényelte, ezért minden felületen egységesen alkalmaztam.

A program a bejelentkező felülettel indul, ahol a felhasználó be tud jelentkezni, miután belépett a felhasználó, átkerül a főmenübe, ahol ki tudja választani, mit szeretne csinálni. A főmenüben négy gomb található, ahonnan a következő menüpontokra lehet gombnyomásra átugrani. A járműkereskedés járműveit a járművek panelen lehet kiválasztani. A járművek hozzáadása panelen járműveket lehet hozzáadni az adatbázishoz. A szerviz panelen pedig az ügyfél által behozott járművek szervizelésének menetét lehet elindítani.

3.3.1 Bejelentkezés

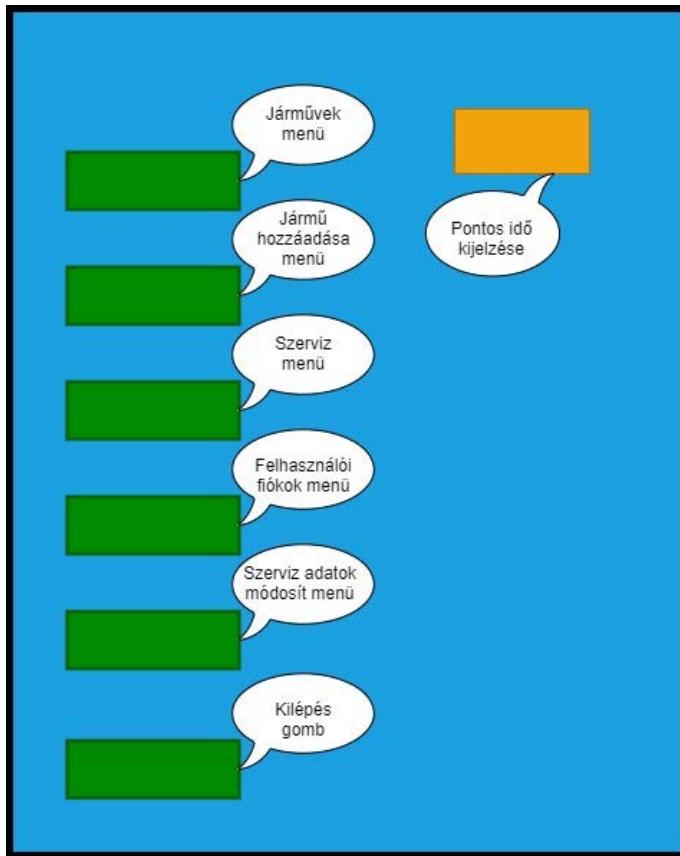
Az alkalmazás ezzel a felülettel indul, ahol a felhasználó betud jelentkezni. Véleményem szerint ennek a felületnek az egyszerűség és letisztultság a lényege, ezért ennek a gondolatmenetnek megfelelően terveztem meg. A felhasználónak meg kell adnia a felhasználónévét, és a jelszavát a bejelentkezéshez. Miután sikeresen bejelentkezett, átkerül a főmenübe.



2. ábra Bejelentkezés ablak GUI terv

3.3.2 Főmenü

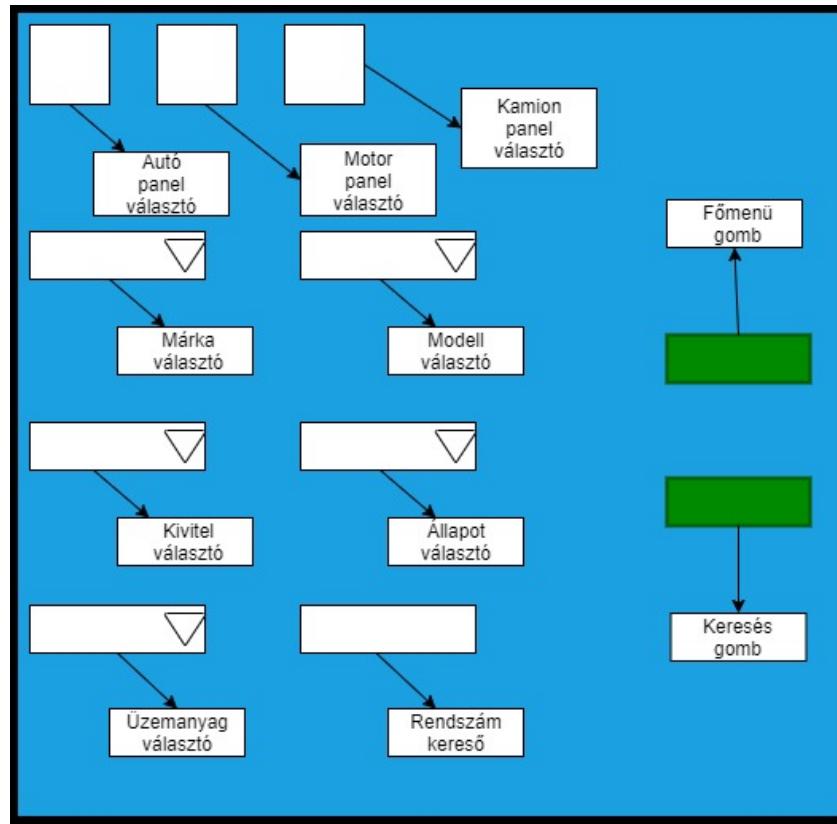
A főmenü arra szolgál, hogy az alkalmazás funkcióit egy helyről el tudjuk érni. Ebből a menüpontból tudunk elnavigálni a többi menübe, ezenkívül kijelzi a pontos időt.



3. ábra Főmenü ablak GUI terv

3.3.3 Járművek keresése

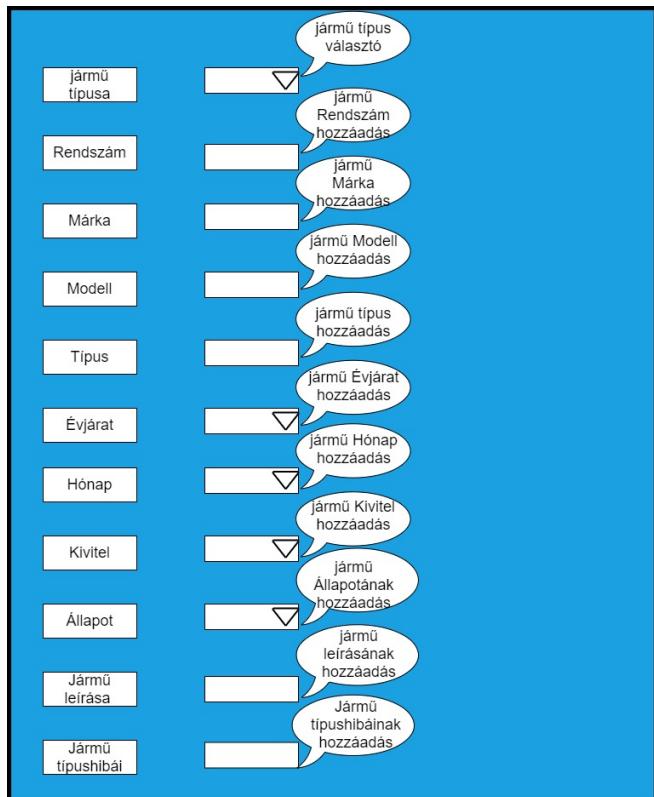
Ebben a menüpontban járművek szerint lehet keresni. A felületet a JTabbedPane komponens használatával három részre bontottam. Mind a három járműtípusnál azonosak a keresési feltételek. A keresési feltételek a következők, márka, modell, típus, kivitel, állapot, üzemanyag, rendszám, adott évjárattól, adott évjáratig, adott hengerűrtartalomtól, adott hengerűrtartalomig, adott kilométerorra állástól, adott kilométerorra állásig, adott vételártól, adott vételárig. Miután megadtuk a kívánt keresési paramétereket, egy új panelen megjelenek a járművek. Jelen pillanatban a keresés nem működik ezért, ha a felhasználó a keresés gombra kattint, a keresés feltételétől függetlenül mindegyik jármű adata megjelenik.



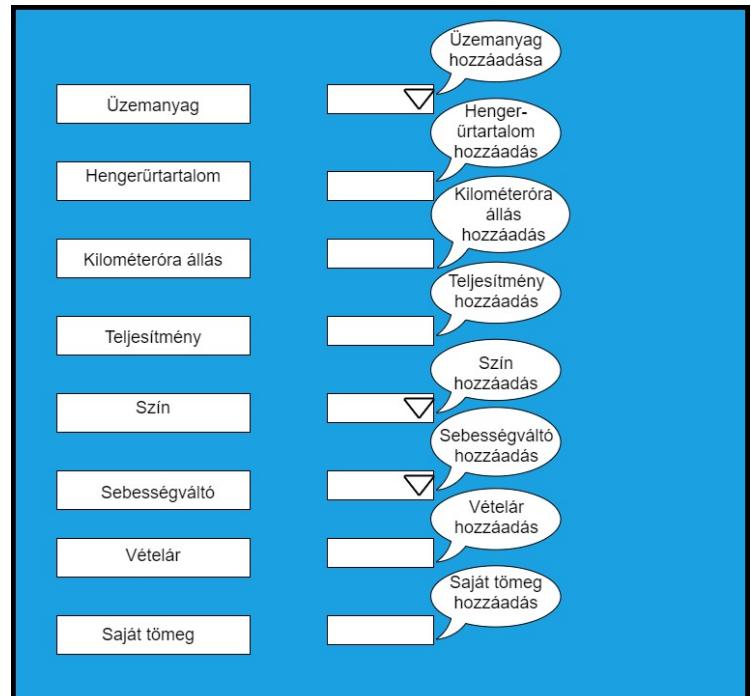
4. ábra Járművek keresése GUI tervezési diagram

3.3.4 Járművek hozzáadása

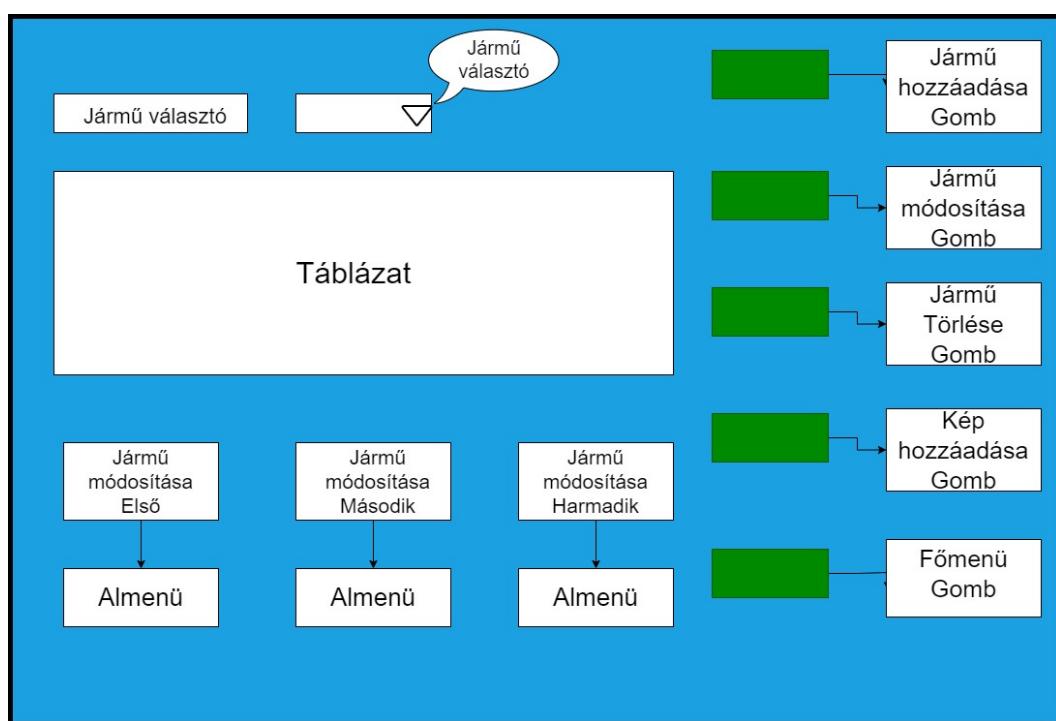
A járművek hozzáadása menüpontban, hozzá tudjuk adni a kiválasztott jármű típus alapján az adott járművet. A menüpontot három részre osztottam a JTabbedPane komponens használatával. Ezenkívül ebben a menüpontban lehet a jármű adatait megváltoztatni, vagy adott járművet törleni. A jármű módosítása ablaka három részből áll, ezek az ablakok hasonlítanak a jármű hozzáadás ablakhoz. Az első részben A jármű rendszámát kell megadni, mivel ez határozza meg az autót. Miután a rendszám kitöltésre került a következő adatokat lehet megváltoztatni: jármű márkkája, modellje, típusa, gyártási évjára és hónapja, kivitele, és állapota. A második részben meg lehet változtatni, hogy az adott jármű milyen típusú üzemanyagot használ a működéséhez, ezenkívül a jármű hengerűrartalmát, kilométeréra állását, saját tömegét és teljesítményét. A harmadik ablakban a jármű színét, sebesség váltó fajtáját, leírását, hibáit/típushibáit, a jármű vételárát és a státuszát, hogy éppen a telephelyen van-e az adott jármű.



5. ábra Jármű hozzáadása első ablak GUI terv



6. ábra Jármű hozzáadása második ablak GUI terv

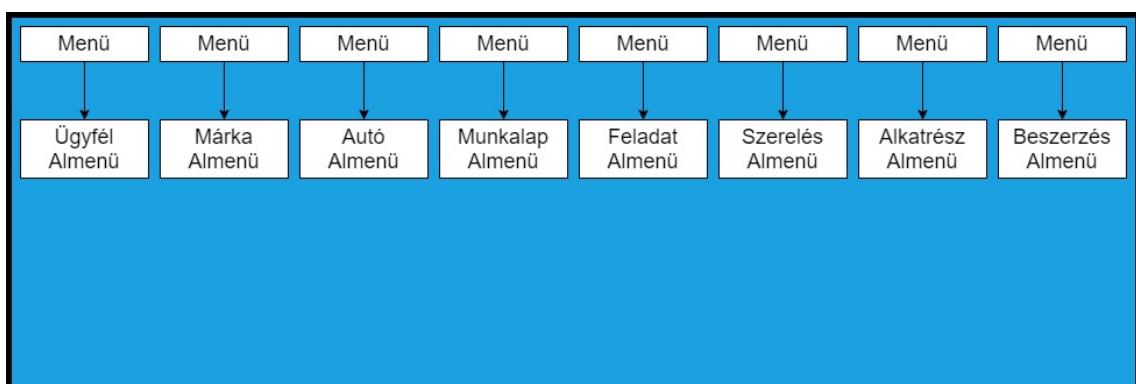


7. ábra Jármű hozzáadása/módosítása/törlése GUI terv

A 7.-ábra a jármű módosításának az átláthatósági ábrája.

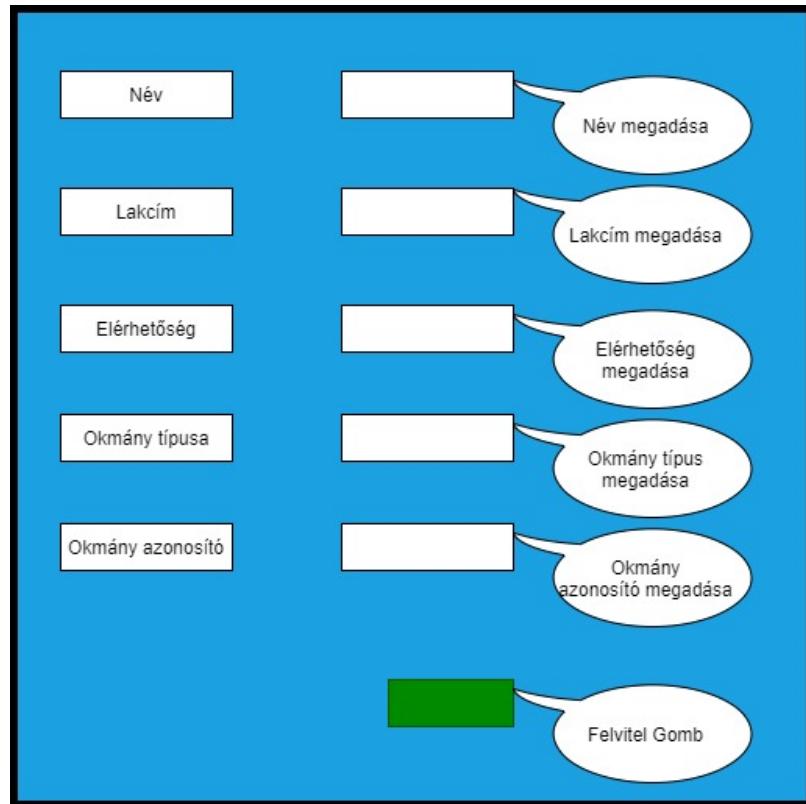
3.3.5 Szerviz

A legösszetettebb menü az alkalmazásban. Új munkalap létrehozására szolgál. Új munkalap létrehozásakor a következő dolgokat kell megadni: ügyfél adatai, márka kódja és elnevezése (ha esetleg nem tartalmazná az adott márkat az adatbázis), autó adatai, munkalap, feladat, szerelés (adott munkalap szerelési információ), alkatrész és beszerzés. Mi-

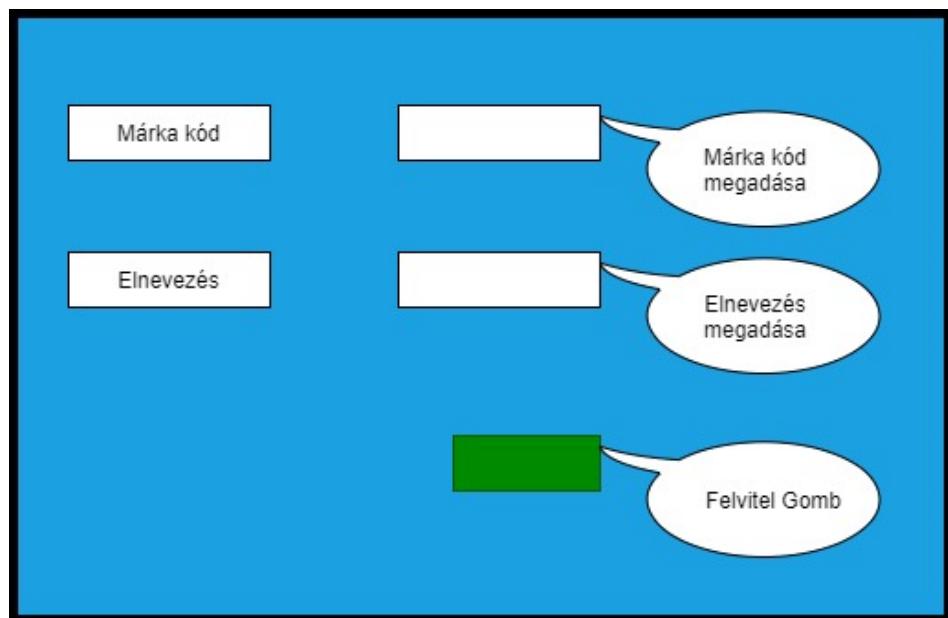


8. ábra Szerviz menü ablak GUI terv

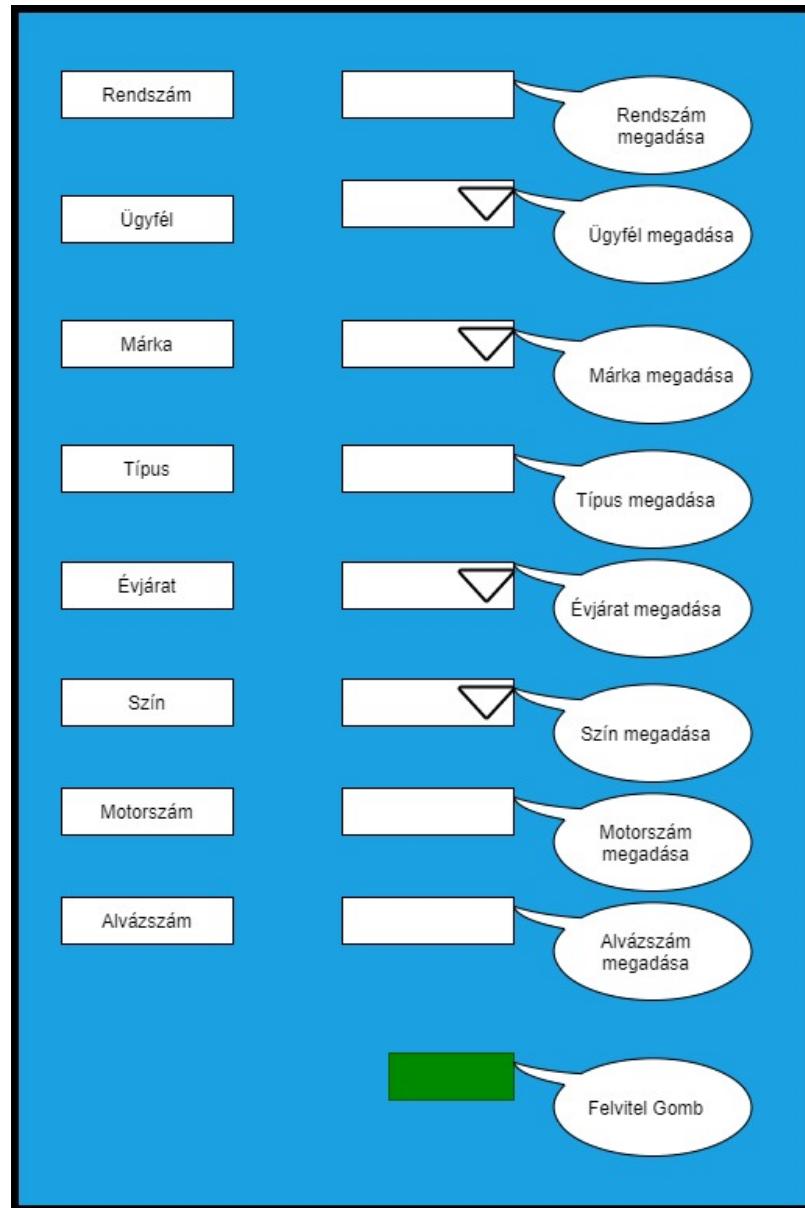
után a felhasználó kitöltötte a fent említett menüpontokat, a beszerzés fülön található munkalap készít gomb megnyomásával munkalapot generál a megadott adatok szerint.



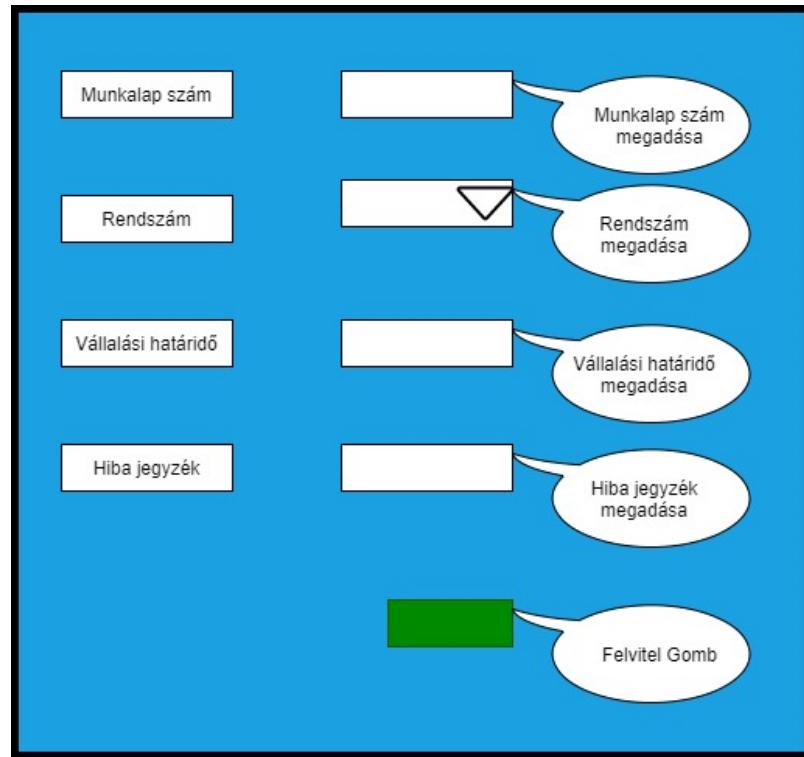
9. ábra Szerviz menü ügyfél hozzáadása GUI terv



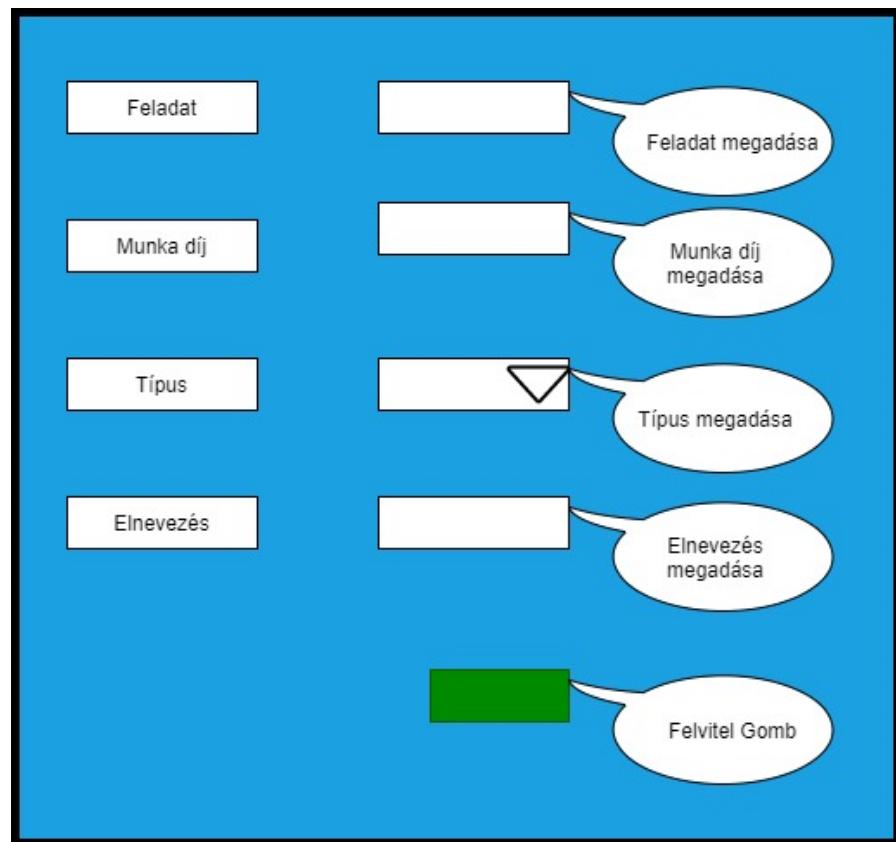
10. ábra Szerviz márka hozzáadása GUI terv



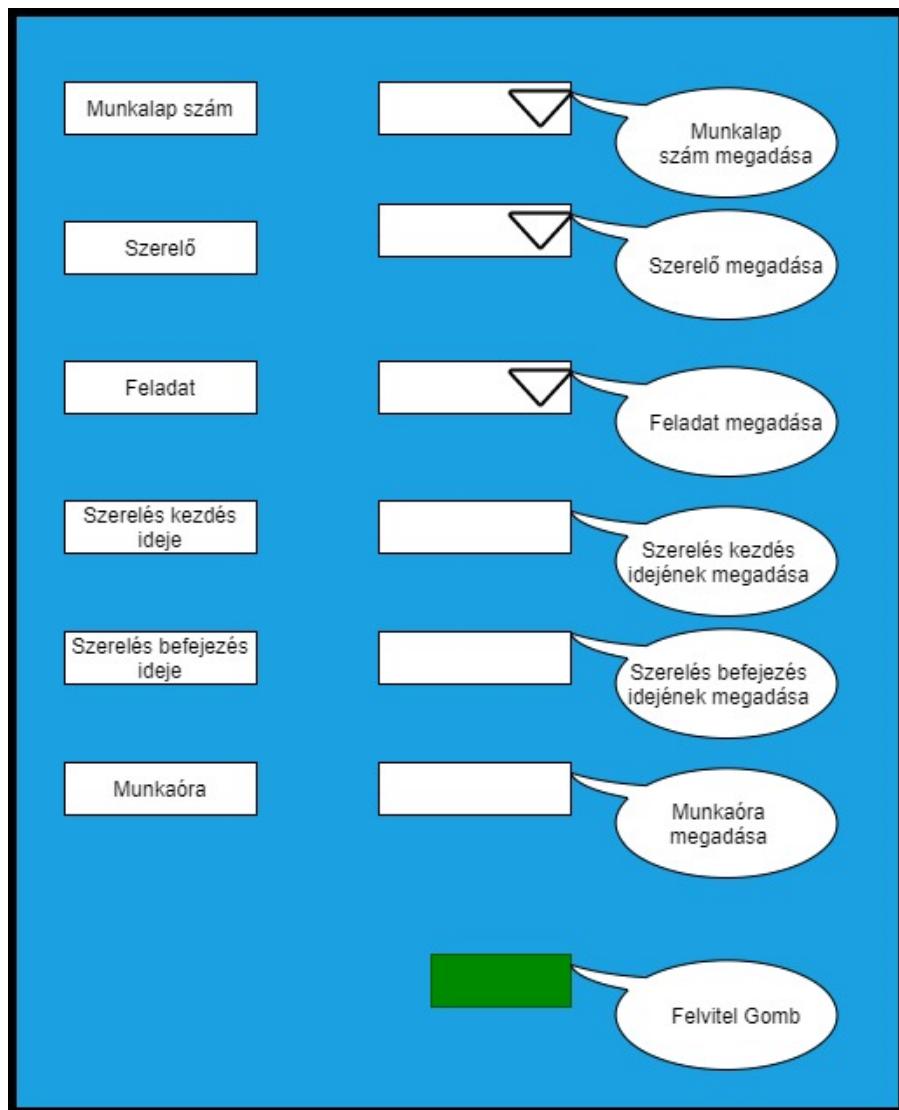
11. ábra Szerviz autó hozzáadása GUI terv



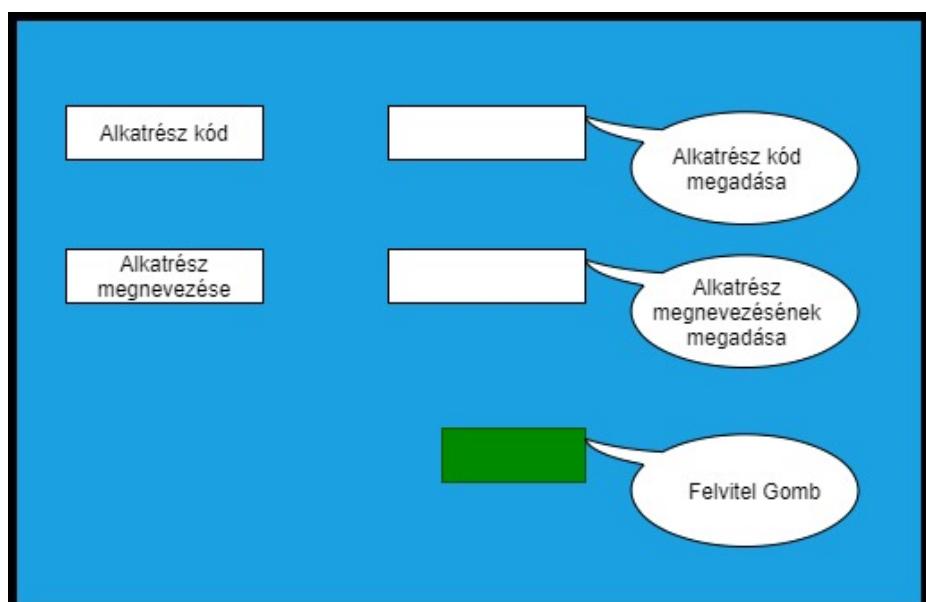
13. ábra Szerviz munkalap hozzáadása GUI terv



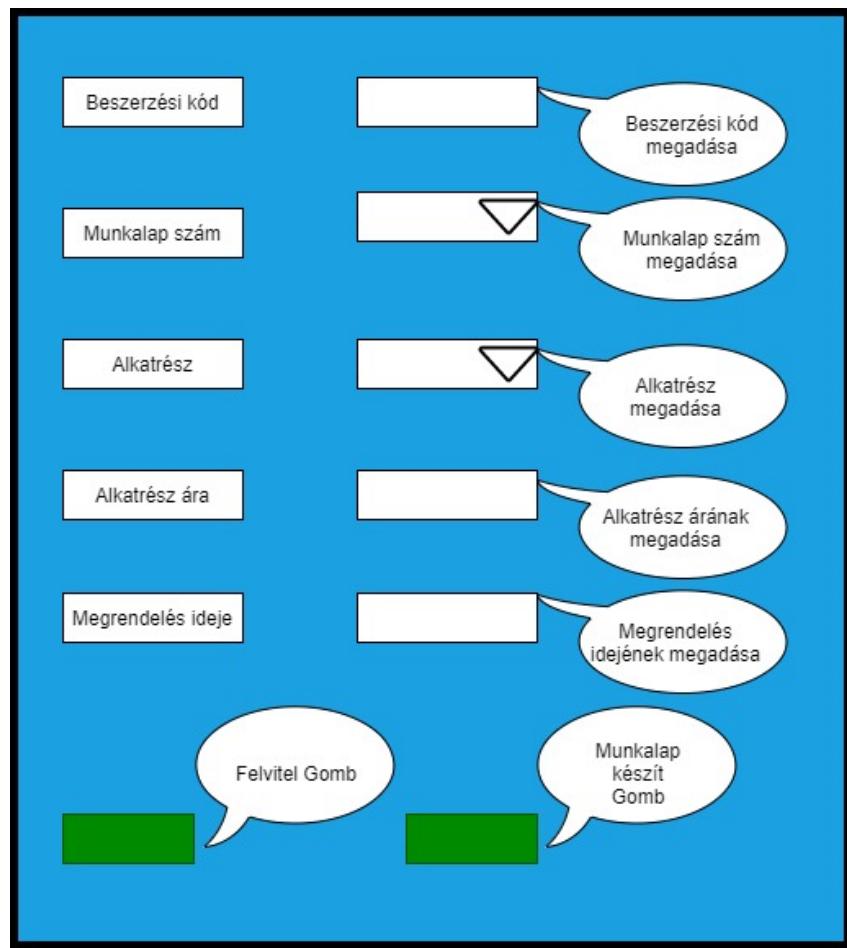
12. ábra Szerviz feladat hozzáadása GUI terv



15. ábra Szerviz szerelés hozzáadása GUI terv



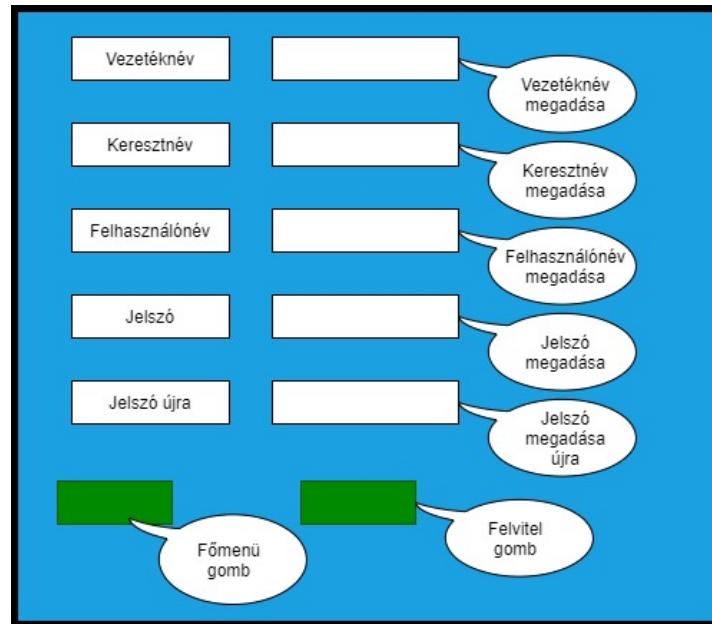
14. ábra Szerviz alkatrész hozzáadása GUI terv



16. ábra Szerviz beszerzés hozzáadása GUI terv

3.4 Felhasználó hozzáadása

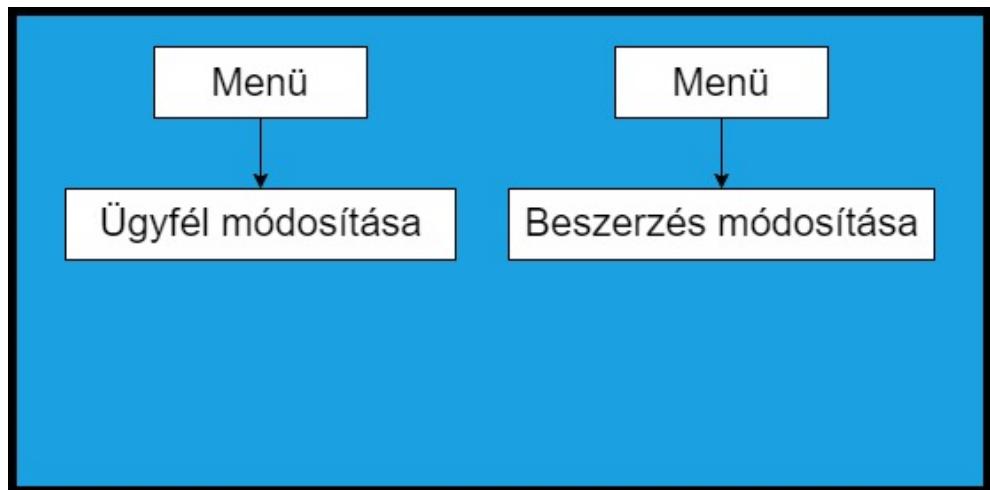
A menüpontban új felhasználót lehet hozzáadni, minden egyik mező kitöltése kötelező.



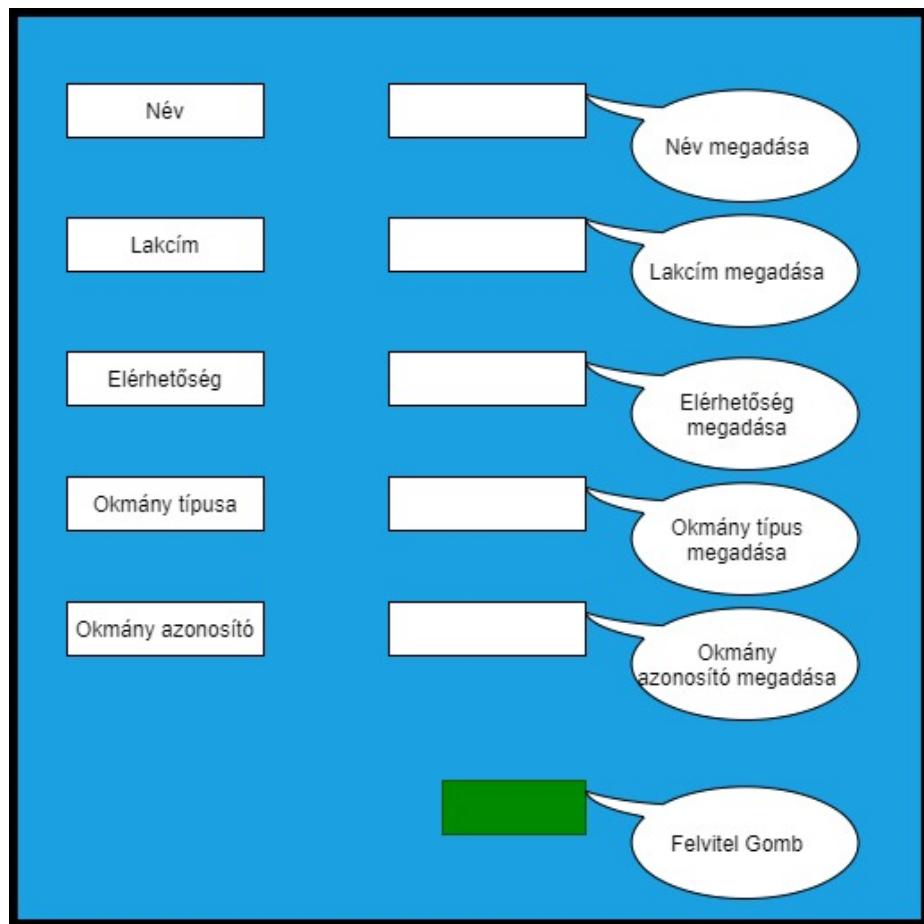
17. ábra Felhasználó hozzáadása ablak GUI terv

3.5 Szerviz módosítása

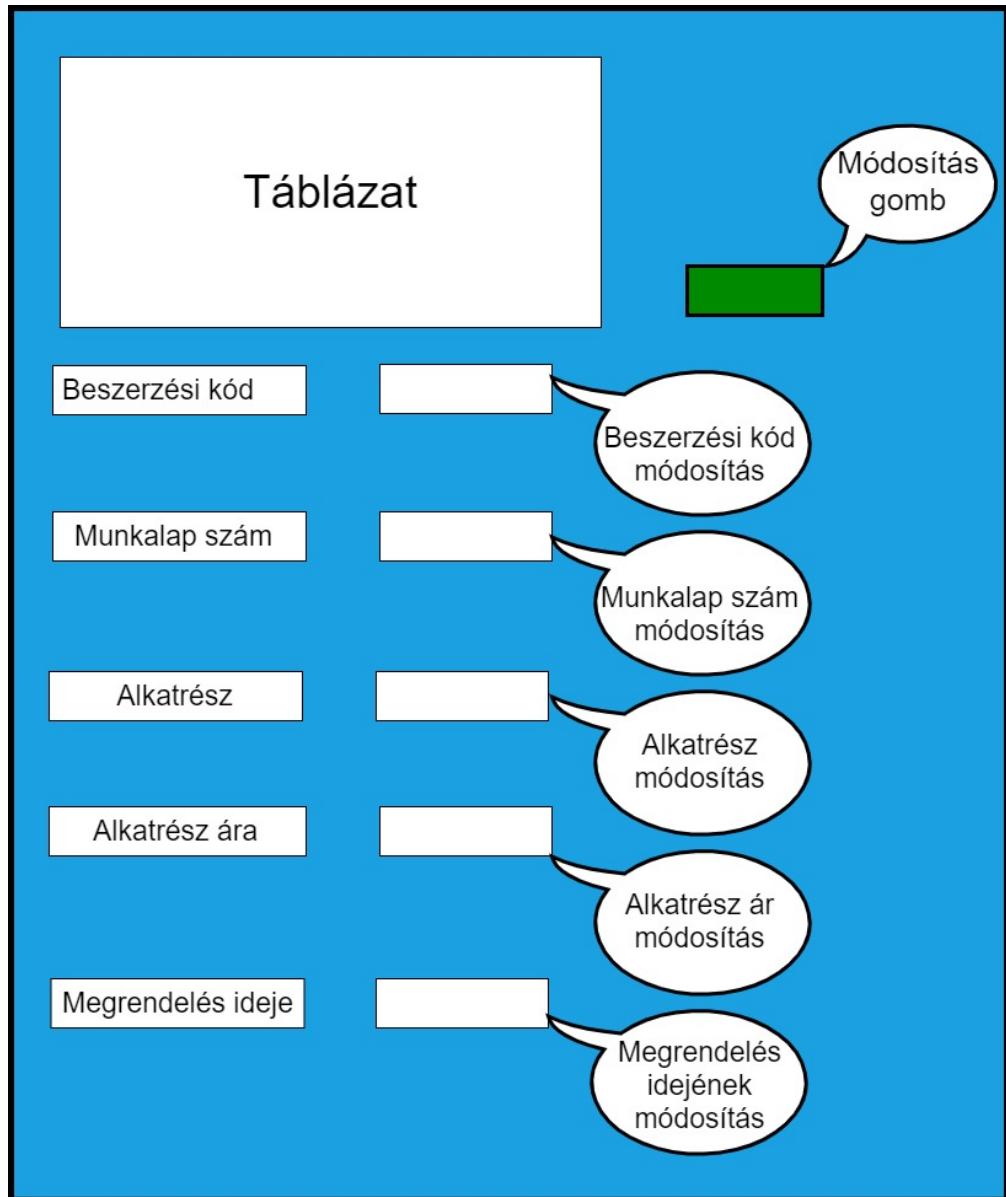
A szerviz módosítása menüpontba az ügyfél adatait, és a beszerzéshez kapcsolatos adatokat lehet megváltoztatni.



18. ábra Szerviz módosítása ablak GUI terv



19. ábra Szerviz ügyfél módosítása GUI terv



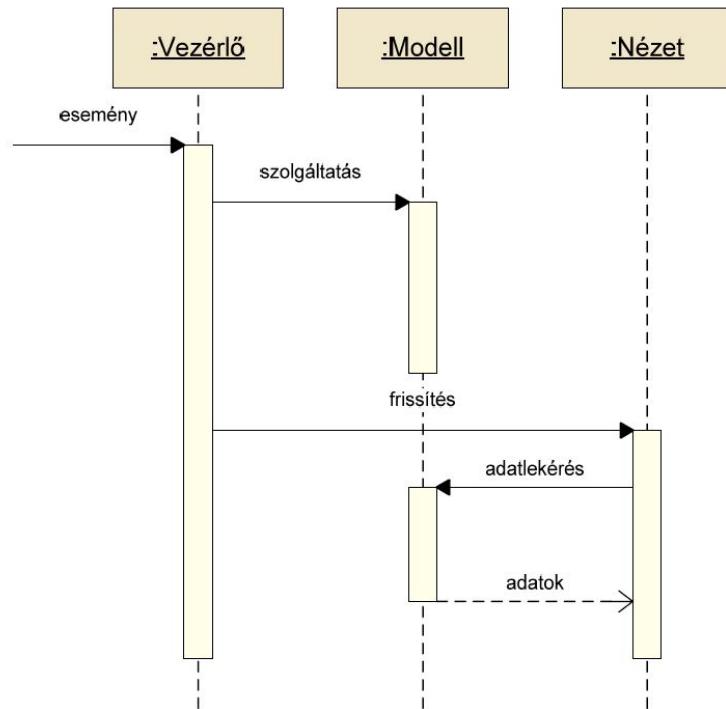
20. ábra Szerviz beszerzés módosítása GUI terv

3.6 MVC

A program az MVC architekturális tervezési minta alapján került elkészítésre. Ebben a tervezési eljárásban a fejlesztő külön részekre választja az alkalmazását. Ezek a részek a model, view és a controller. Magyarul a modell, nézet és a vezérlő. minden résznek megvan a maga feladatköre és szerepe.

- A modell az adatok tárolására.
- A nézet a képernyőn való megjelenítésre
- A vezérlő a belépési pont és biztosítja az eseménykezelést.

A felhasználó által indított eseményt a vezérlő kezeli. Ilyen esemény lehet például egy gomb megnyomása. A vezérlő eldönti, hogy megváltoztatja-e a modellt vagy a nézetet, esetleg mind a kettőt. Amennyiben a modellt megváltoztatta, abban az esetben az értesítést küld a nézetek a változásról. Ezen értesítés hatására olvassa ki a nézet a modellből a megváltozott adatot, majd megjeleníti azt. A 24. ábra az MVC-modell működése (forrás: MVC_mintak.ppt) az én programomhoz való működését ábrázolja.

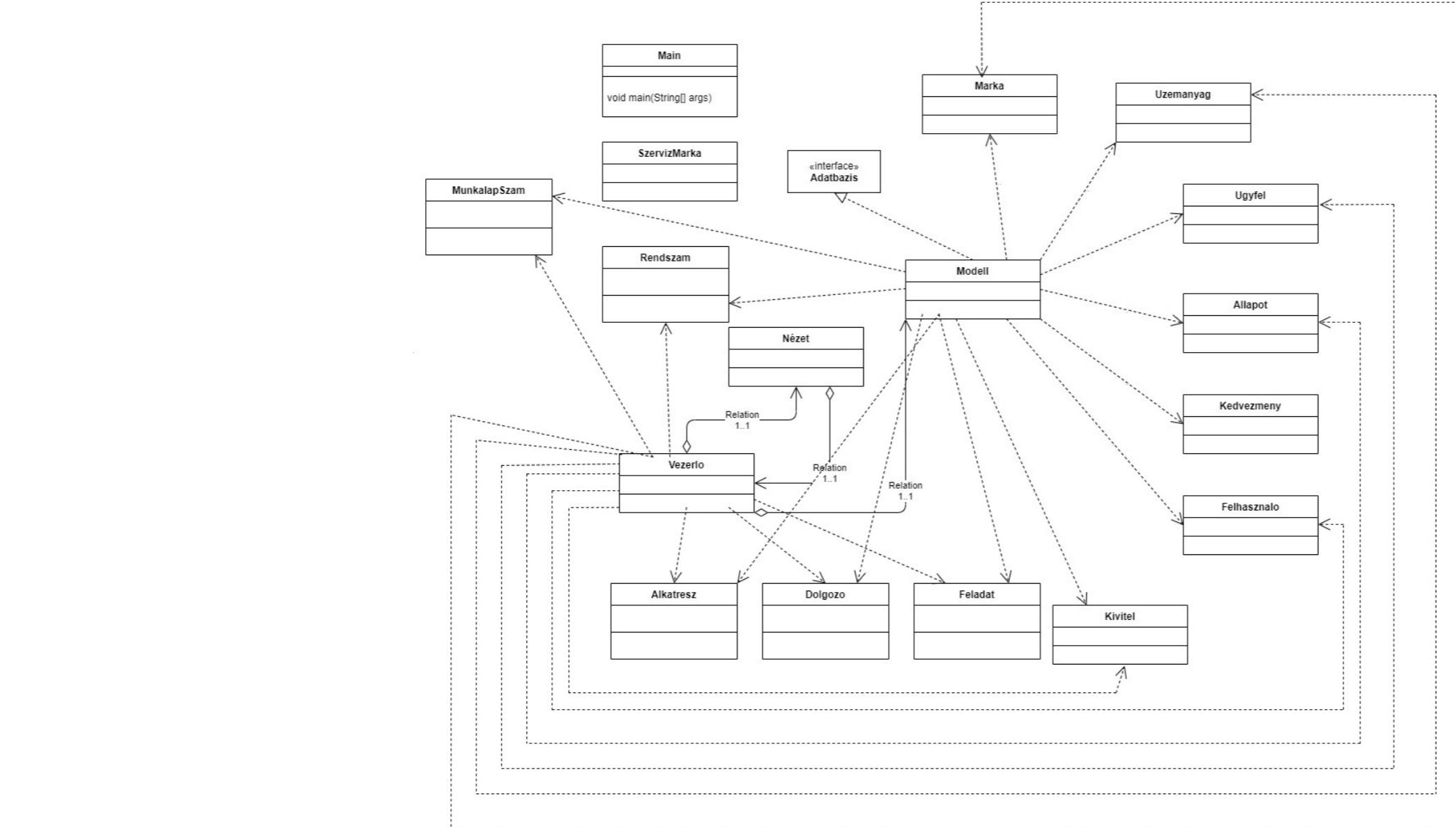


21. ábra MVC modell (forrás: MVC_mintak.ppt)

3.6.1 UML ábra

Az alkalmazás teljes diagramját a következő linken keresztül lehet megtekinteni.

[Járműkezelő Szoftver UML diagram](#)



22. ábra Áttekintő osztálydiagram

4. Megvalósítás

A bevezetés, a feladat specifikáció, és a tervezés meghatározása után következzen a megvalósítás néhány részlete. Négy nagyobb részre bontanám a megvalósítást. Ezeket a részeket lennebb fejtem ki. A forráskódot megpróbálom abban a sorrendben bemutatni, ahogy az alkalmazás működése során feldolgozásra kerül. A változók, metódusok és objektumok névadásánál törekedtem a könnyen érthetőségre, és arra, hogy az azonosítók valamennyire kifejezzék a programban betöltött szerepüket. A dolgozatban a legfontosabb, legnehezebb részeket emelem ki, ami alapján a program működése megérthető. A programot a Main osztály indítja el, ez az osztály felelős a Vezérlő osztály létrehozásáért. A Vezérlő osztály elindítja a Modell és a Nézet osztályt. A Modell osztály tartalmazza a POJO (Plain Old Java Object) osztályokat. A POJO egy olyan Java-objektum, amelyet nem köt semmiféle korlátozás, azokon kívül, amelyek a Java nyelvi specifikációjából adódnak. Ezenkívül az adatbázissal való kapcsolati metódusokat tartalmazza. A Nézet osztály tartalmazza a felhasználói felület elemeit, és megadja, hogy nézzen ki a program, amit a felhasználó lát. Ebben az osztályban találhatóak a szöveges mezőket validáló metódusok, és az osztályban szereplő komponensek eléréséhez szükséges metódusok(getter).

4.1 Járművek hozzáadásának megvalósítása

Járműhözadása / módosítása menüben lehet új járművet hozzá adni. Miután a felhasználó kiválasztotta a cmbJarmuTipusa JComboBox nevű komponenssel az adott járműtípusat, megadhatja a felvinni kívánt jármű adatait. Mikor megnyomja a „Jármű hozzáadása” nevű gombot a felhasználó, akkor először is lefutnak sorban a következő validáló metódusok: „JarmuHozzRendszamValidalas()”, JarmuHozzMarkaValidalas(), JarmuHozzModellValidalas(), JarmuHozzTipusValidalas(), JarmuHozzLeirasValidalas(), JarmuHozzHibaValidalas(), JarmuHozzHengerValidalas(), JarmuHozzKilometerValidalas(), JarmuHozzTeljesitmenyValidalas(), JarmuHozzArValidalas(), JarmuHozsSajattomegValidalas().” minden esetben, mikor a beírt adatt nem egyezik a megadott feltételnek, egy felugró ablakban jelzi a felhasználó felé, hogy javítsa ki. Miután minden adatott helyesen adott meg a felhasználó, akkor a kiválasztott jármű típusa alapján három a vezérlő osztályban található metódus hívódik meg. Mindegyik esetben meghívódik a „jarmuHozzaAdasaSzovegesMezokTorlese()” nevű metódus. Ha a

személyautó van kiválasztva akkor a „jarmuAutoAdatokFeltolt()” nevű metódus indul el. Amikor a motorkerékpár van kiválasztva akkor a „jarmuMotorAdatokFeltolt()” nevű metódus indul el. Egyébként a „jarmuKamionAdatokFeltolt()” nevű metódus hívódik meg. Mind a három metódusnak a működési elve ugyan az, a szöveges mezőkbe beírt és az éppen kiválasztott tartalom a lenyíló mezőkből egy segélistán keresztül átadja a modell osztályban levő „szemelyautoAdatokFeltolt()” nevű metódusnak az értékeket. Az átadott értékek kerülnek fel új adatként az adatbázisba.

4.2 Járművek megjelenítésének megvalósítása

A járművek feliratú gombot kell a felhasználónak megnyomni, hogy eljusson ebbe a menüpontba. Mindhárom jármű típushoz tartozik egy panel. Mindegyik panelen található egy keresés gomb, amit megnyomva megjelenik a keresett jármű találati táblázata, a találati panelben, melyek kereséstől függetlenül jelnek meg az adatbázisban tárolt járművek szerint.

4.3 Szerviz működésének megvalósítása

A szerviz menüpontot nyolc panelre van osztva az átláthatóság és könnyebb kezelés érdekében. Mindegyik panelen található egy „felvitel” nevű gomb, mely panelektől függetlenül ugyan azt a célt szolgálja, az adatok felvitelét az adatbázisba, és a megadott paramétereket egy listába ömleszti, majd a beszerzés panelen található „Munkalap készít” nevű gomb megnyomásával PDF dokumentumot generál a felhasználó által beírt adatok alapján. A felvitel gomb megnyomás után elindulnak a validáló metódusok, melyek az adott felülettől függetlenül mindenki szöveges mező tartalmát megvizsgálják az éppen aktuális panelen természetesen. Az ügyfél felvitelénél használt validáló metódusok a következők: „SzervizUgyfelNevValidallas()”, „SzervizUgyfelLakcimValidallas()”, „SzervizUgyfelElerhetosegValidallas()”, „SzervizUgyfelOkmanyTipusValidallas()”, „SzervizUgyfelOkmanyAzonValidallas()”. Ha mindegyik szöveges mező tartalma megfelel a validálásnak, akkor elindul a vezérő osztályban található „ugyfelAdatokFeltolt()” nevű metódus. A modell osztályban található „ugyfelAdatokFeltolt()” nevű metódus a „ugyfellista()” nevű metódus segítségével feltölti a felhasználó által megadott adatokat.

4.4 Felhasználók kezelése

Az alkalmazás a bejelentkezési ablakkal indul, ahol a beírt felhasználónevet és jelszót ellenőrzi a tárolt felhasználónévvel és jelszóval. A belépés gombbal indul el az esemény, ami ellenőrzi az adatokat. A belépés gombra kattintva elindul egy esemény, amely először ellenőrzi a mezők tartalmát. Mikor a mezők tartalma üres, értesíti a felhasználót erről. Ezek után ellenőrzi a vezérlő osztályban található „`belephet()`” nevű metódus, hogy a megadott felhasználónév és jelszó egyezik-e az adatbázisban tárolt felhasználónévvel és jelszóval. Abban az esetben, mikor a felhasználó helyesen adta meg a bejelentkezési adatokat, akkor a „`belepes()`” nevű metódus hívódik meg, mely tovább irányítja a felhasználót a főmenübe. Miután az Admin belépett az alkalmazásba, a felhasználói fiókok kezelése nevű menübe lépve, új felhasználót tud hozzá adni. Mikor a felhasználó megnyomja a Felvitel nevű gombot, elindul a szöveges mezők validálása a következő nevű metódusokkal az adott sorrendben: „`vezetekNevValidalas()`, `keresztNevValidalas()`, `regFelhasznaloNevValidalas()`, `regJelszoValidalas()`, `regJelszoUjraValidalas()`”. Ezek után ellenőrzi a program, hogy a két megadott jelszó egyezik-e. Mikor egyeznek a jelszavak és minden mező értéke megfelel a validálásnak, akkor lefut a vezérlő osztályban levő „`felhasznaloAdatokFeltolt()`” nevű metódus, és vissza jelzi a felhasználónak egy felugró ablakban, hogy sikeres volt a felhasználó felvitele. A „`felhasznaloAdatokFeltolt()`” nevű metódus a „`felhasznaloLista()`” nevű metódus segítségével értékül adja a megadott adatokat a modell osztályban található „`felhasznaloAdatokFeltolt()`” nevű metódusnak. A metódus beszűrja a megadott adatokat az adatbázisba.

4.5 Osztályok leírása

4.5.1 Main

A Main osztály a program belépési pontja. Ez az osztály indítja el a vezérlő osztályt, ami a nevéből is kiindulva vezérli a programot.

Metódusok:

`void Main():` A metódus indítja el a programot, és deklarálja a Vezérlő osztályt.

4.5.2 Adatbázis

Az adatbázis osztály egy interfész, ami tartalmazza a Modell osztályhoz használt lekérdezéseket, adat beszúrásokat, frissítéseket és törléseket.

Attribútumok:

`String IP_LOCAL:` A szerver fajtája.

`String DATABASE_NAME:` Az adatbázis neve.

`String DRIVER:` Az adatbázishoz való csatlakozási driver megnevezése.

`String URL:` Az adatbázishoz való csatlakozási URL.

`String USER:` Az adatbázishoz való csatlakozási felhasználónév.

`String PASSWORD:` Az adatbázishoz való csatlakozási jelszó.

`String SQL_FELHASZNALO_ADATOK:` A felhasználó nevű táblából lekérdezi a felhasználók kódját, vezetéknévét, keresztnévét, felhasználónevét és jelszavát.

`String SQL_SZERVIZ_MARKA:` A márka nevű táblából lekérdezi a márka kódját és elnevezését, ezenkívül rendezi a lekérdezett sorokat az elnevezés szerint.

`String SQL_UGYFEL:` Az ügyfél nevű táblából lekérdezi az ügyfél személy azonosító számát, típusát, nevét, elérhetőségét és lakkímét, ezenkívül rendezi a lekérdezett sorokat az ügyfelek neve alapján.

`String SQL_RENDSZAM:` Az autó nevű táblából lekérdezi a rendszámokat, és rendezi a sorokat a rendszámok szerint.

`String SQL_MUNKALAP_SZAM:` A munkalap nevű táblából lekérdezi a munkalap számot, és rendezi a sorokat a munkalap szám alapján.

`String SQL_DOLGOZO:` A dolgozók nevű táblából lekérdezi a dolgozók kódját és nevét, és rendezi a sorokat a dolgozók neve szerint.

`String SQL_FELADAT:` A feladat nevű táblából lekérdezi a feladat kódját és az elnevezését, ezenkívül rendezi a sorokat a feladat kódja alapján.

`String SQL_BESZERZES_MUNKALAP_SZAM:` A szerelés nevű táblából lekérdezi a munkalap számát, és rendezi a sorokat a munkalap szám szerint.

`String SQL_ALKATRESZ:` Az alkatrész nevű táblából lekérdezi az alkatrészek megnevezését, és rendezi a sorokat a megnevezés alapján.

`String SQL_AUTO_ADATOK_MODOSIT_TABLAZAT:` minden adatot lekérdez a kereskedés autó táblából.

`String SQL_AUTO_MARKA:` Az összes adatbázisban tárolt márkat lekérdezi a kereskedés autó táblából. Motorkerékpárokra és haszonjárművekre is van ilyen lekérdezés, de mivel használatukat tekintve megegyeznek nem fejtem ki bővebben őket.

`String SQL_AUTO_UZEMANYAG:` Az összes adatbázisban tárolt üzemanyag fajtát lekérdezi a kereskedés autó táblából. Motorkerékpárokra és haszonjárművekre is van ilyen lekérdezés, de mivel használatukat tekintve megegyeznek nem fejtem ki bővebben őket.

`String SQL_AUTO_KIVITEL:` Az összes adatbázisban tárolt jármű kivitelt lekérdezi a kereskedés autó táblából. Motorkerékpárokra és haszonjárművekre is van ilyen lekérdezés, de mivel használatukat tekintve megegyeznek nem fejtem ki bővebben őket.

`String SQL_AUTO_ALLAPOT:` Az összes adatbázisban tárolt jármű állapotot lekérdezi a kereskedés autó táblából. Motorkerékpárokra és haszonjárművekre is van ilyen lekérdezés, de mivel használatukat tekintve megegyeznek nem fejtem ki bővebben őket.

`String SQL_INSERT_FELHASZNALO:` A felhasználó nevű táblába szúrja be a felhasználó vezetéknévét, keresztnévét, felhasználónévét és jelszavát.

`String SQL_INSERT_UGYFEL:` Az ügyfél nevű táblába szúrja be az ügyfél személyazonosító okmány számát, típusát, nevét, elérhetőségét és lakcímét.

`String SQL_INSERT_MARKA`: A márka nevű táblába szúrja be a márka kódját és elnevezését.

`String SQL_INSERT_AUTO`: Az autó nevű táblába szúrja be az autó rendszámát, ügyfél azonosító számát, az autó márkáját, évjáratát, típusát, színét, motorszámát és az alvázszámat.

`String SQL_INSERT_MUNKALAP`: A munkalap nevű táblába szúrja be a munkalap számát, az autó rendszámát, a hiba jegyzéket, munkadíját, anyagdíját és a vállalási határidőt.

`String SQL_INSERT_FELADAT`: A feladat nevű táblába szúrja be a feladat kódját, a munkadíjat, a feladat elnevezését és típusát.

`String SQL_INSERT_SZERELES`: A szerelés nevű táblába szúrja be a szerelés munkalap számát, a hozzá rendelt szerelőt, a hozzá rendelt feladatot, a szerelés kezdési idejét, a szerelés befejezési idejét és a szereléssel eltöltött munkaórák számát.

`String SQL_INSERT_ALKATRESZ`: Az alkatrész nevű táblába szúrja be az alkatrész kódját és megnevezését.

`String SQL_INSERT_BESZERZES`: A beszerzés nevű táblába szúrja be a beszerzési kódöt, a beszerzéshez hozzárendelt munkalap számát, az alkatrész kódját, az alkatrész megrendelési árát és a megrendelési idejét.

`String SQL_INSERT_KERESKEDES_AUTO`: A kereskedés autó nevű táblába szúrja be a személyautó rendszámát, márkáját, modelljét, típusát, a gyártási évjáratát, a gyártási hónapját, a kivitelét, az állapotát, az üzemanyagát, a hengerűrtartalmát, kilométeréra állását, a saját tömegét, a teljesítményét, a színét, a sebességváltó fajtáját, a személyautó rövid leírását, a személyautó hibáit, azt, hogy éppen a telephelyen van-e és a vételárát. A motor és kamion táblákra is van insert parancs, mivel használatukat tekintve megegyeznek, nem fejtem ki bővebben őket.

`String SQL_UPDATE_KERESKEDES_AUTO:` A kereskedés autó nevű tábla sorait frissíti az adott adattokkal. Motorkerékpárokra és haszonjárművekre is van ilyen lekérdezés, de mivel használatukat tekintve megegyeznek, nem fejtem ki bővebben őket.

`String SQL_DELETE_KERESKEDES_AUTO:` A kereskedés autó nevű tábla adott sorát törli ki a parancs. Motorkerékpárokra és haszonjárművekre is van ilyen lekérdezés, de mivel használatukat tekintve megegyeznek, nem fejtem ki bővebben őket.

`String SQL_UPDATE_UGYFEL_ADATOK:` Az ügyfél nevű tábla sorait frissíti az adott adattokkal.

`String SQL_BESZERZES_ADATOK_TABLAZAT:` A beszerzés nevű táblából lekérdezi a beszerzés kódját, munkalap számot, alkatrész, az alkatrész árát, és a megrendelés idejét, ezenkívül rendezi a lekérdezett sorokat a beszerzés kódja alapján.

`String SQL_UPDATE_BESZERZES_ADATOK:` A beszerzés nevű tábla sorait frissíti az adott adatokkal.

4.5.3 Modell

A Modell osztály felelős az adatok lekérdezésért, felviteléhez és módosításához. Az adatbázis interfész implementálva van a modell osztályhoz. Az interfészben lévő lekérdezések alapján az implementáló osztály vezérlő metódusai férnek hozzá a modellben tárolt adatokhoz.

Attribútumok:

`String jelszo:` Felhasználó jelszó feltöltéséhez szükséges segéd változó, ezt a változót a jelszoTitkositas metódus hívja meg.

`String modellTitkositottJelszo:` A jelszoTitkositas metódus vissza térei értéke, ennek az adattagnak az értéké töltöm fel az adatbázisba.

`boolean vanDriver:` A driver betöltéséhez szükséges változó.

`File htmlFajl:` Az index.html fájl létrehozásáért felelős változó.

`String html`: A HTML fájl egy string típusú adatt mely, tartalmazza a weboldal alap elemeit.

Metódusok:

`boolean driverBetolt()`: A metódus betölti a „DRIVER” nevű változót.

`void HTMLFajltGeneral(String, String)`: A metódus a két paraméter alapján: fejléc és az adat, generálja az adatokat, és hozzá adja a „html”, nevű változóhoz.

`ArrayList<Felhasznalo> belepesAdatokFeltoolt()`: Új felhasználó adatainak feltöltése az adatbázisba.

`ArrayList<SzervizMarka> getSzervizMarkaLista()`: A metódus a márka nevű táblából kérdezi le a márkkák kódját és elnevezését.

`ArrayList<Ugyfel> getUgyfellista()`: A metódus az ügyfél nevű táblából kérdezi le a személyazonosító okmány számát, személyazonosító okmány típusát, az ügyfél nevét, elérhetőségét, lakcímét.

`ArrayList<Rendszam> getRendszamLista()`: A metódus az autók nevű táblából kérdezi le a létező autók rendszámokat.

`ArrayList<MunkalapSzam> getMunkalapSzamLista()`: A metódus a munkalap nevű táblából kérdezi le az összes létező munkalap számokat.

`ArrayList<Dolgozo> getDolgozoLista()`: A metódus a dolgozó nevű táblából kérdezi le a dolgozók kódját és nevét.

`ArrayList<Feladat> getFeladatLista()`: A metódus a feladat nevű táblából kérdezi le az összes feladat kódját, és elnevezését.

`ArrayList<MunkalapSzam> getBeszerzesMunkalapSzamLista()`: A metódus a szerelés nevű táblából kérdezi le az összes munkalap számot.

`ArrayList<Alkatresz> getAlkatreszLista()`: A metódus az alkatrész nevű táblából kérdezi le az összes alkatrész megnevezését.

`ArrayList<Marka> getAutoMarkaLista()`: A metódus a kereskedés autó nevű táblából kérdezi le az összes autó márkáját. Motorkerékpárokra és haszonjárművekre is van ilyen metódus, de mivel használatukat tekintve megegyeznek, nem fejtem ki bővebben őket.

`ArrayList<Uzemanyag> getAutoUzemanyagLista()`: A metódus a kereskedés autó nevű táblából kérdezi le az összes létező autóhoz tartozó üzemanyag típusokat. Motorkerékpárokra és haszonjárművekre is van ilyen metódus, de mivel használatukat tekintve megegyeznek, nem fejtem ki bővebben őket.

`ArrayList<Kivitel> getAutoKivitelLista()`: A metódus a kereskedés autó nevű táblából kérdezi le az összes létező autóhoz tartozó kiviteli lehetőségeket. Motorkerékpárokra és haszonjárművekre is van ilyen metódus, de mivel használatukat tekintve megegyeznek, nem fejtem ki bővebben őket.

`ArrayList<Allapot> getAutoAllapotLista()`: A metódus a kereskedés autó nevű táblából kérdezi le az összes létező autóhoz tartozó állapot lehetőségeket. Motorkerékpárokra és haszonjárművekre is van ilyen metódus, de mivel használatukat tekintve megegyeznek, nem fejtem ki bővebben őket.

`void felhasznaloAdatokFeltoolt(ArrayList<String>)`: A lista elemeit tölti fel az adatbázisba, mielőtt a jelszó adattagot feltöltené, előbb át adja a „jelszoTitkositas()” nevű metódusnak. A metódus vissza térsí értékét töltöm fel az adatbázisba.

`void ugyfelAdatokFeltoolt(ArrayList<String>)`: A felhasználó által megadott szerviz menüben lévő ügyfél panel adatait tölti fel az adatbázisba.

`void ugyfelAdatokModosit(ArrayList<String>)`: A felhasználó által módosított, a szerviz módosítás menüben lévő ügyfél adatait frissíti az adatbázisban.

void markaAdatokFeltolt(ArrayList<String>): A felhasználó által megadott szerviz menüben lévő márka panel adatait tölti fel az adatbázisba.

void autoAdatokFeltolt(ArrayList<String>): A felhasználó által megadott szerviz menüben lévő autó panel adatait tölti fel az adatbázisba.

void munkalapAdatokFeltolt(ArrayList<String>): A felhasználó által megadott szerviz menüben lévő munkalap panel adatait tölti fel az adatbázisba.

void feladatAdatokFeltolt(ArrayList<String>): A felhasználó által megadott szerviz menüben lévő feladat panel adatait tölti fel az adatbázisba.

void szerelesAdatokFeltolt(ArrayList<String>): A felhasználó által megadott szerviz menüben lévő szerelés panel adatait tölti fel az adatbázisba.

void alkatreszAdatokFeltolt(ArrayList<String>): A felhasználó által megadott szerviz menüben lévő alkatrész panel adatait tölti fel az adatbázisba.

void beszerzesAdatokFeltolt(ArrayList<String>): A felhasználó által megadott szerviz menüben lévő beszerzés panel adatait tölti fel az adatbázisba.

void beszerzesAdatokModosit(ArrayList<String>): A felhasználó által módosított, a szerviz módosítás menüben lévő beszerzés adatait frissíti az adatbázisban.

void szemelyautoAdatokFeltolt(ArrayList<String>): A felhasználó által megadott jármű hozzáadása menüpontban levő adatokat tölti fel az adatbázisba.

Motorkerékpára, és a haszonjárművekre is van adatok feltölt metódus, de mivel működésüket tekintve megegyeznek nem fejtem ki őket.

void szemelyautoAdatokModosit(ArrayList<String>): A felhasználó által módosított jármű módosítás/törlés menüpontban levő adatokat frissíti az adatbázisban. Motorkerékpára, és a haszonjárművekre is van adatok módosít metódus, de mivel működésüket tekintve megegyeznek nem fejtem ki őket.

`void szemelyautoAdatokTorol(String):` A felhasználó által törölt jármű módosítás/törlés menüpontban levő adatokat törli az adatbázisban. Motorkerékpára, és a haszonjárművekre is van törlési metódus, de mivel működésüket tekintve megegyeznek nem fejtem ki őket.

`String jelszoTitkositas(String, String):` A metódusnak két bemenő paramétere van: az első paraméter a megadott jelszót, melyet titkosít a metódus, a második paraméter pedig a salt. A metódus visszatérési értéke a titkosított jelszó.

`DefaultTableModel getAutoAdatok():` A metódus a kereskedés autók nevű táblából kérdezi le az össze adatot, majd átalakítja a lekérdezett adatokat, olyan módon, hogy azokat táblázat formájába meg tudja jeleníteni. Motorkerékpára, és a haszonjárművekre is van adatok megjelenítés táblázatos formában metódus, de mivel működésüket tekintve megegyeznek nem fejtem ki őket.

`DefaultTableModel getUgyfelAdatokTablazat():` A metódus az ügyfél nevű táblából kérdezi le az össze adatot, majd átalakítja a lekérdezett adatokat, olyan módon, hogy azokat táblázat formájába meg tudja jeleníteni.

`DefaultTableModel getBeszerzesAdatokTablazat():` A metódus a beszerzsé nevű táblából kérdezi le az össze adatot, majd átalakítja a lekérdezett adatokat, olyan módon, hogy azokat táblázat formájába meg tudja jeleníteni.

A Modell osztály tartalmazz több POJO osztályt, ezek a következők: Felhasznalo, SzervizMarka, Marka, Ugyfel, Kedvezmeny, Kivitel, Uzemanyag, Allapot, Rendszam, Munkalapszam, Dolgozo, Feladat, Alkatresz. Mindegyik osztály tartalmazz a szükséges adattagokat, konstruktörököt, gettereket, és ToString metódusokat.

4.5.4 Vezérlő

A vezérlő osztály irányítja a programot. Ez az osztály indítja el a modell és a nézet osztályt.

Attribútumok:

Modell `modell:` A modell osztály példányosításához szükséges adattag.

Nezet **nezet**: A nézet osztály példányosításához szükséges adattag.

Timer **t**: Az időzítőhöz szükséges adattag.

Metódusok:

void setAutoAdatokTablazat(): A nézet osztályban található járművek módosítása táblázatnak adja át a modell osztályban lévő „**getAutoAdatok()**” nevű metódust. A motorkerékpárokra, és a haszonjárművekre is van táblázat beállító metódus, de mivel működésüket tekintve megegyeznek nem fejtem ki őket.

ArrayList<String> felhasznaloLista(String, String, String, String): Felhasználó segédlista az adatbázisba feltöltött adatokhoz.

ArrayList<String> ugyfelLista(String, String, String, String, String): Ügyfél segédlista az adatbázisba feltöltött adatokhoz.

ArrayList<String> markaLista(String, String): Márka segédlista az adatbázisba feltöltött adatokhoz.

ArrayList<String> autoLista(String, String, String, String, String, String, String, String, String): Autó segédlista az adatbázisba feltöltött adatokhoz.

ArrayList<String> munkalapLista(String, String, String, String, String, String): Munkalap segédlista az adatbázisba feltöltött adatokhoz.

ArrayList<String> feladatLista(String, String, String, String): Feladat segédlista az adatbázisba feltöltött adatokhoz.

ArrayList<String> szerelesLista(String, String, String, String, String, String): Szerelés segédlista az adatbázisba feltöltött adatokhoz.

ArrayList<String> alkatreszLista(String, String): Alkatrész segédlista az adatbázisba feltöltött adatokhoz.

`ArrayList<String> beszerzesLista(String, String, String, String, String):` Beszerzés segédlista az adatbázisba feltöltött adatokhoz.

`ArrayList<String> szemelyautoLista(String, String, String):` Személyautó segédlista az adatbázisba feltöltött adatokhoz. Motorkerékpárra és haszonjárműre is van ilyen metódus, de mivel működésüket tekintve megegyeznek, nem fejtem ki őket.

`void felhasznaloAdatokFeltolt():` Felhasználók adatainak adatbázisba való feltöltéséhez szükséges vezérlő metódusa.

`void ugyfelAdatokFeltolt():` Ügyfelek adatainak adatbázisba való feltöltéséhez szükséges vezérlő metódusa.

`void ugyfelAdatokModosit():` Ügyfelek adatainak adatbázisban való módosításához szükséges vezérlő metódusa.

`void markaAdatokFeltolt():` Szervizben levő autók márka adatainak adatbázisba való feltöltéséhez szükséges vezérlő metódusa.

`void autoAdatokFeltolt():` Szervizben levő autók adatainak adatbázisba való feltöltéséhez szükséges vezérlő metódusa.

`void munkalapAdatokFeltolt():` Szervizben levő munkalap adatainak adatbázisba való feltöltéséhez szükséges vezérlő metódusa.

`void feladatAdatokFeltolt():` Szervizben levő feladat adatainak adatbázisba való feltöltéséhez szükséges vezérlő metódusa.

`void szerelesAdatokFeltolt():` Szervizben levő szerelés adatainak adatbázisba való feltöltéséhez szükséges vezérlő metódusa.

void alkatreszAdatokFeltolt(): Szervizben levő alkatrész adatainak adatbázisba való feltöltéséhez szükséges vezérlő metódusa.

PdfPCell cella(String, int, boolean, int): A PDF generáláshoz szükséges PDF-ben található cellák tulajdonságainak beállításához szükséges metódus.

void pdfGeneral(): Munkalapot generál PDF formátumba a Szerviz menü adott ügyfél, autó, munkalap, feladat, szerelés és beszerzés adatai alapján.

void beszerzesAdatokFeltolt(): Szervizben levő beszerzés adatainak adatbázisba való feltöltéséhez szükséges vezérlő metódusa.

void jarmuAutoAdatokFeltolt(): Jármű hozzáadása menüben való járművek adatainak feltöltéséhez szükséges vezérlő metódusa. Motorkerékpárra és haszonjárműre is van ilyen metódus, de mivel működésüket tekintve megegyeznek, nem fejtem ki őket.

void jarmuAutoAdatokModosit(): Jármű módosítás/törlés menüben való járművek adatainak módosításához szükséges vezérlő metódusa. Motorkerékpárra és haszonjárműre is van ilyen metódus, de mivel működésüket tekintve megegyeznek, nem fejtem ki őket.

void jarmuAutoAdatokTorol(): Jármű módosítás/törlés menüben való járművek adatainak törléséhez szükséges vezérlő metódusa. Motorkerékpárra és haszonjárműre is van ilyen metódus, de mivel működésüket tekintve megegyeznek, nem fejtem ki őket.

boolean belephet(): A belépés panelben megadott felhasználónévet és jelszót hasonlítja össze az adatbázisban tárolt felhasználónévvel és jelszóval.

String ido(): A főmenüben megjelenő pontos idő kijelzéséhez szükséges metódus.

DefaultComboBoxModel szervizMarkaListaKeszit(ArrayList<Szerviz-Marka>): Kivitel lista készítéséhez szükséges metódus. A kivitelekhez használt legörökült menübe való adatokat adja át.

`DefaultComboBoxModel ugyfelListaKeszit(ArrayList<Ugyfel>):` Ügyfél lista készítéséhez szükséges metódus. Az ügyfelekhez használt legördülő menübe való adatokat adja át.

`DefaultComboBoxModel rendszamListaKeszit(ArrayList<Rendszam>):` Rendszám lista készítéséhez szükséges metódus. A rendszámokhoz használt legördülő menübe való adatokat adja át.

`DefaultComboBoxModel munkalapszamListaKeszit(ArrayList<MunkalapSzam>):` Munkalapszám lista készítéséhez szükséges metódus. A munkalapszámokhoz használt legördülő menübe való adatokat adja át.

`DefaultComboBoxModel dolgozoListaKeszit(ArrayList<Dolgozo>):` Dolgozó lista készítéséhez szükséges metódus. A dolgozókhoz használt legördülő menübe való adatokat adja át.

`DefaultComboBoxModel feladatListaKeszit(ArrayList<Feladat>):` Feladat lista készítéséhez szükséges metódus. A feladatakhoz használt legördülő menübe való adatokat adja át.

`DefaultComboBoxModel alkataszListaKeszit(ArrayList<Alkatasz>):` Alkatrész lista készítéséhez szükséges metódus. Az alkatrészekhez használt legördülő menübe való adatokat adja át.

`DefaultComboBoxModel ugyfelModositListaKeszit(ArrayList<Ugyfel>):` Az ügyfél módosítása lista készítéséhez szükséges metódus. Az ügyfél módosításához használt legördülő menübe való adatokat adja át.

`void jarmuKeresesAuto():` A nézet osztályban található találati táblázatnak adja át a modell osztályban lévő „getAutoAdatok()” nevű metódust. Motorkerékpárra és haszonjárműre is van ilyen metódus, de mivel működésüket tekintve megegyeznek, nem fejtem ki őket.

4.5.5 Nézet

A Nézet osztály valósítja meg a GUI-t és az adatok megjelenítését a különböző komponensek használatával. Az implicit getter és setter metódusok nem kerültek jelölésre az átláthatóság fenntartása érdekében. Mindazonáltal néhány metódust leírok, mert véleményem szerint, a működésüket, és a hasznosságukat tekintve kiemelkedő szerepük van a programban.

Attribútumok:

`Vezerlo vezero`: A vezérlő osztály példányosításához szükséges adattag.

`ImageIcon autoKep`: Jármű keresés panelben lévő személyautó panel képe.

`ImageIcon motorKep`: Jármű keresés panelben lévő motorkerékpár panel képe.

`ImageIcon kamionKep`: Jármű keresés panelben lévő haszonjármű panel képe.

`ImageIcon felhasznaloKep`: Bejelentkezés panelen lévő felhasználóhoz tartozó kép.

`ImageIcon jelszoKep`: Bejelentkezés panelen lévő jelszóhoz tartozó kép.

`JFileChooser openFile`: Kép kiválasztására szolgáló fájlkiválasztó adattag.

`JFileChooser saveFile`: Kép mentésére szolgáló fájlkiválasztó adattag.

`BufferedImage bf`: Kép adattag.

`String jelszo`: Jelszó adattag, amit a bejelentkezéshez és a jelszó titkosításhoz használunk.

`String[] autoKivitel`: String típusú tömb, ami tartalmazza az összes személyautó kivitel típusokat.

`String[] motorKivitel:` String típusú tömb, ami tartalmazza az összes motorkerékpár kivitel típusokat.

`String[] kamionKivitel:` String típusú tömb, ami tartalmazza az összes haszon jármű kivitel típusokat.

`String[] motorUzemanyag:` String típusú tömb, ami tartalmazza az összes motorkerékpárhoz tartozó üzemanyag típusokat.

`String[] autoKamionUzemanyag:` String típusú tömb, ami tartalmazza az összes személyautóhoz és haszonjárműhöz tartozó üzemanyag típusokat.

Metódusok:

`void setLblIdoKijelzes():` A vezérlő osztály „ido()” nevű metódusát adja értélük a „lblIdoKijelzes” nevű adattagnak.

`boolean SzervizUgyfelNevValidalas():` Boolean típusú metódus, ami a „txtUgyfelNev” nevű szöveges mező értékét ellenőrzi a következő minta alapján: „^ [A-Za-z\\sÖÜÓÚÁÉÍ-öüöőúéáűí] {1,50} \$”.

`boolean SzervizMunkalapszamValidalas():` Boolean típusú metódus, ami a „txtMunkMunkalap” nevű szöveges mező értékét ellenőrzi a következő minta alapján: „^mu[\\/] {1} (?:19 | 20) \\d{2} [\\/] {1} [\\d]{4} \$”.

`boolean SzervizMunkalapValHataridoValidalas():` Boolean típusú metódus, ami a „txtMunkHatarido” nevű szöveges mező értékét ellenőrzi a következő minta alapján: „^ (?: ((?:19 | 20)[0-9]{2}) [\\/] \\- .)? (?: (0[1-9] | 1[0-2]) [\\/] \\- .)? ([0-2][1-8] | [12]0 | 09 | 19) | (0[13-9] | 1[0-2]) [\\/] \\- .)? (29 | 30) | (0[13578] | 1[02]) [\\/] \\- .)? (31) | (19 (?: [0][48] | [2468][048] | [13579][26]) | 20 (?: [02468][048] | [13579][26]) [\\/] \\- .)? (02) [\\/] \\- .)? (29) | ((00 | [0-9] | 1[0-9] | 2[0-3]) : ([0-9] | [0-5][0-9]) : ([0-9] | [0-5][0-9])) \$”.

```
boolean SzervizSzerelesKezdesIdoValidalas(): Boolean típusú metódus, ami a „txtSzerelesKezdido” nevű szöveges mező értékét ellenőrzi a következő minta alapján: „^((?:((?:19|20)[0-9]{2})[\\]/\\-.)?((?:0[1-9]|1[0-2])[\\]/\\-.)?([0-2][1-8]| [12]0|09|19)|(0[13-9]|1[0-2])[\\]/\\-.)?(29|30)|(0[13578]|1[02])[\\]/\\-.)?(31))|(19(?:[0][48]| [2468][048]| [13579][26]))|20(?:[02468][048]| [13579][26]))[\\]/\\-.)?(02)[\\]/\\-.)?(29))$”.
```

```
boolean SzervizSzerelesMunkaOraValidalas(): Boolean típusú metódus, ami a „txtSzerelesMOra” nevű szöveges mező értékét ellenőrzi a következő minta alapján: „^(\\d?[1-9]| [1-9]0)$”.
```

```
boolean JarmuHozzRendszamValidalas(): Boolean típusú metódus, ami a „txtRendszam” nevű szöveges mező értékét ellenőrzi a következő minta alapján: „^ [A-Z]{3}\\-[\\d]{3}$”.
```

```
boolean regJelszoValidalas(): Boolean típusú metódus, ami a „txtRegJel-szo” nevű szöveges mező értékét ellenőrzi a következő minta alapján: „^(?=.*[A-Za-z])(?=.*\\d)[A-Za-z\\d]{8,}$”.
```

Természetesen nagyon sok validációs metódust használok a programba, de próbáltam ki-emelni a legösszetettebbeket, és validálás szempontjából különböző értékek szerint.

String jelszoTitkosita(String, String): A metódusnak két bemenő paramétere van: az első paraméter a megadott jelszót, melyet titkosít a metódus, a második paraméter pedig a salt. A metódus visszatérési értéke a titkosított jelszó.

void jarmuHozzaAdasaSzovegesMezokTorlese(): A megadott szöveges mezők értékét állítja üresre.

ArrayList<String> ugyfelAdatLista(): A lista arra szolgál, hogy az adott adat-tagokat átadva PDF-et generáljak. Ezenkívül öt hasonló metódust használok erre a célra, de mivel használhatóságukat és működésüköt tekintve megegyeznek, nem fejtem ki rész-lesebben őket.

`BufferedImage kepAtmeretez(BufferedImage, int, int):` BufferedImage típusú metódus, melyet arra használok, hogy a kiválasztott képet felbontástól függetlenül, az adott felbontásra redukálja a képet. A metódus három paraméteres, az első paraméter az eredeti kép, melynek a felbontását csökkentem, a második paraméter a kép szélessége, a harmadik paraméter a kép magassága. A metódus vissza térsí értéke a csökkentett felbontású kép.

`void kepBetolt():` Az „openFile” nevű változó segítségével a kiválasztott képet betölти a „bf” nevű változóba.

`void kepMent():` A „saveFile” nevű változó segítségével a kiválasztott helyre menti el a „bf” nevű változóban tárolt adatot.

`void loEroBeallit():` A „txtTeljes” nevű szöveges mező értékét váltja át, majd értékül adja a „lblLoero” nevű változónak.

5. Tesztelés

5.1 Rendszerkövetelmények

A program futtatásához Microsoft Windows 10-es operációs szükséges, de mivel a Java, egy platformfüggetlen nyelv, véleményem szerint más operációs rendszeren is futatható, ezenkívül szükséges a Java futtatási környezet. A keresett járművek megtekintéséhez szükséges egy webböngésző, az ajánlott a Google Chrome, hogy funkcionálisában helyesen működjön minden. A szoftver futtatásához szükséges minimális hardverkövetelmények a következők:

- legalább 1 GHz-es processzor
- 1 GB RAM
- 512 MB-s videókártya
- minimum 1024x768-as felbontású monitor

5.2 Tesztelés menete

Program indítása adatbázis kapcsolat nélkül		
TESZTESET	ELVÁRT EREDMÉNY	KAPOTT EREDMÉNY
Program elindulása	Hiba!(Vezérlő konstruktor nem tudott lefutni!) Communications link failure	Hibaüzenet, nem megfelelt.
Belépés gomb megnyomása	Hiba!(Beléphet metódus nem tudott lefutni!) Communications link failure	Hibaüzenet, nem megfelelt.

Ügyfél nevének tesztelése			
BEVITT ÉR-TÉK	TESZTESET	ELVÁRT EREDMÉNY	KAPOTT EREDMÉNY
üres mező	Felvitel gomb megnyomása	Hiba! Az ügyfél nevének formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.

111	Felvitel gomb megnyomása	Hiba! Az ügyfél névnek formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.
Kiss J3nő	Felvitel gomb megnyomása	Hiba! Az ügyfél névnek formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.
„ ”	Felvitel gomb megnyomása	Hiba! Az ügyfél névnek formátuma nem megfelelő!	Nincs hibaüzenet, megfelelt
Nagy Béla	Felvitel gomb megnyomása	Az alkalmazás a következő szöveges mező tartalmát vizsgálja.	Nincs hibaüzenet, megfelelt

Ügyfél elérhetőség tesztelése				
BEVITT ÉR-TÉK	TESZTESET	ELVÁRT ERED-MÉNY	KAPOTT ERED-MÉNY	
üres mező	Felvitel gomb megnyomása	Hiba! Az ügyfél elérhetőségének formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	
„ ”	Felvitel gomb megnyomása	Hiba! Az ügyfél elérhetőségének formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	
telefonszám	Felvitel gomb megnyomása	Hiba! Az ügyfél elérhetőségének formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	
+36201234567	Felvitel gomb megnyomása	Az alkalmazás a következő szöveges mező tartalmát vizsgálja.	Nincs hibaüzenet, megfelelt	
0611234567/369	Felvitel gomb megnyomása	Az alkalmazás a következő szöveges	Nincs hibaüzenet, megfelelt	

		mező tartalmát vizsgálja.	
--	--	---------------------------	--

Autó rendszámának tesztelése				
BEVITT ÉRTÉK	TESZTESET	ELVÁRT EREDMÉNY	KAPOTT EREDMÉNY	
üres mező	Felvitel gomb megnyomása	Hiba! Az autó rendszám formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	
„ ”	Felvitel gomb megnyomása	Hiba! Az autó rendszám formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	
AAA 123	Felvitel gomb megnyomása	Hiba! Az autó rendszám formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	
abc-123	Felvitel gomb megnyomása	Hiba! Az autó rendszám formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	
ABC-123	Felvitel gomb megnyomása	Az alkalmazás a következő szöveges mező tartalmát vizsgálja.	Nincs hibaüzenet, megfelelt	

Munkalap számának tesztelése				
BEVITT ÉRTÉK	TESZTESET	ELVÁRT EREDMÉNY	KAPOTT EREDMÉNY	
üres mező	Felvitel gomb megnyomása	Hiba! A munkalap szám formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	
„ ”	Felvitel gomb megnyomása	Hiba! A munkalap szám formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	

munkalap 1234.	Felvitel gomb megnyomása	Hiba! A munkalap szám formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.
um/2020/1234	Felvitel gomb megnyomása	Hiba! A munkalap szám formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.
mu/2020/1234	Felvitel gomb megnyomása	Az alkalmazás a következő szöveges mező tartalmát vizsgálja.	Nincs hibaüzenet, megfelelt

Munkalap vállalási határ idejének tesztelése				
BEVITT ÉR-TÉK	TESZTESET	ELVÁRT ERED-MÉNY	KAPOTT ERED-MÉNY	
üres mező	Felvitel gomb megnyomása	Hiba! A munkalap vállalási határidő formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	
„ ”	Felvitel gomb megnyomása	Hiba! A munkalap vállalási határidő formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	
2020/03/30,17.00	Felvitel gomb megnyomása	Hiba! A munkalap vállalási határidő formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	
dátum	Felvitel gomb megnyomása	Hiba! A munkalap vállalási határidő formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	
2020-03-30	Felvitel gomb megnyomása	Hiba! A munkalap vállalási határidő	Hibaüzenet, nem megfelelt.	

		formátuma nem megfelelő!	
2020-03-30 19:00:00	Felvitel gomb megnyomása	Az alkalmazás a következő szöveges mező tartalmát vizsgálja.	Nincs hibaüzenet, megfelelt

Feladat munkadíjának tesztelése				
BEVITT ÉRTÉK	TESZTESET	ELVÁRT ERedmény	KAPOTT ERedmény	
üres mező	Felvitel gomb megnyomása	Hiba! A feladat munka díjának formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	
„ ”	Felvitel gomb megnyomása	Hiba! A feladat munka díjának formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	
szám	Felvitel gomb megnyomása	Hiba! A feladat munka díjának formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	
30000,0	Felvitel gomb megnyomása	Hiba! A feladat munka díjának formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.	
30000	Felvitel gomb megnyomása	Az alkalmazás a következő szöveges mező tartalmát vizsgálja.	Nincs hibaüzenet, megfelelt	

Felhasználó jelszó felvitel tesztelése

BEVITT ÉR-TÉK	TESZTESET	ELVÁRT ERED-MÉNY	KAPOTT ERED-MÉNY
üres mező	Felvitel gomb megnyomása	Hiba! A jelszó formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.
„ ”	Felvitel gomb megnyomása	Hiba! A jelszó formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.
12345	Felvitel gomb megnyomása	Hiba! A jelszó formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.
jelszo	Felvitel gomb megnyomása	Hiba! A jelszó formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.
jelszol	Felvitel gomb megnyomása	Hiba! A jelszó formátuma nem megfelelő!	Hibaüzenet, nem megfelelt.
jelszo11	Felvitel gomb megnyomása	Az alkalmazás a következő szöveges mező tartalmát vizsgálja.	Nincs hibaüzenet, megfelelt

5.3 Teszteléskor felfedezett Hibák

A tesztelést követően felfedezett hibák leírása. Az szerviz menüpontban lévő munkalaphoz tartozó munkadíj, és anyagdíj felvitele nem jó, mivel az adatok felvitelenél nem az adott adatot tölti fel, hanem egy bedrótozott adatot. Mindegyik táblázatnál, melyek adatokat jelenítenek meg az adatbázisból, miután elkezdtem görgetni a táblázatot horizontális irányban, az oszlopok nem jellenek meg, beragad. A ki buildelt JAR fájlt elindítva, nem az előre beállított felbontásban jellenek meg a komponensek a képernyőn.

6. Továbbfejlesztési lehetőségek

Továbbfejlesztési lehetőségeket én két részre bontanám. Az első rész a meglévő funkciók bővítése (karbantartás), a másik rész új funkciók bevezetése (továbbfejlesztés).

6.1 Meglévő funkciók

Meglévő funkciókat több szempontból is tovább lehetne fejleszteni néhány példa: Az alkalmazás, és az adatbázis optimalizálása, biztonsági problémák kezelésének továbbfejlesztése, felhasználó barátabb felület kialakítása. Az adatbázis táblái közt szerepel a kedvezmény tábla. Jelen pillanatban nem foglalkozom kedvezményekkel, melyek az alkalmazás szerviz részéhez kötődnek, de továbbfejlesztési szempontból az adatbázisban hagyom. A jármű keresés megvalósítása, az adott feltételek szerint.

6.2 Új funkciók

Egyéb járművek kezelését is lehessen kezelni az alkalmazással, mint például, kishaszon-járművek, munkagépek, autóbuszok, lakókocsik, hajók, és alkatrészek.

Ezenkívül az alkalmazást olyan módon is tovább lehetne fejleszteni, hogy a felhasználó a bejelentkezés pillanatában ki tudná választani a program nyelvét az általa választott nyelvre. Az alkalmazást webre is ki lehetne vezetni, olyan módon, hogy az ügyfelek tudjanak keresni az adott feltételekre szűrve a járművek között.

7. Felhasználói dokumentáció

7.1 Rendszer követelmények

A program futtatásához Microsoft Windows 10-es operációs szükséges, de mivel a Java, platformfüggetlen nyelv, véleményem szerint más operációs rendszerek is futtathatók, ahol rendelkezésre áll a Java futtatási környezet. A szoftver futtatásához szükséges minimális hardverkövetelmények a következők:

- legalább 1 GHz-es processzor
- 1 GB RAM
- 512 MB-s videókártya
- minimum 1024x768-as felbontású monitor

7.2 Telepítés és futtatás

A program nem igényel külön telepítést, egyetlen futtatható állományt tartalmaz, a „JarmukezeloSzoftver.jar”-t. A program a „JarmukezeloSzoftver.jar” nevű fájlra kattintva indítható el.

7.3 Program leírása

Az alkalmazás egy autókereskedés járműveit tartja számon, ezen felül az ügyfelek autóvak kapcsolatos szervizeléseit látja el. Három fajta járművet lehet kezeln a programban, ezek a következők:

- személyautó
- motorkerékpár
- haszonjármű

Az alkalmazást az autó kereskedők, vagy szervizzel rendelkező tulajdonosok használhatják bármikor mikor szükségük van a járművek kezelésére, vagy esetleg egy ügyfélnek szerethnék megmutatni a készleten lévő járműveket.

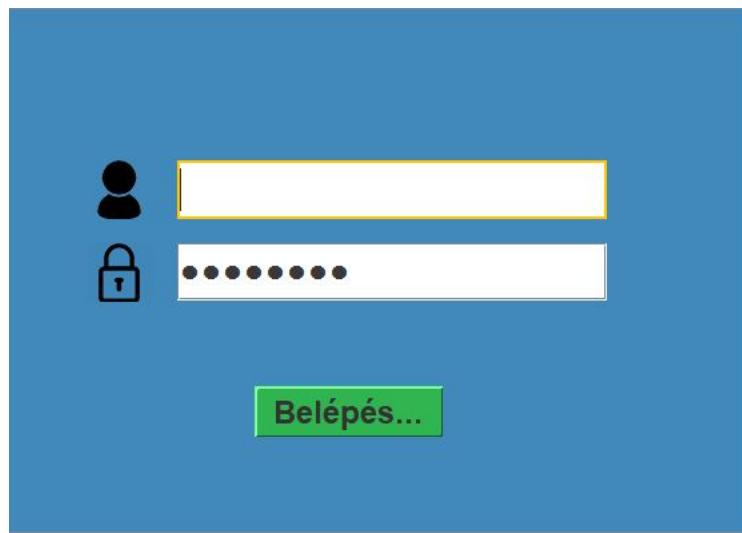
7.4 Menürendszer

A programban az alábbi menüpontok találhatóak. Mindegyik menüpontban miután a felhasználó megadta az adatokat, egy felugró ablak ugrik fel, mely jelzi, hogy az adatok feltöltése sikeres volt. Azonban mikor valamilyen módon, rosszul adott meg a felhasználó

egy bizonyos adatott, értesítés kap erről, és a program addig nem engedi tovább a felhasználót, amíg nincs helyesen kitöltve az adott mező.

7.4.1 Belépés/Kilépés

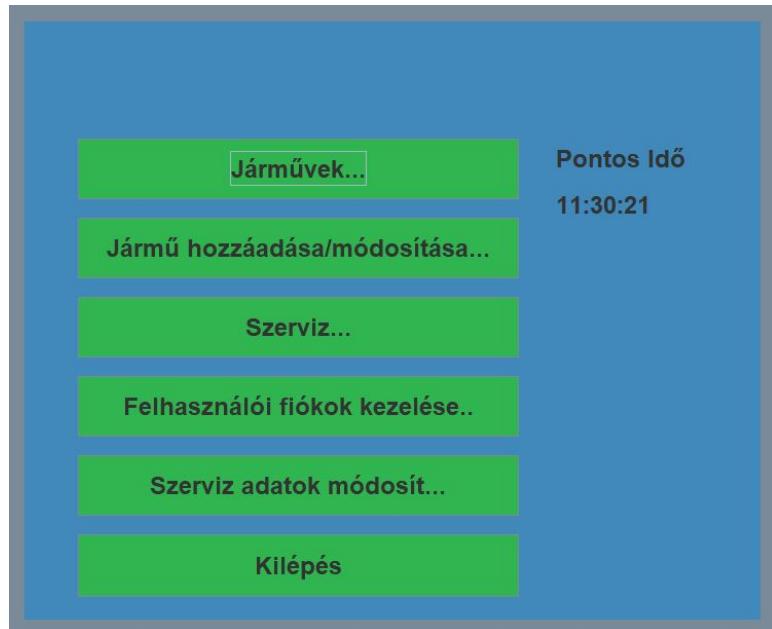
Az alkalmazás ezzel a felülettel indul el. Mikor a felhasználó sikeresen megadta a helyes felhasználónév és jelszó kombinációt akkor a program a főmenübe irányítja át. Ha a felhasználó ki szeretne lépni a programból azt a főmenüben lévő kilépés gombbal teheti meg, illetve, ha bármelyik menüben bezárja az alkalmazást, a program ki fogja léptetni.



23. ábra Belépés ablak képernyőkép

7.4.2 Főmenü

A főmenüből lehet elnavigálni az alkalmazásban található többi menüpontba.



24. ábra Főmenü képernyőkép

7.4.3 Szerviz

Ebben a menüpontban az ügyfelek, márkkák, autók, munkalapok, feladatok, szerelések, alkatrészek, és beszerzések adatait lehet feltölteni az adatbázisba, és a jelenleg magadott adatokat PDF-be lehet exportálni a beszerzés fülön található „munkalap készítése” nevű gombra kattintva. Ebben a menüben logikus sorrendben vannak megadva az almenük, ennek tudatában kell végig menni, balról jobbra.

A szerviz menü első ablakában az ügyfelek nevét, lakcímét, elérhetőségét, okmány típusát, és azonosítóját lehet megadni.

Ügyfél	Márka	Autó	Munkalap	Feladat	Szerelés	Alkatrész	Beszerzés
Név	<input type="text"/>						
Lakcím	<input type="text"/>						
Elérhetőség	<input type="text"/>						
Okmány típusa	<input type="text"/>						
Okmány azonosító	<input type="text"/>						
Felvitel							

25. ábra Ügyfél hozzáadása képernyőkép

A második ablakban a márkkákat lehet megadni, a hozzájuk tartozó márka kóddal, és elnevezéssel.

Ügyfél	Márka	Autó	Munkalap	Feladat	Szerelés	Alkatrész	Beszerzés
Márka kód	<input type="text"/>						
Elnevezés	<input type="text"/>						
Felvitel							

26. ábra Márka hozzáadása képernyőkép

Az autók almenüpontban, az autó rendszámát, a hozzá tartozó ügyfelet, az autó márkáját és típusát, a gyártási évét, színét, motorszámát, és alvázszámát lehet megadni.

The screenshot shows a user interface for registering a vehicle. At the top, there is a horizontal menu bar with several tabs: Ügyfél, Márka, Autó, Munkalap, Feladat, Szerelés, Alkatrész, and Beszerzés. The 'Autó' tab is currently selected, indicated by a blue background and white text. Below the menu, there are four rows of input fields:

- Rendszám:** A text input field containing an empty white rectangle.
- Típus:** A text input field containing an empty white rectangle.
- Ügyfél:** A dropdown menu showing the value '112511TR'.
Szín: A dropdown menu showing the value 'fehér' (white).
- Márka:** A dropdown menu showing the value 'ma/005'.
Motorszám: An empty text input field.
- Évjárat:** A dropdown menu showing the value '2020'.
Alvázszám: An empty text input field.
- Felvitel:** A large green rectangular button with the text 'Felvitel' in white.

27. ábra Autó hozzáadása képernyőkép

A negyedik fülön a munkalapot lehet létrehozni, mégpedig úgy, hogy a munkalapszám megadása után, megadjuk az adott autó rendszámát, a vállalási időt, hogy mégis mikor lesz kész az adott autó, és a hibákat, amik leírják, hogy mit kell csinálni az autón.

Ügyfél	Márka	Autó	Munkalap	Feladat	Szerelés	Alkatrész	Beszerzés
Munkalap szám <input type="text"/>							
Rendszám		ASD-123 <input type="button" value="▼"/>					
Vállalási határidő <input type="text"/>							
Hiba jegyzék <input type="text"/>							
<input type="button" value="Felvitel"/>							

28. ábra Munkalap hozzáadása képernyőkép

A feladat almenüpontban lehet megadni, a feladatokat, amiket az autókon kell csinálni. minden feladatnak van egyedi feladat azonosítója, munkadíja, típusa és elnevezése. Négy fajta típusú feladat van:

- javítás
- alkatrész csere
- felkészítés vizsgára
- vizsgáztatás

Ügyfél Márka Autó Munkalap Feladat Szerelés Alkatrész Beszerzés

Feladat

Munka díj

Típus javítás

Elnevezés

Felvitel

29. ábra Feladat hozzáadása képernyőkép

A szerelés almenüpontban a megadott munkalapszámhoz lehet szerelési feladatokat adni, az adott feladat számmal, és szerelővel. minden szerelésnek van kezdeti, és befejezési ideje, és a szereléssel eltöltött munkaórák száma.

Ügyfél Márka Autó Munkalap Feladat Szerelés Alkatrész Beszerzés

Munkalap szám mu/2020/0000

Szerelő 3

Feladat fe/000

Szerelés kezdés ideje

Szerelés befejezés ideje

Munkaóra

Felvitel

30. ábra Szerelés hozzáadása képernyőkép

Az alkatrész menüpontban új alkatrészek megadására van lehetőség. minden alkatrésznek van saját alkatrész kódja és megnevezése.

The screenshot shows a mobile application interface for managing parts. At the top, a blue header bar contains several tabs: 'Ügyfél', 'Márka', 'Autó', 'Munkalap', 'Feladat', 'Szerelés', 'Alkatrész' (which is highlighted in blue, indicating it's the active tab), and 'Beszerzés'. Below the header, there are two input fields: 'Alkatrész kód' (Part code) and 'Alkatrész megnevezése' (Part name), each followed by a white input box. At the bottom center is a green rectangular button with the text 'Felvitel' (Input) in white.

31. ábra Alkatrész hozzáadása képernyőkép

Az utolsó almenüpontban az adott munkalaphoz lehet hozzárendelni, egyedi beszerzési kóddal az alkatrészeket, és az árukat. minden alkatrészhez ki kell tölteni az megrendelési idejét.

Beszerzési kód	<input type="text"/>
Munkalap szám	mu/2020/0000 <input type="button" value="▼"/>
Alkatrész	<input type="text"/> fékbetét <input type="button" value="▼"/>
Alkatrész ára	<input type="text"/>
Megrendelés ideje	<input type="text"/>
	<input type="button" value="Felvitel"/> <input type="button" value="Munkalap készít"/>

32. ábra Beszerzés hozzáadása képernyőkép

7.4.4 Szerviz módosítása

A menüpontban az ügyfél adatait, és a beszerzés adatai lehet módosítani a 34.-ábrától a 35.- ábráig látható módon.

Ügyfél módosítása		Beszerzés módosítása
szemelyazon_okmany_szam	szemelyazon_okmany_tipus	
112511TR	Személy azonosító	Kiss I.
459632PO		Nagy
46464646	asdadasd	asdas
564678DA		Ková
<input type="button" value="Módosítás"/>		
Okmány azonosító	<input type="text"/>	
Okmány típusa	<input type="text"/>	
Név	<input type="text"/>	
Elérhetőség	<input type="text"/>	
Lakcím	<input type="text"/>	
<input type="button" value="Fómenü"/>		

33. ábra Ügyfél adatainak módosítása képernyőkép

Ügyfél módosítása		Beszerzés módosítása
besz_kod	munklap_szam	alkatresz
be/2020/0000	mu/2020/0000	al/005
be/2020/0001	mu/2020/0001	al/007
be/2020/0002	mu/2020/0003	al/006

Módosítás

Főmenü

Beszerzési kód

Munkalap szám

Alkatrész

Alkatrész ára

Megrendelés ideje

34. ábra Beszerzés adatainak módosítása képernyőkép

7.4.5 Jármű keresése

The screenshot shows a search form for vehicles. At the top are three icons: a car, a motorcycle, and a truck. Below them are two dropdown menus: 'Márka' (Brand) set to 'Mindegy' and 'Modell' (Model). Further down are three dropdown menus: 'Kivitel' (Type) set to 'Mindegy', 'Állapot' (Status) set to 'megkímélt' (inspected), and 'Üzemanyag' (Fuel) set to 'Mindegy'. There are also two more dropdown menus: 'Rendszám' (Registration number) set to 'Mindegy' and 'Évjárat(-tól -ig)' (Year range) set to 'Mindegy'. To the right of the search area are two green buttons: 'Főmenü...' (Main menu...) and 'Keresés...' (Search...).

35. ábra Járművek keresése képernyőkép

Ebben a menüpontban az adatbázisban tárolt járművek adatai lehet megtekinteni táblázat formájában.

The screenshot shows a table of vehicle data. The columns are labeled 'rendszer', 'marka', 'modell', 'tipus', and 'éve'. The data includes:

rendszer	marka	modell	tipus	éve
KDA-593	Audi	A3	Sportback	2006
MTS-124	Alfa Romeo	156	TS	2008
NOW-867	Citroen	Berlingo	Family	2007
PCU-977	BMW	330	D	2010
PYL-909	Chrysler	Voyager	CRD LX	2005
UIG-469	Honda	CBF 600	SA	2007

To the right of the table are two green buttons: 'Új jármű keresése...' (New vehicle search...) and 'Főmenü...' (Main menu...).

36. ábra Járművek találati táblázata képernyőkép

7.4.6 Jármű hozzáadása

Járműveket lehet hozzá adni a 34.-ábrától a 35.-ábráig látható módon. mindenéppen ki kell választani a feltöltendő jármű típusát, ezek után lehet megadni a jármű adatait.

Jármű hozzáadása I | Jármű hozzáadása II | Módosítás / Törlés

jármű típusa

Válasszon! ▾

Rendszám:

Márka:

Modell:

Típus:

Évjárat: 2020

Hónap: 01

Kivitel:

Állapot: normál Telephelyen van

Jármű leírása:

Jármű típushibái:

Jármű Hozzáadása...
Jármű Módosítása...
Jármű Törlése...
Képek hozzáadása...
Főmenü..

37. ábra Jármű hozzáadása első oldal képernyőkép

Jármű hozzáadása I	Jármű hozzáadása II	Módosítás / Törlés	
Üzemanyag			
Hengerürtartalom		cm ³	
Kilométeróra állás		km	
Teljesítmény		kW	Lóerő
Szín	fehér		Jármű Hozzáadása...
Sebességváltó fajtája	mechanikus		Jármű Módosítása...
Vételár		Ft	Jármű Törlése...
Saját tömeg		kg	Kép hozzáadása...
Főmenü..			

38. ábra Jármű hozzáadása második oldal képernyőkép

Jármű hozzáadása I	Jármű hozzáadása II	Módosítás / Törlés			
Személyautó ▾					
rendszer	marka	modell	tipus	evjárat	
KDA-593	Audi	A3	Sportback	2006	06
MTS-124	Alfa Romeo	156	TS	2008	03
NOW-867	Citroen	Berlingo	Family	2007	11
PCU-977	BMW	330	D	2010	03
PYL-909	Chrysler	Voyager	CRD LX	2005	12
UIG-469	Honda	CBF 600	SA	2007	03
Jármű módosítása I Jármű módosítása II Jármű módosítása III					
Rendszám		Évjárat	2020		
Márka		Hónap	01		
Modell		Kivitel	cabrio		
Típus		Állapot	normál		

39. ábra Jármű módosítása első oldal képernyőkép

Jármű hozzáadása I Jármű hozzáadása II Módosítás / Törlés					
Motorkerékpár					
rendszer	marka	modell	tipus	evjárat	hona
UWB-306	Suzuki	GSF 1200 S	Bandit	2010	04

Jármű módosítása I Jármű módosítása II Jármű módosítása III					
Üzemanyag	benzin				
Hengerűrtartalom					
Kilóméteróra állás					
Saját tömeg					
Teljesítmény					

41. ábra Jármű módosítása második oldal képernyőkép

Jármű hozzáadása I Jármű hozzáadása II Módosítás / Törlés					
Haszonjármű					
rendszer	marka	modell	tipus	evjárat	hona
MTT-344	DAF	XF	105	2010	04
NDU-405	DAF	95 XF	430	2014	08
NNY-305	DAF	CF	85.410	2017	07

Jármű módosítása I Jármű módosítása II Jármű módosítása III					
Szín	fehér	<input checked="" type="checkbox"/> Telephelyen van			
Sebességváltó fajtaja	mechanikus				
Jármű leírása:					
Jármű típushibái:					
Vételár					

40. ábra Jármű módosítása harmadik ablak képernyőkép

7.4.7 Felhasználó hozzáadása

Ebben a menüpontban új felhasználókat lehet hozzáadni a programhoz. minden felhasználóhoz tartozik egy felhasználónév és egy jelszó, melyet az ellenőrzés miatt kétszer kell meg adni. Ezenkívül meg kell adja a vezetéknévét és a keresztnévét.

Vezetéknév	<input type="text"/>
Keresztnév	<input type="text"/>
Felhasználónév	<input type="text"/>
Jelszó	<input type="text"/>
Jelszó újra	<input type="text"/>

Főmenü **Felvitel**

42. ábra Felhasználó hozzáadása képernyőkép

8. Összefoglalás

A kezdetben elgondolt módszer, hogy a járműkezelő szoftver annak érdekében készítem, hogy mélyebb és tágasabb tudást szerezzenek mind az adatbázis kezelésben, JAVA Swinges környezet, és egyéb programozási nyelvek megismerése terén, teljesen bevált. Sikerült olyan dolgokat megvalósítanom, amelyekre sose gondoltam volna, hogy menni fog –bár a kíváncsiság, és a feladat nehézségének mértéke rendkívül hajtott. Véleményem szerint, ha lett volna még pár hetem, több funkcionálitással bíró alkalmazást tudtam volna fejleszteni, de próbáltam megfelelni az elvárásoknak. A projekt nagyban fejlesztette az alap JAVA, és az adatbázissal kapcsolatos tudásomat. Legközelebb mindenkorban több osztályra szeretném szedni a nézet osztályt, mivel átláthatósági szempontból sokkal jobb lenne a programom. Ezenkívül meggondolnám, hogy melyik típusú adatbázist használnám a következő projektben. Sok hasznos előnye van a MySQL-nek a többi adatbázishoz képest, viszont sok szempontból jobbnak tartom a Microsoft által fejlesztett MSSQL-t. Mindenképpen tovább szeretnék az alkalmazással foglalkozni na jövőben, bár lehetséges, hogy pár dologhoz másképp kezdenék neki, mint ahogy a jelen helyzetben tettem, ennek ellenére sokat fejlődtem a kiválasztott programozási környezetekben. A legfontosabb, amit tanultam a szakdolgozatom írása alatt, az az, hogy legközelebb mindenkorban jobban kell beosztanom az időmet, a különböző feladat modulokat tekintve.

9. Irodalomjegyzék

1. Autó kereskedés kép

<https://cdn1.iconfinder.com/data/icons/ios-11-glyphs/30/car-512.png>

2020.01.08.

2. Motor kereskedés kép

https://cdn3.iconfinder.com/data/icons/industry-7/32/industry_motorbike-512.png

2020.01.08.

3. Kamion kereskedés kép

<https://image.flaticon.com/icons/png/512/31/31520.png>

2020.01.08.

4. Felhasználói ikon

https://www.iconexperience.com/o_collection/icons/?icon=user

2020.02.23.

5. MySQL connector jar file windows rendszerhez

<https://dev.mysql.com/downloads/connector/j/5.1.html>

2020.02.10.

6. MySQL csatlakozása JAVA nyelven

<https://www.javatpoint.com/example-to-connect-to-the-mysql-database>

2020.02.10.

7. Használt autók és egyéb eladó járművek oldala

<https://www.hasznalauto.hu/>

2020.02.16.

8. Forgalmi engedélyben szereplő színkódok leírása

<https://www.autoszektor.hu/hu/content/anyukam-erten-fogja-jam-dzsem-lekvar>

2020. 02.16.

9. Az idei tanévben Kaczur Sándor Tanár Úr által tanított tananyagok gyűjteménye

<http://kaczursandor.hu/SZKI/2019-2020-1/>

2020.02.16.

10. Szakdolgozat minták

<http://kaczursandor.hu/SZKI/szakdolgozat/>

2020.02.16.

11. Reguláris kifejezés jelszavak ellenőrzésére

<https://stackoverflow.com/questions/19605150/regex-for-password-must-contain-at-least-eight-characters-at-least-one-number-a>

2020.03.06.

12. Jelszó titkosítása

<https://stackoverflow.com/questions/33085493/how-to-hash-a-password-with-sha-512-in-java>

2020.03.07.

13. Reguláris kifejezés használata JAVA metódussal

<https://stackoverflow.com/questions/1378978/using-the-netbeans-gui-editor-how-can-i-create-a-jtextfield-or-jformattedtext-f>

2020.03.09.

14. Reguláris kifejezések ellenőrzésére szolgáló oldal

<https://regex101.com/>

2020.03.13.

15. Dátum ellenőrzése reguláris kifejezéssel

<http://blog.vadimkononov.ca/2014/09/html5-pattern-regex-date-validator-yyyy.html>

2020.03.14.

16. Dátum és idő formátum ellenőrzése reguláris kifejezéssel

<https://www.regextester.com/1966>

2020.03.14.

17. Csak számok elfogadása szöveges mezőkbe

<https://www.youtube.com/watch?v=SKXq-2o6gdc>

2020.03.19.

18. Kép átméretezéshez szükséges metódus

<https://stackoverflow.com/questions/9417356/bufferedimage-resize>

2020.03.29.

