



Vas Megyei Szakképzési Centrum
Nádasy Tamás Technikum és Kollégium

PROJEKTFELADAT

Univerzális szervizkönyv

Molnár Dániel Bence,

Tóth Erik János

**Konzulens:
Balics Gábor**

2023

Nyilatkozat

Alulírott, **Molnár Dániel Bence**, **Tóth Erik János** Szoftverfejlesztő és -tesztelő szakos hallgatók kijelentjük, hogy az *Univerzális szervizkönyv* című szakdolgozat feladat kidolgozása a saját munkánk, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtuk fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeink saját munkán, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudásunk szerint hitelesek.

Csepreg, 2023.03.10

hallgató

Csepreg, 2023.03.10

hallgató

Kivonat

Univerzális szervizkönyv

Az *Univerzális szervizkönyv* projekt célja, hogy megoldást nyújtson a járművek szervizelésének megbízható nyomon követésére. A webalkalmazás két részből áll, az első a műhelyek számára készült, és bármelyik szerviz csatlakozhat hozzá: az alvázszám alapján az alkalmazás segítségével egy adott járműhöz hozzáadhatnak új, digitális szervizbejegyzést, amit a tulajdonos egyszerre meg is tud tekinteni. Ez az alkalmazás a papíralapú munkalapot is leválthatja a műhelyekben. A második rész a járműtulajdonosoknak szól, itt bárholnan és bármikor megtekinthetik járműveik előéletét a jármű kategóriájától és márkájától függetlenül. Az alkalmazás további előnye, hogy segítséget nyújt a jármű eladásához is, mivel lehetővé teszi a megosztást, így a leendő vásárlók is láthatják a korábbi szervizbejegyzéseket.

Abstract

Universal Servicebook

The aim of our project task is to provide a solution to the reliable tracking of vehicle services. Our web application consists of two parts, the first one is meant for repair garages and can be joined by any of them: based on the chassis number, the application can be used to add a new digital service record relating to a given vehicle, which can be viewed by the owner immediately. This app can replace the paper-based worksheet in repair garages. The second part is for vehicle owners, here they can see and follow the history of their vehicles anytime, regardless of the category and model of the vehicle. A further advantage of the application is that it provides help when selling a vehicle by allowing its sharing so that potential buyers can see all previous service records.

Tartalomjegyzék

1.	BEVEZETÉS	7
2.	FEJLESZTŐI DOKUMENTÁCIÓ	8
2.1.	Használt programozási nyelvek és technológiák.....	8
2.1.1.	JavaScript.....	8
2.1.2.	React.....	8
2.1.3.	Material UI.....	9
2.1.4.	Node.js	9
2.1.5.	Express.js.....	10
2.1.6.	Jest.....	10
2.1.7.	GitHub	10
2.2.	Fejlesztői környezetek	11
2.2.1.	Visual Studio Code.....	11
2.2.2.	MongoDB Compass	12
2.2.3.	Postman.....	12
2.2.4.	GitKraken	13
2.3.	Az adatbázis.....	14
2.3.1.	Tervezési szint.....	14
2.3.2.	Fizikai szint (implementálás).....	16
2.3.2.1.	Jármű paraméterek.....	17
2.3.2.2.	Jármű	17
2.3.2.3.	Felhasználó	18
2.3.2.4.	Műhely.....	19
2.3.2.5.	Képek	20
2.3.2.6.	Legutóbbi aktivitások	20
2.3.2.7.	Log.....	21
2.4.	Teszt dokumentáció	22
2.4.1.	Modul (unit) tesztek	22
2.4.2.	Stresszteszt és terheléses teszt	23
2.4.3.	Alfa teszt.....	23
2.4.4.	Béta teszt.....	23

2.5.	Fejlesztési lehetőségek.....	25
2.5.1.	Üzenetküldés az oldalon.....	25
2.5.2.	Értesítés.....	25
2.5.3.	Értékelés, műhelyajánlás.....	25
2.5.4.	Többnyelvűség.....	25
2.5.5.	Natív alkalmazás.....	25
3.	FELHASZNÁLÓI DOKUMENTÁCIÓ	26
3.1.	Rendszerkövetelmények	26
3.2.	Jogosultsági szintek.....	26
3.3.	Felhasználói felület	26
3.3.1.	Regisztráció.....	27
3.3.2.	Fiókaktiválás és bejelentkezés.....	28
3.3.3.	Főoldal.....	29
3.3.4.	Járműveim	30
3.3.5.	Egy jármű megtekintése	33
3.3.6.	Beállítások	36
3.3.7.	Oldalkezelői adminisztráció	37
3.3.8.	Műhely adminisztráció	38
3.3.9.	Műhely	39
3.3.10.	Elfelejtett jelszó.....	40
4.	ÖSSZEFOGLALÁS	41
5.	IRODALOMJEGYZÉK.....	42
6.	ÁBRAJEGYZÉK.....	44
7.	MELLÉKLETEK.....	46

1. Bevezetés

Az eladó járműveknél egyre inkább fontossá vált, hogy van-e szervizelőéletük, ugyanis a vásárlók jobban bíznak azokban a járművekben, amikről tudják, hogy rendszeresen szervizelték őket. De ez se olyan megbízható, mivel a régi járműveknél még papíralapú szervizkönyv létezett, és abba bárki beleírhatott, nem biztos, hogy a reális adatot tartalmazta. Az újonnan vásárolt járművekhez mára már mindenhol digitális szervizkönyvet biztosítanak, amibe csak az adott márka specifikus szerelője tud beleírni.

Márkaspecifikus digitális szervizkönyvek már léteznek, de nem lehet megtekinteni a tárolt szervizbejegyzéseket, csak akkor, ha a tulajdonos bemegy a kereskedésbe, és lekéri az adatokat. A megfelelő összeg kifizetése után kinyomtatva megkaphatja a jármű előéletét.

Az általunk létrehozott *Univerzális szervizkönyv* egy webalkalmazás, amit két részre lehet bontani. Az alkalmazás első része a műhelyeknek biztosít egy felületet, amihez bármilyen szerviz csatlakozhat. Ezen a felületen egy alvázszám alapján a járművek adatait a műhelyekben elérhetik, és ehhez hozzáadhatnak új, digitális szervizbejegyzést, amit a jármű tulajdonosa egyszerre megtekinthet. Ezen felül ezzel a módszerrel leválthatják a papíralapú munkalapot. A második rész a járműtulajdonosoknak szól, ide bárki regisztrálhat. A tulajdonosok tárolhatják a járműveiket és ezek adatait járműkategóriától és márkától függetlenül. Bárhol és bármikor megtekinthetik a korábbi szervizbejegyzéseket. Ezen kívül az alkalmazás hozzásegít a jármű eladásához, mivel a tulajdonos megoszthatja másokkal is a jármű adatait, és a tulajdonosváltás rögzítését is meg tudja valósítani.

2. Fejlesztői dokumentáció

2.1. Használt programozási nyelvek és technológiák

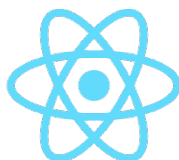
2.1.1. JavaScript¹



1. ábra:
JavaScript logó

A JavaScript egy kliens- és szerveroldali programozási nyelv. A kliensoldali JavaScript arra szolgál, hogy interaktív és dinamikus felhasználói élményeket biztosítson a weboldalakon. A kliensoldali JavaScript lehetővé teszi a weboldalak tartalmának dinamikus frissítését, animációkat és átmeneteket, valamint valós idejű adatkommunikációt a szerverrel. A kliensoldali JavaScript gyakran használ kiegészítő könyvtárakat és keretrendszereket, mint például a React, Angular vagy a Vue.js, amelyek segítségével a fejlesztők egyszerűbbé tehetik a webalkalmazások fejlesztését. A JavaScript nem csak a kliensoldalon, hanem a szerveroldalon is használható, a Node.js környezetben. A Node.js egy nyílt forráskódú, cross-platform JavaScript futtatókörnyezet, amely lehetővé teszi a fejlesztők számára, hogy JavaScript-ben írják a szerveroldali alkalmazásokat is. A kliensoldali JavaScript és a szerveroldali JavaScript együtt alkotják a modern webalkalmazások két oldalát, és lehetővé teszik az alkalmazások teljes funkcionalitását.

2.1.2. React²



2. ábra: React
logó

A React (vagy ReactJS) egy nyílt forráskódú JavaScript keretrendszer, amely lehetővé teszi felhasználói felületek készítését webalkalmazásokhoz. A React alapvetően egy komponens alapú keretrendszer, ami azt jelenti, hogy az alkalmazásokat egymástól függetlenül épített, újra felhasználható komponensekre bontja. Ezek a komponensek tartalmazzák a logikát és az adatok megjelenítését is, és lehetővé teszik a fejlesztők számára, hogy az alkalmazásukat dinamikusan kezeljék.

A React a virtuális DOM (Document Object Model) koncepcióját használja, amely lehetővé teszi a hatékonyabb és gyorsabb alkalmazásfrissítést a valódi DOM-hoz képest. Ez azt jelenti, hogy az alkalmazás állapotának megváltozása esetén a virtuális DOM-nál történik

¹ https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript [2022.10.13]

² https://www.w3schools.com/whatis/whatis_react.asp [2022.10.13]

meg az állapotfrissítés, majd csak azokat a változásokat jeleníti meg a valódi DOM-ban, amelyek valóban változtak.

2.1.3. Material UI³



3. ábra: Material UI logó

A Material UI egy nyílt forráskódú React komponenskönyvtár, amely segít az alkalmazás felületeinek gyors és egyszerű elkészítésében. A Material UI az anyagtervezési nyelvet követi, amely a Google által fejlesztett dizájnminták és stílusok összessége, amelyeket az Android és más Google termékek felhasználnak. A Material UI készletében számos alapvető UI elem található, mint például gombok, input mezők, lista elemek, kártyák és sok más. A Material UI lehetővé teszi a fejlesztők számára, hogy a készletekben található előre elkészített komponenseket felhasználják, vagy személyre szabott komponenseket hozzanak létre. Mi azért használtuk a MUI-t, mert egységes megjelenést ad alkalmazásunknak és lerövidítette a fejlesztési időt.

2.1.4. Node.js⁴



4. ábra: NodeJS logó

A Node.js egy nyílt forráskódú, backend oldali JavaScript futtatókörnyezet. Használatával JavaScript alkalmazásokat és kódokat futtathatunk böngészőkön kívül, akár saját számítógépünkön a Google által fejlesztett V8 JavaScript motor segítségével. Mivel támogatja az összes főbb webes protokollt, beleértve a HTTP-t, a HTTPS-t és a WebSockets-t, így könnyedén használhatjuk webkiszolgálók fejlesztésére. A segítségével könnyedén készíthetünk webszolgáltatásokat, chat alkalmazásokat, játékokat és más valós idejű alkalmazásokat. Az összes főbb platformot támogatja, beleértve a Windows-t, Linux-t és a Mac OS X-t is. Rendelkezik egy fejlett modulrendszerrel, segítségével könnyedén hozzáadhatunk új funkciókat az alkalmazáshoz. Azért választottuk a Node.js-t, mert egy nyílt forráskódú platform, és nagyon aktív fejlesztői közössége van, ami jelentősen növeli a platform életképességét és gyors fejlődését. Számos népszerű keretrendszere van, mint például a mi esetünkben az Express.

³ <https://mui.com/material-ui/getting-started/overview/> [2022.11.18]

⁴ <https://nodejs.org/en/blog/announcements/foundation-v4-announce/> [2022.11.23]

2.1.5. ExpressJs⁵



5. ábra: ExpressJS logó

A ExpressJs egy népszerű és könnyen használható web backend keretrendszer a Node.js-hez. Segítségével könnyedén készíthetünk webszolgáltatásokat, webalkalmazásokat és API-kat, alapértelmezetten támogatja a HTTP protokollt és a routing-ot, ami segít a kérések kezelésben és a válaszadásban. Támogatja a middleware-eket, amelyek segítenek a kérés- és válaszfeldolgozásban. Lehetővé teszi a különböző domaineokról érkező kérések kezelését a CORS támogatásával. Alapértelmezetten támogatja a biztonsági funkciókat, mint például a hitelesítést és a jogosultság kezelését. Ezen felül számos kiegészítő modul érhető el az npm segítségével, ami egy világ nagyságú JavaScript könyvtárnyilvántartó. Azért is választottuk ezt a keretrendszert, mert nagyon sok információt találhatunk az interneten hozzá.

2.1.6. Jest⁶



6. ábra: Jest logó

A Jest egy nyílt forráskódú JavaScript tesztelő keretrendszer. A Jest célja, hogy egyszerűvé tegye a JavaScript alkalmazások tesztelését, és támogassa a különböző típusú teszteket, mint például a modulteszteket, az integrációs teszteket és az UI teszteket. A Jest magában foglalja az olyan funkciókat, mint az automatikus teszt futtatását, az összefoglaló jelentéseket, a snapshot teszteket és az aszinkron teszteket. A Jest a Node.js és a React közösségben is népszerű, és gyakran használják a modern JavaScript alkalmazások teszteléséhez.

2.1.7. GitHub⁷



7. ábra: GitHub logó

A GitHub egy webes alapú verziókövető és együttműködési platform szoftverfejlesztők számára. A GitHub lehetővé teszi a fejlesztőknek, hogy közösen dolgozzanak egy projekt fejlesztésén, kövessék a kód verzióját, ellenőrizzék és felügyeljék a változtatásokat, dokumentációkat készítsenek, hibákat javítsanak.

⁵ <https://expressjs.com/> [2022.12.02]

⁶ <https://jestjs.io/> [2022.12.02]

⁷ <https://kinsta.com/knowledgebase/what-is-github/> [2022.12.02]

A GitHub felhasználói előnyöket élveznek a szolgáltatás nyújtotta könnyű hozzáféréshez az együttműködési platformhoz, lehetőségük van nyomon követni a fejlesztések történetét, visszatekinteni a korábbi változatokra, hozzászólni a kóddal kapcsolatban, illetve javaslatokat, módosításokat tenni az adott projektben. A GitHub lehetővé teszi a közösségi munkát, ami segíti a fejlesztők kapcsolódását, az együttműködést és tudásuk megosztását a szoftverfejlesztés területén.

2.2. Fejlesztői környezetek

2.2.1. Visual Studio Code⁸



8. ábra: Visual Studio Code logó

A Visual Studio Code (röviden VS Code) egy nyílt forráskódú, cross-platform szövegszerkesztő és fejlesztői környezet, amelyet a Microsoft fejlesztett ki. Az alkalmazás ingyenesen letölthető és használható Windows, macOS és Linux operációs rendszereken.

A VS Code több programozási nyelvet támogat, és számos funkcióval rendelkezik, amelyek megkönnyítik a fejlesztők számára a kódírási folyamatát. Ezek közé tartozik például az intelligens kódkiegészítés, a szintaxis kiemelés, a debuggolás, a Git integráció, a szerveroldali fejlesztést támogató bővítmények.

A VS Code rendkívül testre szabható, és lehetőséget biztosít a felhasználóknak, hogy testre szabják az alkalmazást a saját igényeik szerint. Az alkalmazás maga is bővíthető, és lehetőséget kínál a felhasználók számára, hogy telepítsenek különböző bővítményeket és kiterjesztéseket, amelyek további funkciókat adnak az alkalmazáshoz.

Azért választottuk, mert az alkalmazás számos szolgáltatást kínál a fejlesztők számára, és lehetővé teszi a hatékony és produktív kódolást az egyszerű kódtól a nagyobb projektekig.

⁸ <https://code.visualstudio.com/docs/editor/whyvscode> [2022.12.20]

2.2.2. MongoDB Compass⁹



9. ábra:
MongoDB
Compass logó

A MongoDB Compass egy ingyenes grafikus felhasználói felület (GUI) a MongoDB adatbázis-kezelő rendszer számára. A MongoDB Compass lehetővé teszi a felhasználók számára, hogy könnyedén és hatékonyan vizsgálják és kezeljék a MongoDB adatbázisokat.

A Compass egy intuitív felhasználói felületet kínál, amely lehetővé teszi a fejlesztők számára, hogy gyorsan áttekintsék az adatokat, valamint hozzáadjanak, módosítsanak vagy töröljenek dokumentumokat az adatbázisban. A felhasználók könnyen navigálhatnak az adatbázisok és a kollekciók között, és használhatják a beépített keresési és szűrési funkciókat az adatok megtalálásához.

2.2.3. Postman¹⁰



10. ábra: Postman
logó

A Postman egy szoftver, amely lehetővé teszi az API-k (Application Programming Interface) tesztelését, dokumentálását és fejlesztését. Az API-k olyan programozási felületek, amelyek könnyebb kommunikációt biztosítanak az alkalmazások és a szolgáltatások között

A Postman egy grafikus felhasználói felületet (GUI) biztosít, segítségével a fejlesztők tesztkérdéseket küldhetnek az API-k felé, és ellenőrizhetik a válaszokat. A Postman a HTTP, HTTPS és más protokollok használatára is alkalmas, és támogatja a REST (Representational State Transfer) API-kat és más elterjedt API típusokat.

A Postman funkciói közé tartozik a kérések automatizálása, a tesztek és adatok mentése, a dokumentáció és az API-keresések megosztása a csapatokkal.

⁹ <https://www.mongodb.com/products/compass> [2022.12.20]

¹⁰ <https://www.javatpoint.com/postman> [2023.01.09]

2.2.4. GitKraken¹¹



11. ábra: GitKraken
logó

A GitKraken egy grafikus felhasználói felületet kínál, amely könnyen használható és intuitív, így a fejlesztők könnyen megérthetik és kezelhetik a Git-ben tárolt adatokat és műveleteket.

A GitKraken funkciói közé tartozik a Git-kereső, a Git flow támogatása, a gitignore fájlok kezelése, a pull és push műveletek, a tag-ek kezelése, a hibakereső eszközök, a történeti nézetek, a diff-ek és a merge műveletek kezelése. A GitKraken számos szolgáltatással működik együtt, például a GitHub, a Bitbucket és a GitLab azért, hogy a fejlesztők könnyen együtt tudjanak dolgozni más fejlesztőkkel és csapatokkal.

¹¹ <https://dev.to/iphytech/gitkraken-what-is-it-and-common-actions-5531> [2023.01.09]

2.3. Az adatbázis

Az *Univerzális szervizkönyv* projekt lényege a skálázhatóság megvalósítása, ami egy jól megtervezett adatbázissal kezdődik. Az adatbázis készítésénél a skálázhatóságot és dinamikus adatbevitelt figyelembe véve egy NoSQL¹²-es megoldást választottunk, ami a MongoDB lett. A MongoDB egy NoSQL alapú adatbázis, ami azt jelenti, hogy „nem SQL”, de gyakran hívják „nem csak SQL” -nek. Ezek az adatbázisfajták gyorsan változó, strukturálatlan adatok nagy tömegének kezelésére alkalmasak a sorokat és oszlopokat kezelő relációs adatbázistól eltérő módon. A MongoDB tárolási formája a kulcs-érték párok. Ennek a lényege, hogy van egy kulcs, amivel elérjük az adott értéket, ezen felül könnyű kezelni bármilyen JavaScript projektben, mert ez egyfajta JSON formátum. A NoSQL-nek még van ennél több típusú tárolási formája, mint például dokumentumalapú, oszlopalapú és gráfolapú. A MongoDB-ben nem táblának hívják az adatstruktúra felépítését, hanem collection-nek, és az alapértelmezett elsődleges kulcs értéke nem szám, hanem úgynevezett ObjectId¹³, ami 24 karakterből áll.

2.3.1. Tervezési szint

A tervezésnél elsősorban a skálázhatóságot vettük figyelembe és azt, hogy a soft delete-t mindenhol meg lehessen valósítani.

Először is a felhasználók eltárolására van szükségünk, mivel nekik vannak járműveik, amiket tárolhat a rendszer. A felhasználóknál szükségünk van az e-mail-címre, jelszóra, vezeték- és keresztnemekre, ezek mellett egy jogosultságot tároló mezőre. A felhasználó jelszavát az npm bcrypt nevezetű csomagja végzi, ez egy titkosítási algoritmus, amely egy stringet konvertál egy véletlenszerű és irreverzibilis karakterláncra. Az irreverzibilis jelentése, hogy a karakterlánc nem fordítható vissza az eredeti stringre. Ezen kívül használja a véletlenszerű szóást, ami véletlenszerű adatokat fűz hozzá a jelszóhoz mielőtt elvégezné a hash függvényt. Telefonszámot és lakhelyet is tárolunk, de ezt a felhasználónak nem kötelező megadni.

A regisztráció során a megadott e-mail-címre egy megerősítő kódot szeretnénk küldeni,

¹² <https://azure.microsoft.com/hu-hu/resources/cloud-computing-dictionary/what-is-nosql-database> [2023.01.11]

¹³ <https://desktop.arcgis.com/en/arcmap/latest/manage-data/using-sql-with-gdbs/object-id.htm> [2023.01.11]

aminek van egy lejárat dátuma, ezért ezt egy külön collectionban tároljuk. Ugyanebben a gyűjteményben egy elfelejtett jelszó kérésnél generált kódot is ebben szeretnénk tárolni, ezért kell egy kategória attribútum.

Az alkalmazás funkciói a járművekre épülnek. A járművek tárolására tervezett adatstruktúra a tervezés során többször változott, de alapszinten a jármű márkáját, modelljét, kiviteli dizájnját, a hajtás típusát, üzemanyag fajtáját és váltótípusát tároljuk. Ezeken felül a jármű alvázszámát, gyártási évét, saját tömegét, össztömegét, hengerűrtartalmát, teljesítményét lóerőben, okmányok jellegét, a jármű rögzítésekor a jelenlegi futásteljesítményét. Ha a jármű rendelkezik rendszámmal vagy műszaki érvényességgel, akkor tudjuk azt is tárolni. A járműről képeket is tárolunk: egy előnézeti képet és maximum 10 darab galériaképet.

A járművek hozzáadásának könnyítésére 7 darab collection-t terveztünk. A legfőbb a jármű-kategória vagy nevezhetjük járműtípusnak is. Ettől függ 4 darab gyűjtemény: a járműmárka, hajtástípus, kiviteli dizájn és a sebességváltó típusa. Ezen felül a járműmárka a járműmodellel van kapcsolatban. Az üzemanyag collection független a járműkategóriától.

A szerelőműhelyek tárolásához szükségünk van a műhely nevére, országára, településére, címére, a tulajdonosára, aki már előtte regisztrált a rendszerbe, és ha vannak alkalmazottak, a tárolásukra alkalmas mezőre. Ezen felül a műhely elérhetőségéhez telefonszámot és e-mailt tudunk tárolni, de ez nem fontos.

A szervizbejegyzés tárolásánál fontos, hogy melyik járművet, melyik szerviz és melyik szerelő szerelte. Ezen kívül legyen egy leírást, a futásteljesítményt tartalmazó és - ha a szerelő készítette a szerelésről fotókat - a képeknek egy mező.

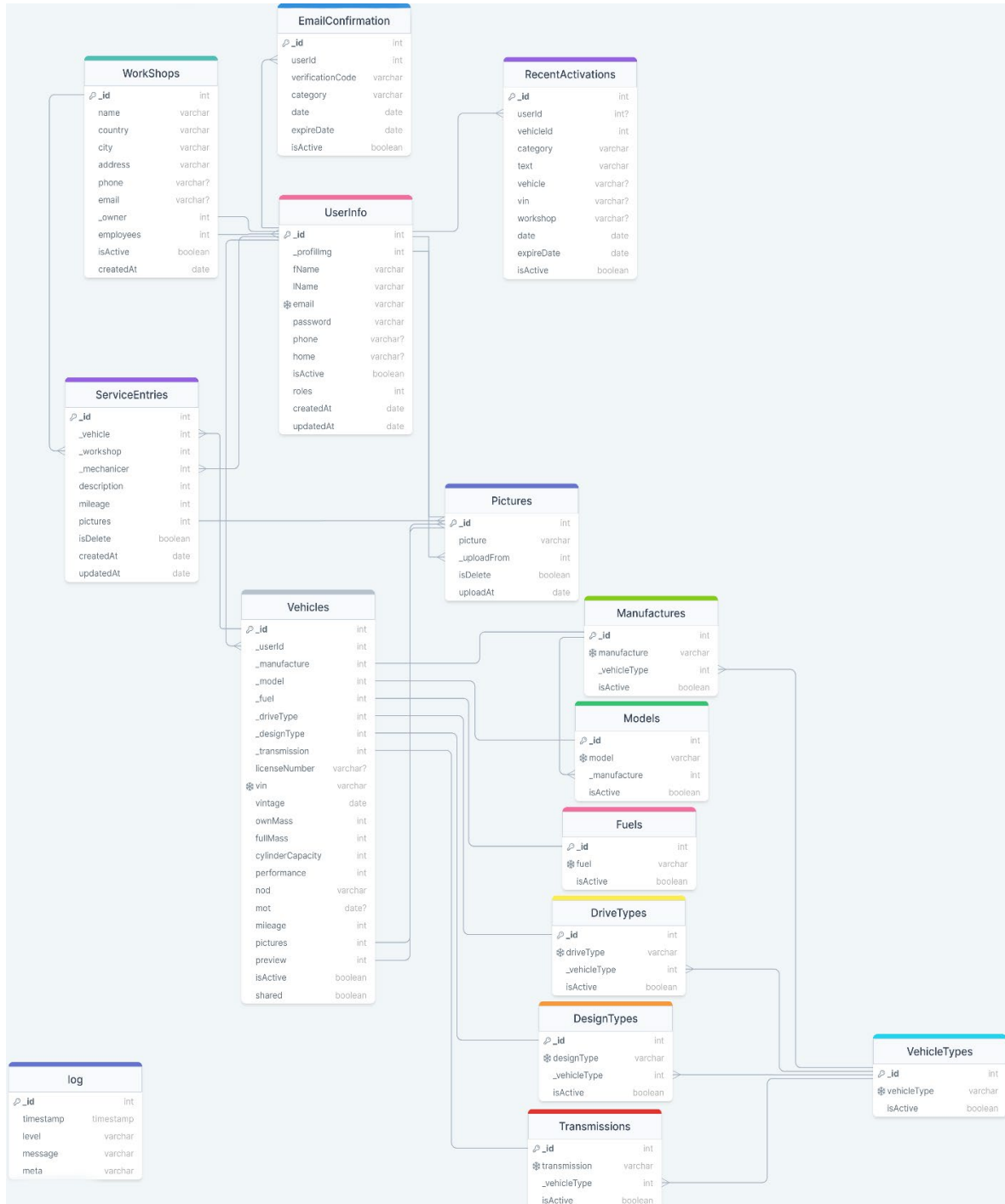
Az automatizált értesítés küldése érdekében szükségünk van egy gyűjteményre, amiben a felhasználó azonosítóját (akihez tartozik az értesítés), egy értesítési szöveget és a lejárat dátumát tároljuk. Szükségünk volt egy kategóriamezőre, hogy tudjuk kategorizálni az értesítéseket. Emellett, ha járműhöz tartozik: az alvázszámot, a jármű azonosítóját és ha van műhely, annak a nevét tároljuk.

A képek tárolásánál egy különálló gyűjteményre volt szükségünk, amiben a kép elérési útvonalát, a feltöltő felhasználó azonosítóját és az időpontot tároljuk.

A loggoláshoz a collection-t a winston nevű npm csomag fogja kialakítani.

2.3.2. Fizikai szint (implementálás)

Az előző részben leírt elvárásokat implementáltuk gyűjteményekbe. A legtöbb gyűjtemény között kapcsolat van. A kapcsolatoknál előfordul egy az egyhez, több az egyhez és egy a többhöz kapcsolat.



12. ábra: Adatbázis-modell

2.3.2.1. Jármű paraméterek

A 15 gyűjteményből 7 collection felel a járművek paramétereinek eltárolásáért: jármű kategória, márka, modell, üzemanyagfajta, hajtástípus, kivitel és a sebességváltó-típusok. Ezek a gyűjtemények egymáshoz hasonlóan épülnek fel. Minden collection-ben megtalálható az isActive mező, ami egy Boolean típusú változó igaz-hamis érték tárolására. Ezzel a mezővel tudunk inaktívvá tenni egyes paramétereket. A legtöbb paraméter függ a járműtípustól, mint például a márka, hajtástípus, itt egy a többhöz kapcsolat van.

Collection:

- VehicleTypes: Járműkategóriák
- Manufactures: Járműkategóriától függő járműmárkák
- Models: Járműmárkától függő járműmodellek
- Fules: Üzemanyagfajták, beleértve az elektromos típust is
- DriveTypes: Járműkategóriától függő hajtástípusok
- DesignTypes: Járműkategóriától függő kivitelek (tervezési stílusok)
- Transmissions: Járműkategóriától függő sebességváltó-típusok

2.3.2.2. Jármű

A járműveket a Vehicles-séma szerint tároljuk.

Vehicles attribútumai:

Kulcs	Leírás
_id	elsődleges kulcs, egyedi azonosító
_userId	idegen kulcs, a járműtulajdonos felhasználó azonosítója
_manufacture	idegen kulcs, a jármű márkájának azonosítója
_model	idegen kulcs, a jármű modelljének azonosítója
_fuel	idegen kulcs, a jármű üzemanyagfajtájának az azonosítója
_driveType	idegen kulcs, a járműhajtás-típus azonosítója
_designType	idegen kulcs, kivitel azonosítója
_transmission	idegen kulcs, sebességváltó azonosítója
licenseNumber	rendszám tárolása, maximum 10 karakter, nem kötelező
vin	jármű alvázszáma, kötelező adat, mert ennek egyedinek kell lennie
vintage	jármű gyártási éve

ownMass	jármű saját tömege
fullMass	jármű teljes tömege
cylinderCapacity	ha belsőégésű motorral rendelkezik a jármű, akkor a hengerűrtartalom mennyisége cm ³ mértékegységben
performance	jármű teljesítménye lóerő mértékegységben
mileage	jármű futásteljesítménye kilométer mértékegységben
mot	jármű műszaki érvényességének lejárat dátuma, nem kötelező
nod	járműokmányok jellege
pictures	idegen kulcs, a járműgalériába feltöltött képeknek az azonosítója
preview	idegen kulcs, jármű előképének feltöltött azonosítója
isActive	elérhető-e a jármű
shared	publikus-e a jármű

2.3.2.3. Felhasználó

A felhasználói adatokat a UserInfo collectionban tároljuk.

UserInfo attribútumai:

Kulcs	Leírás
_id	elsődleges kulcs, egyedi azonosító
fName	keresztnév
lName	vezetéknév
email	e-mail-cím, egyedinek kell lennie, nem lehet kettő ugyanolyan
password	jelszó a NodeJS bcrypt csomaggal titkosítva
phone	telefonszám, nem kötelező
home	település, ahol él a felhasználó, nem kötelező
isActive	fiók aktiválása
roles	felhasználói jogosultságok
createdAt	felhasználó létrehozásának dátuma
updatedAt	felhasználó legutolsó módosítási időpontja
_profileImg	felhasználó profilképe, nem kötelező

A felhasználó regisztrálásánál vagy elfelejtett jelszó kérésénél egy kódot generálunk, amit az EmailConfirmation collectionban tárolunk.

EmailConfirmation attribútumai:

Kulcs	Leírás
_id	elsődleges kulcs, egyedi azonosító
verificationCode	a 20 karakterből álló generált kód
userId	idegen kulcs, a felhasználó azonosítója
expireDate	a megerősítő kód lejáratási dátuma
date	a megerősítő kód létrehozási dátuma
category	a megerősítő kód kategóriája, 'email' vagy 'password'
isActive	aktív-e még a megerősítő kód

2.3.2.4. Műhely

A műhely rész két collectionból áll. A Workshops gyűjteményben egy műhelynek az alapadatai vannak.

Workshops attribútumai:

Kulcs	Leírás
_id	elsődleges kulcs, egyedi azonosító
name	műhely neve
country	műhely országa
city	műhely települése
address	műhely címe
phone	műhely telefonszáma, nem kötelező
email	műhely e-mail-címe, nem kötelező
_owner	idegen kulcs, a műhely tulajdonosának felhasználó azonosítója
employees	alkalmazottak azonosítói
isActive	elérhető-e még a műhely
createdAt	műhely létrehozásának dátuma

A szervizbejegyzéseket a ServiceEntries collectionba tároljuk.

ServiceEntries attribútumai:

Kulcs	Leírás
_id	elsődleges kulcs
_vehicle	idegen kulcs, a jármű azonosítója
_workshop	idegen kulcs, a műhely azonosítója
_mechanicer	idegen kulcs, a szerelő azonosítója
description	jármű szervizelésének a leírása
mileage	jármű futásteljesítménye szereléskor
pictures	idegen kulcs, a szerelésről készített képek azonosítója, nem kötelező
createdAt	szervizbejegyzés létrehozásának dátuma
isDelete	törölt-e a szervizbejegyzés
updatedAt	a szervizbejegyzés módosításának dátuma

2.3.2.5. Képek

A képeket, amit feltölthet a felhasználó, fájlokban tároljuk, de az elérési útvonalat a Pictures collectionban.

Pictures attribútumai:

Kulcs	Leírás
_id	elsődleges kulcs, egyedi azonosító
picture	kép elérési útvonala vagy elérési útvonalak elválasztva @[kukac] szimbólummal
_uploadFrom	idegen kulcs, a felhasználó azonosítója, aki feltöltötte a képet
isDelete	törölt-e a kép
uploadAt	kép feltöltési időpontja

2.3.2.6. Legutóbbi aktivitások

A legutóbbi aktivitásokat és a szervizinformációkat a főoldalon automatikusan generálja a rendszer, amit a RecentActivations collectionban tárolunk.

RecentActivations attribútumai:

Kulcs	Leírás
_id	elsődleges kulcs
userId	idegen kulcs, a felhasználó azonosítója, akihez tartozik az aktivitás
vehicleId	idegen kulcs, a jármű, amihez tartozik az aktivitás, nem kötelező
category	kategória, hogy milyen fajta az aktivitás
expireDate	dátum, amely után eltűnik az aktivitás a főoldalról
date	aktivitás létrehozásának dátuma
isActive	elérhető-e a legutóbbi aktivitás
text	szöveg, amit kiír szervizinformációként, nem kötelező
vehicle	jármű márkája és modellje, nem kötelező
vin	jármű alvázszáma
workshop	műhely neve a szervizinformáció kiíráshoz, nem kötelező

2.3.2.7. Log

A loggolást a winston npm csomag végzi. A hozzá tartozó collection-t ez a csomag generálja le, de létezik egy alapvető sémája.

log attribútumai:

Kulcs	Leírás
_id	elsődleges kulcs
timestamp	naplózási adat létrehozásának dátuma
level	naplózás szintje
message	naplózási üzenet
meta	loghoz fűzött adatok, amik dinamikusan változhatnak

2.4. Tesztdokumentáció

A tesztelés során csináltunk modul (unit-), stressz-, illetve felhasználók által végzett teszteket. A modulteszteket a Jest JavaScript tesztkeretrendszerével írtuk. A stressztesztet a Postman API végpont tesztelőjével végeztük, és a felhasználói teszteket, az úgynevezett béta teszteket, ahol a felhasználók a fejlesztőktől (tőlünk) függetlenek voltak, egy 10 fős társaság végezte el. Ezen felül az alfa tesztelés is megtörtént, ahol mi ketten teszteltük az alkalmazást. A webalkalmazásunk a DigitalOcean webszerver szolgáltatónál egy szerveren fut.

Szerver adatai:

- Processzor: 2 x Intel Xeon Processor E5-2650 v4
- Memória: 2 GB
- Háttértár: 25 GB Intel SSD
- Operációs rendszer: Debian 11

2.4.1. Modul (unit) tesztek

A modul, más néven unit-tesztek írásához a Jest keretrendszert alkalmaztuk. A tesztelés közben nem az éles adatbázist használtuk, hanem erre a célra egy memóriában, vagyis csak a tesztelési időszakban létező adatbázist, ami a tesztesetek lefutása során törlődött. A memória-adatbázishoz a MongoDB-Memory-Server csomagot használtuk. Az összes backend API végpont kontrolleréhez írtunk teszteket, amelyek majdnem az összes lehetséges esetet lefedik a coverage-ben. A teszteseteknél a véletlenszerű adatbázis-kapcsolat megszakadást nem tudtuk tesztelni, de az éles rendszerben nem okoz problémát, mert a hibalehetőséget lekezeljük.

Fájl	Statements	Branches	Functions	Lines
AuthController.js	93.94%	93.33%	100%	93.39%
RecentActivationController.js	100%	100%	100%	100%
UserController.js	93.04%	95%	80%	92.98%
VehicleController.js	93.57%	90.42%	100%	93.52%
VehicleParameterController.js	95.55%	100%	92.85%	95.55%
WorkshopController.js	93.30%	80.95%	94.44%	93.21%
router.js	100%	100%	100%	100%

2.4.2. Stresszteszt és terheléses teszt

Fejlesztés közben a végpontokat a Postman segítségével teszteltük. Ennek az alkalmazásnak az egyik funkciója a stressztesztelés. A tesztelést több végponton mértük le, és ezek végeredményét vettük átlagul. A stresszteszt végén azt tapasztaltuk, hogy nem lenne probléma a hirtelen fellépő, nagy mennyiségű terhelés, és nem omlott össze a rendszer, de ha összeomlott volna, a pm2 futtatókörnyezet egyszerre újraindította volna. A terheléses teszteknel nagyban függ a szerver paramétereitől az eredmény. Nálunk egy kis szerver futtatja az oldalt. A backend részt a pm2 környezet futtatja, ami alapján láttuk a processzor- és memória-terheléseket. A Loadster segítségével csináltunk terheléses tesztet, ahol először 50 robottal (bot) próbálkoztunk, és a szerver bírta a terhelést, ami annyit takar, hogy a backend rész a processzornak a 60-70%-át használta ki, a maradék a frontend kiszolgálás volt. Ezután kipróbáltuk 150 és 300 felhasználóval, de a szerver ezt már nem bírta, a processzor a backend oldalon maximum kihasználtságon volt, a frontendre alig jutott teljesítmény. Mind a három esetben a memóriahasználat 150 és 200 MB között mozgott, így azt nem olyan arányban kell bővíteni, mint a processzort. A végeredményekből arra jutottunk, hogy ezen a szerveren egyidejűleg maximum 50 fő tud lekérdezéseket csinálni.

2.4.3. Alfa teszt

Az alfa tesztet mi ketten végeztük el, ahol az összes funkciót kipróbáltuk. Néhány validációs hibát találtunk a frontenden és a backenden is, amit javítottunk, majd újból teszteltünk. A mi tesztelésünk után raktuk át a státuszt béta verzióba.

2.4.4. Béta teszt

Az oldalt teszteltük ötfajta böngészővel:

- Google Chrome
- Brave
- Microsoft Edge
- Safari
- Firefox

A felület tesztelése során a böngészők között nem volt számottevő különbség. A felhasználói felületet többféle eszközön, többféle képernyőméretben teszteltük. Az oldal támogatja a portré és a fektetett nézetet is.

Tesztelt képernyőméretek:

Eszköz	Képernyőméret
iPhone 5s/SE	320 x 640 pixel
iPhone XR	1792 x 828 pixel
Xiaomi 11T	1080 x 2400 pixel
iPad Pro 11	1194 × 834 pixel
MacBook Air 2017	1440 x 900 pixel
HP ProDisplay P240va	1920 x 1080 pixel
Samsung Odyssey G5	2560 x 1440 pixel
Samsung Odyssey G7	3840 x 2160 pixel

A béta tesztet 10 fő végezte el. Elsősorban felhasználó jogosultsággal, majd később 2 fő kapott adminisztrátor, 1 fő műhelytulajdonos és 2 fő alkalmazott jogosultságot, és így tesztelték. Már a felhasználó jogosultsággal rendelkezőknél előkerült egy hiba: a kérés mérete túl nagy volt, így nem tudtak 1 képnél többet feltölteni. Természetesen ezt a futtatókörnyezetben hamar lehetett javítani, a kérés méretét a konfigurációban változtattuk. A másik probléma a jogosultság kezelésénél adódott a backend oldalon. A probléma az volt, hogy ha be van jelentkezve a felhasználó és kap egy jogosultságot, akkor a token nem frissült, de ezt javítottuk.

2.5. Fejlesztési lehetőségek

2.5.1. Üzenetküldés az oldalon

A jövőben szeretnénk beépíteni a felhasználó és a szerelőműhely közti levelezést, ami elősegítené az időpontfoglalást is.

2.5.2. Értesítés

A Backend oldalon elő van készítve már az értesítési központ, de még további fejlesztési lehetőségnek számítana, hogy minden egyes értesítést e-mailben is kiküldենék.

2.5.3. Értékelés, műhelyajánlás

Szeretnénk a jövőben bevezetni a műhelyértékelést, és létrehozni egy oldalt, ahol az összes műhely megjelenik, és lehetne szűrni város és értékelés szerint.

2.5.4. Többnyelvűség

Jelenleg csak magyar nyelven érhető el a webalkalmazás, de a jövőben az angol és a német nyelvet szeretnénk hozzáadni.

2.5.5. Natív alkalmazás

A végső célunk, hogy ne csak böngészőn keresztül lehessen elérni az oldalt, hanem fejleszteni egy telefonos és egy asztali alkalmazást, amivel még könnyebben lehetne használni a programunkat.

3. Felhasználói dokumentáció

3.1. Rendszerkövetelmények

Projektünk egy webes felület, ezért csak egy eszközt igényel, amivel az internethez lehet kapcsolódni. Ez lehet laptop, asztali számítógép, telefon vagy akár tablet is, mert a felületet optimalizáltuk az összes méretre. Az eszközön internetelésnek és a felsorolt böngészők egyikének kell lennie: Google Chrome, Brave, Microsoft Edge, Safari, Firefox.

3.2. Jogosultsági szintek

- **Felhasználó:**
A felhasználói jogosultságot minden egyes regisztrált felhasználó megkapja. A felhasználó ranggal hozzáadhat saját járművet és ezt kezelheti. Ezen kívül a felhasználó a saját adatait módosíthatja.
- **Adminisztrátor:**
Az adminisztrátori jogosultsággal lehet új műhelyeket létrehozni a rendszerben és ezeket kezelni. A műhely létrehozásával adja ki a műhelytulajdonos jogosultságot a megfelelő felhasználónak. Ezen kívül új járműtípusok, -márkák és -modellek felvételét végezheti. A felhasználói jogosultság megmarad.
- **Műhelytulajdonos:**
A műhelytulajdonos jogosultsággal kezelheti saját műhelyét. Felvehet és törölhet alkalmazottakat a műhelyében. Az alkalmazottak felvételével adja ki az alkalmazott jogosultságot a megfelelő felhasználónak. Ezen kívül a járművekhez képes hozzáadni szervizbejegyzéseket. A felhasználói jogosultság megmarad.
- **Alkalmazott:**
Az alkalmazott jogosultsággal a járművekhez lehet új szervizbejegyzéseket rendelni. Emellett a felhasználói jogosultság megmarad.

3.3. Felhasználói felület

A felhasználói felület több részből áll. A felület kezelése nem igényel különösebb tudást. Az oldalon mindenhol segédletek, képek és tippek állnak a felhasználó segítségére. A felület minden eszközön tökéletesen működik, akár telefon, akár asztali számítógép az.

3.3.1. Regisztráció

Regisztráció

A regisztrációval elfogadod a szerződési feltételeinket.
* A jelszónak minimum 8 karakter hosszúnak kell lennie.

[Van már fiókod?](#)

13. ábra: Regisztrációs felület

Amennyiben még nincs felhasználónk az alkalmazásban, akkor regisztrálni kell egyet. A regisztráláshoz 4 kötelező mezőt kell kitöltenünk és 1 választható mező áll rendelkezésünkre. A családnévünket, keresztnévünket, e-mail-címünket és jelszavunkat kell megadni, ezen felül megadhatjuk a telefonszámunkat. Az e-mail-cím nem lehet már regisztrált e-mail és valósnak kell lennie. A jelszónak minimum 8 karakternek kell lennie, és tartalmazni kell kis- és nagybetűt, ezen kívül számot is. A regisztrációs gombra kattintva az adatok formátumát ellenőrzi a rendszer. és a megadott e-mail-címre kapunk egy levelet, amivel aktiválni lehet a fiókunkat. Ha bármi probléma van a megadott adatokkal, a rendszer jelezni fogja.

3.3.2. Fiókaktiválás és bejelentkezés

Erősítsd meg az e-mail címed

Kattints a gombra, hogy megerősítsd a regisztrációd. Ha nem te voltál akkor vedd fel velünk a kapcsolatot.

Megerősít

Ha nem működik, akkor másold ki és illeszd be a böngészőbe:
<https://uniszervizkonyv.hu/aktivallas/63fe72f385e38695a59c154b/FHc4HobZyHtU3CzPlv4>

Üdvözléssel,
 UniversalServiceBook csapata

14. ábra: Aktiváló e-mail

A sikeres regisztráció után a megadott e-mail-címre kapunk egy levelet, amiben egy gomb van *Megerősítés* szöveggel. A gombra kattintva átirányít minket a weboldalra, ahol aktiválja a fiókunkat.

Ezek után már az aktív fiókkal szabadon bejelentkezhethetünk.



Bejelentkezés

* A jelszónak legalább 8 karakternek kell lennie.

[Nincs még fiókod?](#)

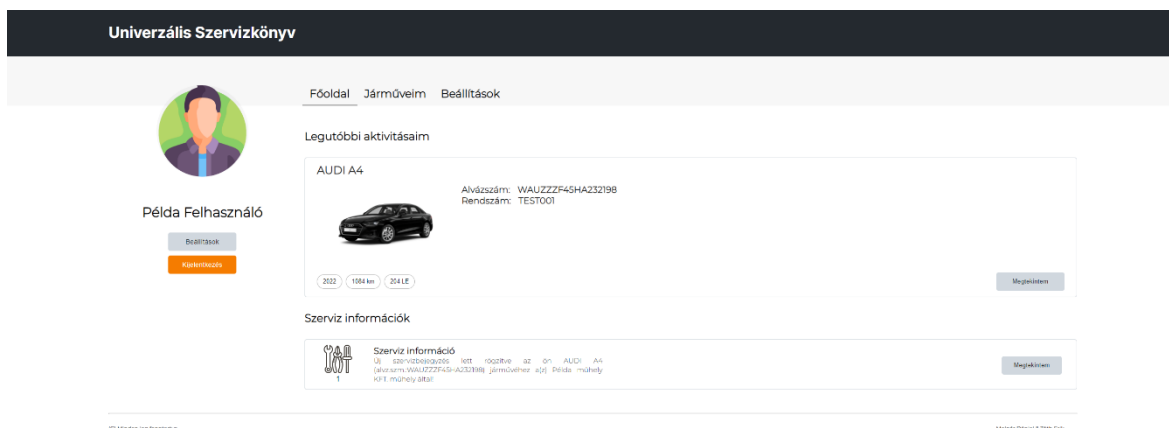
[Elfelejtett jelszó](#)

Belépés

15. ábra: Bejelentkezési felület

A bejelentkezéshez a megfelelő e-mail-cím és jelszó párost kell beírunk, ezután a *Belépés* gombra kattintva máris beléptünk az oldalra. Bármilyen hibás adatot adunk meg, az oldalon egy hibaüzenet jelenik meg.

3.3.3. Főoldal



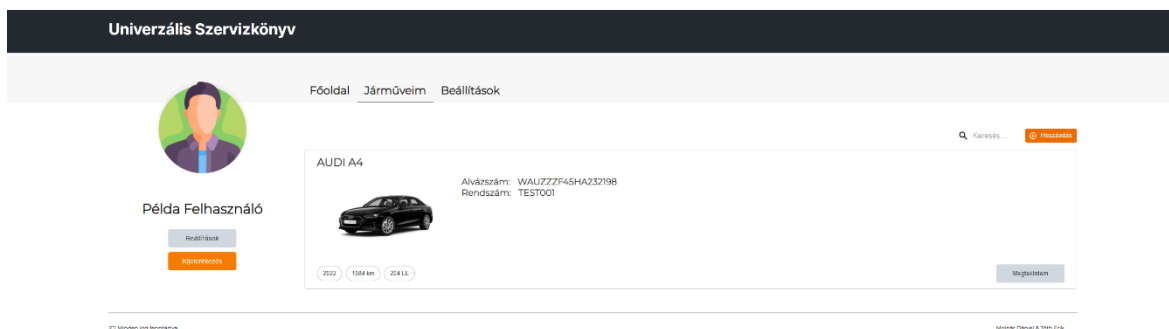
16. ábra: Főoldal

A sikeres bejelentkezés után a rendszer a főoldalra irányít minket. Ezen a felületen láthatjuk legutóbbi aktivitásainkat és a szervizinformációkat. A legutóbbi aktivitásban azok a járművek vannak, amiket legutoljára szerkesztettünk vagy megtekintettünk. A szervizinformációkban található a legutóbb hozzáadott szervizbejegyzés vagy a járművünk műszaki lejárata előre jelző értesítés.

Ezen felül láthatjuk a navigációs részt, ahol az oldalak között válthatunk.

A felület bal oldali részén láthatjuk profilképünket, egy *Beállítás* gombot, ami a beállítás oldalra irányít és egy *Kijelentkezés* gombot, amivel kijelentkezhetünk.


3.3.4. Járműveim



17. ábra: Járműveim oldal

A *Járműveim* navigációs gombra kattintva a rendszer átirányít minket a járműveim részre, ahol láthatjuk a már hozzáadott járműveket. Ezek egy rövid ismertetővel jelennek meg, mint például gyártási év, futásteljesítmény. Ezen felül megjelenik egy előnézeti kép, az alvázszám és a rendszám is. Ha még nem adtunk hozzá járművet, akkor „*Jelenleg nincs járműved!*” üzenet fogad a felületen. A navigációs menü alatt egy sávban elérhető a *Keresés* mező, amibe beírva a jármű márkáját vagy modelljét, visszaadja a keresett feltétel alapján a járművet, ha van ilyen, ha nincs, akkor „*Nincs ilyen jármű!*” üzenet fogad. Mellette találjuk a *Hozzáadás* gombot, amit, ha megnyomunk, megjelenik egy jármű hozzáadásához szükséges felület.

Új jármű rögzítése



Előnézeti kép feltöltése

Fájl kiválasztása | Nincs fájl kiválasztva

Jármű általános adatai

Alvázszám *

WUZZZF45H4232198

Rendszám

TE ST 001

Jármű kategória *

Autó

Jármű márkája **

AUDI

* Kérjük, hogy a mezőbe írt értéknek megfelelően válassza ki a legördülő listából.

Jármű modell **

A4

* Kérjük, hogy a mezőbe írt értéknek megfelelően válassza ki a legördülő listából.

Kivitel *

Sedan

Jármű évszáma *

2022

Jármű adatai

Km. óra állás *

5

* Ezen mező értéke 1 és 1500000 km lehet!

Saját tömeg *

1710

kg

* Ezen mező értéke 1 és 45000 kg lehet!

Teljes tömeg *

2180

kg

* Ezen mező értéke 1 és 45000 kg lehet!

Jármű motor adatai

Üzemanyag *

Dízel

Hengerűrtartalom *

1968

cm3

Teljesítmény *

204

LE

Hajtás *

Összekerek

Sabeségválasztás *

Automata

Jármű okmányai

Okmányok jellege **





Magyar

* Az okmányok jellege a jármű származási helyét várja értékel.

2028. 03. 01

Képgaléria

Fájl kiválasztása | Nincs fájl kiválasztva

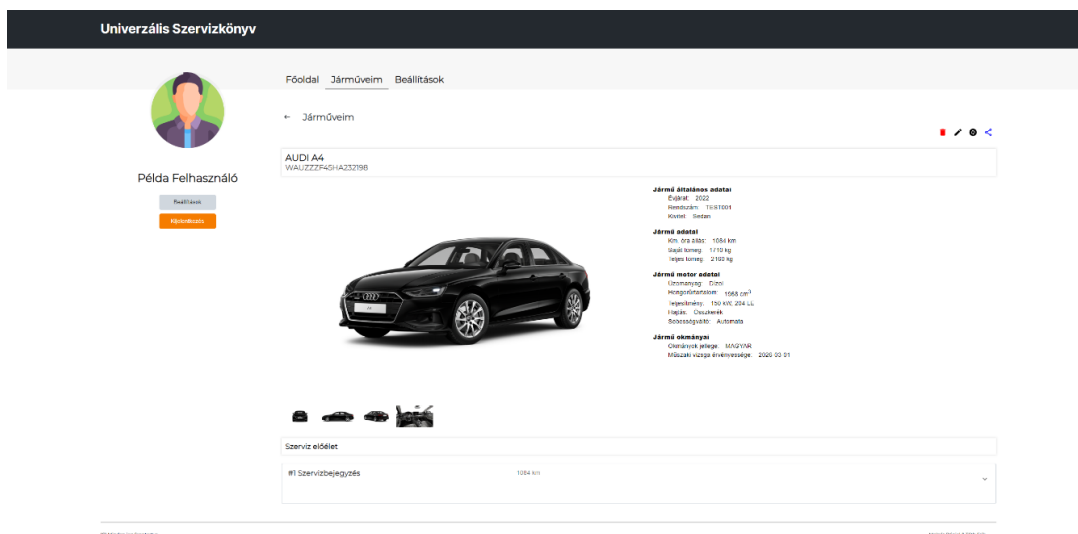
Megnéztem Mentés

18. ábra: Jármű hozzáadási felület

A jármű hozzáadásához szükséges pár beviteli mezőt kitöltenünk. A kötelező mező az alvázszám, aminek egyedinek és maximum 17 karakternek kell lennie, a jármű kategóriát

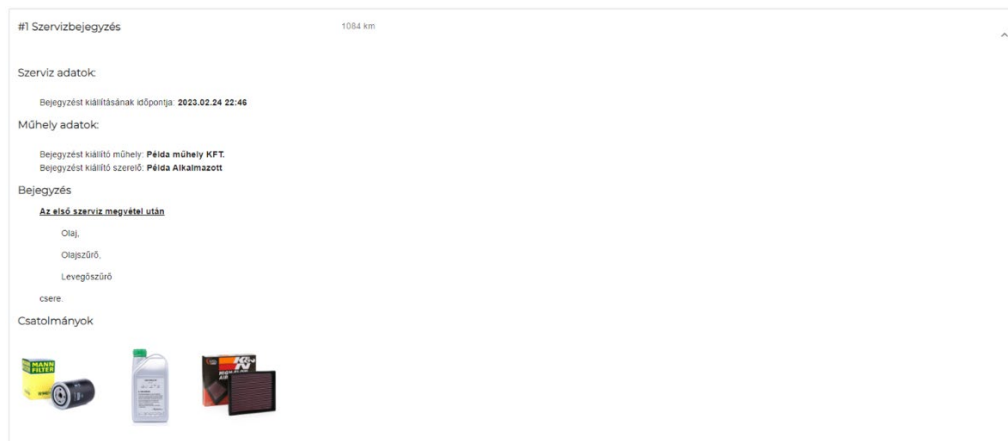
kell kiválasztani, alpból az autó kategória a kiválasztott. A kategória kiválasztása után ki kell választanunk a jármű márkáját, majd a modelljét. Ezek után a kivitel kategóriáját és a jármű évjáratát kell beírunk. A jármű adatai részénél a jelenlegi kilométeróra állását, saját tömegét és teljes tömegét kell megadnunk. A jármű motor adatai részénél az üzemanyag típusát, a hengerűrtartalmat, teljesítményét lóerőben, a hajtás típusát és sebességváltóját kell kitöltenünk. Az okmányok részénél a jármű származási helyét kell beírunk. A képgalériában egy előnézeti képet kell feltölteni, minimum 1-et és maximum 10-et. Ezen felül megadhatjuk a jármű rendszámát és a műszaki érvényességi lejárat dátumát. Ha mindent kitöltöttünk, akkor a *Mentés* gombra kattintva a rendszer feldolgozza az adatokat és elmenti a járművet. Ha bármilyen probléma van a bevitt adatokkal, akkor a rendszer tájékoztat minket.

3.3.5. Egy jármű megtekintése



19. ábra: Egy jármű felülete

Amikor a *Járműveim* résznél rákattintunk a *Megtekintés* gombra, akkor átirányít az adott jármű oldalára minket a rendszer. Itt láthatjuk járművünk összes adatát, képeit és szervizelőéletét. A képek mellett négy szekcióra bontva találjuk az adatokat: általános adatok, járműadatok, motoradatok és okmányok.



20. ábra: Szervizbejegyzés

Az adatok alatt található a szervizelőélet, ahol a *Szervizbejegyzések* vannak dátum és futásteljesítmény alapján rendezve, mindig a legfrissebb van legelöl. Egy szervizbejegyzésben láthatjuk a futásteljesítményt, a bejegyzés kiállításának időpontját, a kiállító műhelyt és szerelőt, magát a leírást, hogy mi történt a járművel és a hozzá kapcsolódó csatolmányokat, képeket.



21. ábra: Jármű menüsor

A *Jármű* menü részében található 4 funkció. Balról jobbra haladva a törlés, módosítás, tulajdonosváltás és megosztás.

Figyelmeztetés

Biztosan törölni kívánja az alábbi járművet?


Mégsem

Törlés

22. ábra: Törlés megerősítése

Ha a *Törlés* gombra kattintunk, akkor a rendszer egy megerősítést vár.

Szerkesztés



Fájl kiválasztása | Nincs fájl kiválasztva

* Saját tömeg

1710 kg

* Teljes tömeg

2160 kg

* Teljesítmény

204 l.p.

Rendszám



TEST001



* Országos jellege

MACYAR

Érvényességi dátum

2026. 03. 01.

Mégsem

Mentés

23. ábra: Jármű adatainak szerkesztése

Ha a *Módosítást* választjuk, akkor bejön egy új felület, ahol az előnézeti képet, saját tömeget, teljes tömeget, teljesítményt, rendszámot, származási helyet és a műszaki érvényességi dátumot lehet módosítani. Ezen felül a képgalériából tudunk képeket törölni.

Tulajdonosváltás

* Következő tulajdonos e-mail címe

Mégsem
Mentés

24. ábra: Tulajdonosváltási ablak

A *Tulajdonosváltás* gombra kattintva előugrik egy ablak, ahová a leendő tulajdonos e-mail-címét kell beírunk, majd rányomunk a mentés gombra, és már az új tulajdonosnál vannak az adatok.

Megosztás

<https://uniszervizkonyv.hu/megosztott/63f9266185e38695a59c0d9>

Bezárás
Megosztás jóváhagyása

25. ábra: Megosztás-kezelő ablak

A *Megosztást* választva feljön egy ablak, ahol kezelhetjük a megosztást: be- és kikapcsolhatjuk és a linket másolhatjuk ki.


3.3.6. Beállítások

Univerzális Szervizkönyv

Főoldal

Járműveim


Beállítások



Példa Felhasználó

beállítások

kiegészítés



Profilkép beállításai

Fájl kiválasztása Nincs fájl kiválasztva

Felhasználói adatok

Felhasználói e-mail

user@leszt.com

Keresztnév

Felhasználó

Családnév

Példa

Telefonszám

Város

Jelszavak kezelése

Régi jelszó

Új jelszó*

Új jelszó**

mentés

Elvetés

26. ábra: Beállítások oldal

A *Beállítások* oldalon módosíthatjuk profilképünket, keresz- és családnévünket, telefonszámunkat és a lakhelyünket adhatjuk hozzá. A jelszavunkat is módosíthatjuk, ha a jelenlegi jelszavunkat megadjuk. Az e-mail-címet nem módosíthatjuk, mert egyedi azonosítónak számít a rendszerben.

36


3.3.7. Oldalkezelői adminisztráció

Sorszám	Műhely név	Ország	Város	Cím	Alkalmazottak	
1	Példa műhely KFT.	Magyarország	Budapest	Főti út 86.	Megtekintem	Törítés
1-1 of 1						

27. ábra: Műhelyek táblázata

Amennyiben oldalkezelői (adminisztrátor) jogosultsággal rendelkezünk, a navigációs menüben megjelenik egy *Adminisztráció* gomb, ami az adminisztráció oldalra irányít. Ezen a felületen láthatjuk egy táblázatban a már létező műhelyeket és azoknak az alkalmazottait.

Univerzális Szervizkönyv



Példa Admin

Beállítások

Kijelentkezés

Főoldal

Járműveim

Adminisztráció

Beállítások

Műhely kezelő

+

Műhely hozzáadása

Műhely neve *

Motor és Autó műhely KFT.

Címzet *

Magyarország

Város *

Baleatonfüred

Utca *

Sipos utca 2

Műhelytulajdonos e-mail címe *

kiss.jasznel@gmail.com

Telefonos elérhetőség

+30703671234

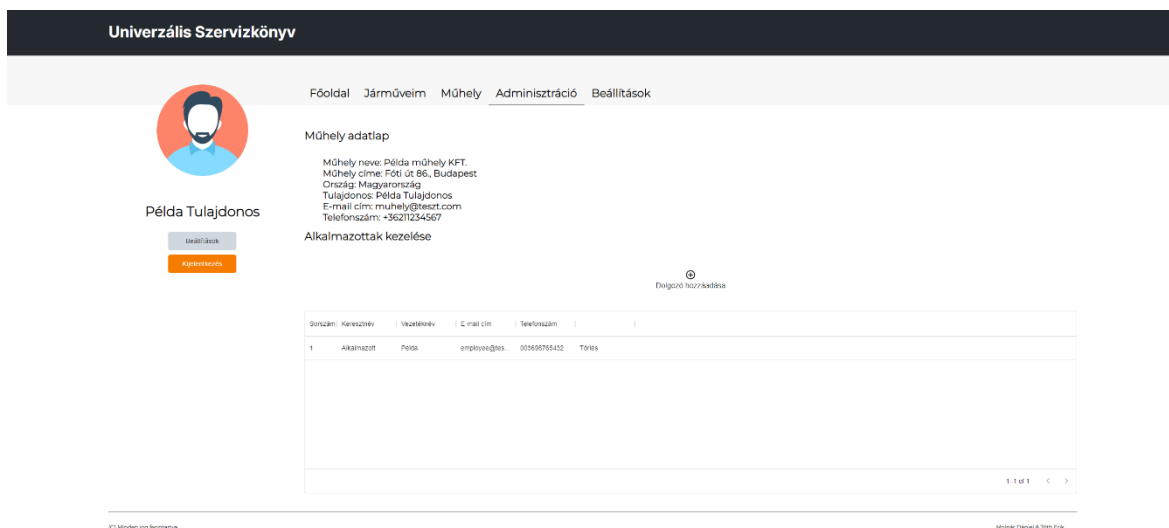
Műhely e-mail elérhetőség

+ Hozzáadás

28. ábra: Műhely hozzáadása

Ezen felül a *Műhely hozzáadása* gombra kattintva létrehozhatunk új műhelyeket a rendszerben, amivel kiadjuk a műhelytulajdonos jogosultságot a műhely tulajdonosának. A műhely létrehozásához a műhely nevét, országát, települését, címét, a tulajdonos e-mail-címét kötelező megadnunk, ezek mellett megadhatjuk a telefonos és e-mailes elérhetőséget is.

3.3.8. Műhely adminisztráció



29. ábra: Műhely adminisztrációs oldal

Amennyiben műhelytulajdonos jogosultsággal rendelkezünk, akkor a navigációs menüben megjelenik az adminisztrációs fül, ami a *Műhely adminisztráció* oldalra vezet. Az adminisztrációs részen láthatjuk műhelyünk adatait és az alkalmazottainkat.

The screenshot shows the 'Dolgozó hozzáadása' (Add Employee) form. It has a text input field for 'Dolgozó e-mail címe' (Employee email address) and a green 'Hozzáadás' (Add) button.

30. ábra: Dolgozó hozzáadása

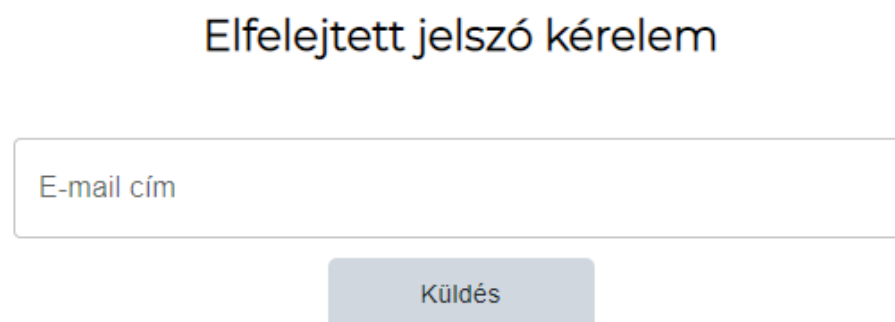
A *Dolgozó hozzáadása* gombra kattintva új alkalmazottat tudunk felvenni. Az új alkalmazott e-mail-címe megadása után a *Hozzáadás* gombra kattintva a rendszer ellenőrzi, hogy létezik-e vagy dolgozik-e egy másik műhelynél, ha nem, akkor sikeresen hozzáadja a műhelyhez, ezzel együtt az alkalmazott jogosultságot a felhasználóhoz. Ha sikertelen a művelet, akkor a rendszer értesít a hibáról. Ezen felül az alkalmazottakat törölni is tudjuk.

3.3.9. Műhely

31. ábra: Szervizbejegyzés hozzáadása

Ha műhelytulajdonos vagy alkalmazott jogosultsággal rendelkezünk, akkor a navigációs menüben megjelenik a *Műhely* gomb, amire, ha rányomunk, átirányít a *Műhely* oldalra. Itt egy jármű alvázszámának a beírása után a *Keresés* gombra kattintva megkeresi a járművet. Ha nem találja meg, akkor egy hibaüzenetet ad a rendszer. Ha sikeresen megtalálja a járművet, akkor láthatunk a járműről pár információt, mint például márka, modell, rendszám. Továbbá betölti a szervizbejegyzés létrehozásához szükséges mezőket. Kötelezően ki kell töltenünk a futásteljesítményt és a leírást. Feltölthetünk képet is a szervizbejegyzéshez. Ha mindent kitöltöttünk, és rákattintunk a *Feltöltés* gombra, akkor a rendszer ellenőrzi, hogy minden ki lett-e töltve és azt is, hogy a beírt futásteljesítmény nagyobb a régi futásteljesítményhez képest, ezzel a csalás szűrve van.

3.3.10. Elfelejtett jelszó



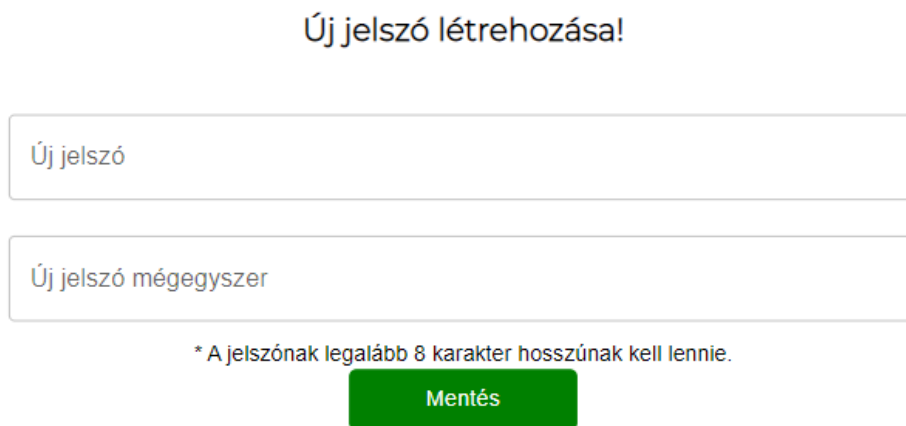
Elfelejtett jelszó kérelem

E-mail cím

Küldés

32. ábra: Elfelejtett jelszó oldala

Ha elfelejtettük a jelszavunkat, akkor a bejelentkezési oldalon az *Elfelejtett jelszó* gombra kattintva kérhetünk egy másikat. A gombra rányomva átirányít minket egy másik oldalra, ahol meg kell adnunk azt az e-mail-címünket, amivel regisztrálva vagyunk az oldalon. Az e-mail megadása után a *Küldés* gombra kattintva a rendszer kiküld nekünk egy levelet, amiben lesz a link.



Új jelszó létrehozása!

Új jelszó

Új jelszó megegyezően

* A jelszónak legalább 8 karakter hosszúnak kell lennie.

Mentés

33. ábra: Új jelszó beviteli oldala

A linkre kattintva megnyílik egy oldal, ahol meg kell adnunk az új jelszavunkat, aminek minimum 8 karakternek kell lennie. Ezután rá kell nyomnunk a *Mentés* gombra, hogy a rendszer elmenthesse az új jelszavunkat. Az e-mailben kapott link maximum 15 percig érvényes, ezután hiába nyomunk rá a *Mentés* gombra, hibát fog dobni a rendszer.

4. Összefoglalás

A program megírása során rengeteget fejlődtünk. Nagyon sok új készség alakult ki bennünk, a legfontosabb a csapatmunka és a munkamegosztás. Természetesen az elején sok hibát követtünk el az alkalmazás megtervezésében, de a projektfelosztással ezeket közösen hamar javítottuk. A program elkészítése során rengeteg szakmai tapasztalattal bővült a tudásunk, és megismerkedtünk egy komplex szoftver megírásának elemeivel, annak minden nehézségével együtt. Szakmai ismereteink legfőképpen a webfejlesztés irányába növekedtek. Megismertük a NodeJs-t, amit a projekt elején még nem annyira tudtunk használni, de a végén már rutinszerűen alkalmaztuk. A NoSQL-es adatbázisokkal is sokat dolgoztunk a program fejlesztése során, és tervezésben, optimalizálásában is sokat fejlődtünk.

A közös munkát összességében nagyon élveztük, volt pár nézeteltérés köztünk a fejlesztés közben, de azt megbeszéltük, és mentünk tovább.

Van pár funkció, amit nem tudtunk belerakni a programunkba, de a jövőben szeretnénk azt továbbfejleszteni, és ezeket a funkciókat hozzáadni az alkalmazáshoz.

5. Irodalomjegyzék

1. **Boduch, Adam** 2019 React Material-UI: <https://edu.anarcho-copy.org/Programming%20Languages/Node/react-material-ui-cookbook.pdf> (magyarra fordítva) Letöltés ideje: 2022.09.20
2. **Brown, Ethan** 2014 Web Development with Node & Express: https://www.vanmeegern.de/fileadmin/user_upload/PDF/Web_Development_with_Node_Express.pdf (magyarra fordítva) Letöltés ideje: 2022.09.20
3. **Durant, Kathleen** 2016 Introduction to NoSQL and MongoDB: <https://www.ccs.neu.edu/home/kathleen/classes/cs3200/20-NoSQLMongoDB.pdf> (magyarra fordítva) Letöltés ideje: 2022.09.28
4. **ExpressJS**: <https://expressjs.com/> (magyarra fordítva) Letöltés ideje: 2022.12.02
5. **GitHub**: <https://kinsta.com/knowledgebase/what-is-github/> (magyarra fordítva) Letöltés ideje: 2022.12.02
6. **GitKraken**: <https://dev.to/iphytech/gitkraken-what-is-it-and-common-actions-5531> (magyarra fordítva) Letöltés ideje: 2023.01.09
7. **JavaScript**: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript (magyarra fordítva) Letöltés ideje: 2022.10.13
8. **Jest**: <https://jestjs.io/> (magyarra fordítva) Letöltés ideje: 2022.12.02
9. **Material UI**: <https://mui.com/material-ui/getting-started/overview/> (magyarra fordítva) Letöltés ideje: 2022.11.18
10. **MongoDB Compass**: <https://www.mongodb.com/products/compass> (magyarra fordítva) Letöltés ideje: 2022.12.20
11. **NodeJS**: <https://nodejs.org/en/blog/announcements/foundation-v4-announce/> (magyarra fordítva) Letöltés ideje: 2022.11.23
12. **NoSQL**: <https://azure.microsoft.com/hu-hu/resources/cloud-computing-dictionary/what-is-nosql-database> Letöltés ideje: 2023.01.11
13. **ObjectId**: <https://desktop.arcgis.com/en/arcmap/latest/manage-data/using-sql-with-gdbs/object-id.htm> (magyarra fordítva) Letöltés ideje: 2023.01.11
14. **Postman**: <https://www.javatpoint.com/postman> (magyarra fordítva) Letöltés ideje: 2023.01.09

15. **React:** https://www.w3schools.com/whatis/whatis_react.asp (magyarra fordítva)
Letöltés ideje: 2022.10.13
16. **Visual Studio Code:** <https://code.visualstudio.com/docs/editor/whyvscode>
(magyarra fordítva) Letöltés ideje: 2022.12.20

6. Ábrajegyzék

1. ábra: JavaScript logó	8
2. ábra: React logó.....	8
3. ábra: Material UI logó	9
4. ábra: NodeJS logó.....	9
5. ábra: ExpressJS logó.....	10
6. ábra: Jest logó	10
7. ábra: GitHub logó	10
8. ábra: Visual Studio Code logó	11
9. ábra: MongoDB Compass logó	12
10. ábra: Postman logó	12
11. ábra: GitKraken logó	13
12. ábra: Adatbázis-modell	16
13. ábra: Regisztrációs felület	27
14. ábra: Aktiváló e-mail	28
15. ábra: Bejelentkezési felület.....	28
16. ábra: Főoldal	29
17. ábra: Járműveim oldal	30
18. ábra: Jármű hozzáadási felület.....	31
19. ábra: Egy jármű felülete	33
20. ábra: Szervizbejegyzés	33
21. ábra: Jármű menüsor.....	34
22. ábra: Törlés megerősítése	34
23. ábra: Jármű adatainak szerkesztése	34
24. ábra: Tulajdonosváltási ablak	35
25. ábra: Megosztás-kezelő ablak.....	35
26. ábra: Beállítások oldal	36
27. ábra: Műhelyek táblázata.....	37
28. ábra: Műhely hozzáadása	37
29. ábra: Műhely adminisztrációs oldal.....	38
30. ábra: Dolgozó hozzáadása	38
	44

Ábrajegyzék

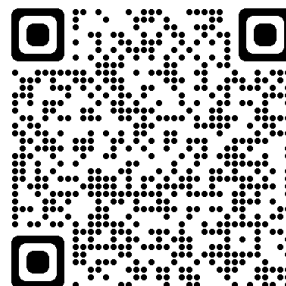
31. ábra: Szervizbejegyzés hozzáadása	39
32. ábra: Elfelejtett jelszó oldala	40
33. ábra: Új jelszó beviteli oldala	40

7. Mellékletek

GitHub Repository

Link:

<https://github.com/tothe98/UniversalServiceBook>



Univerzális szervizkönyv - Webalkalmazás

Link: <https://uniszervizkonyv.hu>

