



# EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

Információs Rendszerek Tanszék

## **Munkaidő nyilvántartó**

Témavezető:

Varga Dániel

Tanársegéd, Programtervező Informatikus MSc

Szerző:

Zsuravszkij Tamás

Programtervező informatikus BSc

*Budapest, 2023*

# Tartalomjegyzék

1.Bevezetés .....	1
1.1. Szakdolgozati témabejelentő .....	1
1.2. A dolgozat felépítése .....	1
1.3. Motiváció .....	2
2.Felhasználói dokumentáció.....	3
2.1 Megvalósítandó feladat.....	3
2.2 Hardver- és Szoftverkövetelmények.....	3
2.3 Weboldal felépítése .....	4
2.3.1 Főállású munkavállaló .....	5
2.3.2 Gyakornok .....	8
2.3.3 Munkacsoport vezető.....	9
2.3.4 Adminisztrátor .....	13
2.4 Weboldal telepítése .....	17
2.5 Kártya olvasó telepítése .....	17
3.Fejlesztői dokumentáció .....	20
3.1 Architektúra .....	20
3.2 Használati eset diagram .....	22
3.3 Adatbázis táblák.....	36
3.4 UML osztálydiagram .....	40
3.4.1 UserService .....	42
3.4.2 GroupService .....	43
3.4.3 RoleRepository .....	43
3.4.4 UsersInRoleRepository .....	44

3.4.5 Kártya olvasó .....	44
3.5 Alkalmazás részei .....	45
3.6 Tesztelés .....	46
3.7 Fejlesztési lehetőség .....	49
4.Hivatkozások .....	51
Szakirodalom .....	53
Mellékletek .....	53

# 1.Bevezetés

## 1.1. Szakdolgozati témabejelentő

A szakdolgozatom témája egy munkaidő nyilvántartó webes applikáció. Ez az applikáció a munkáltató, és a dolgozók munkáját szeretné elsősorban segíteni, leegyszerűsíteni az általában sokszor bonyolult folyamatokat. Átláthatóvá, valamint könnyebbé tenné a ledolgozott órák nyilvántartását és a szabadság kérelmek adminisztrációját.

Maga az applikáció rugalmasra lenne tervezve, ezáltal egyaránt használható teljes- illetve részmunkaidős munkavállalók és gyakornokok számára is. Kiszűrhetővé és áttekinthetővé tenné a dolgozók általi munkaidő sértéseket, ezzel növelve a cég hatékonyságát.

A programban szeretném megvalósítani az autentikálást és autorizálást. A szoftverben lenne lehetőségünk kimutatásokat készíteni, attól függően, hogy felhasználóként milyen jogosultsággal rendelkezünk. A szabadságolást egy jegyes rendszerrel szeretném megvalósítani, ami azt jelenti, hogy beadunk egy kérelmet, melyet a felettesünknek kell jóváhagynia. Illetve tartalmazna egy jogosultság kezelő platformot, ahol emberekhez rendelhetünk különböző munkaköröket. Mivel a programot eléggé rugalmasra tervezem, ezért lehetőség lenne egy beléptető kapus, mágnes-kártyás rendszerhez való csatlakozásra. Az adatok tárolása adatbázisban lesz megvalósítva, illetve az alkalmazás rendelkezni fog egy kényelmes webes felülettel.

## 1.2. A dolgozat felépítése

A szakdolgozatom dokumentációja három külön szekcióra van bontva. Most szeretném bemutatni a különböző szekciókat, külön kitérve ezek tartalmára.

A dolgozat I. szekciójában bemutatom a témám ötletét, valamint a mögöttes tartalmát, illetve, hogy mely felhasználók számára lesz hasznos a létrejött webalkalmazás.

A II. szekció tartalmazza a felhasználói dokumentációt. Az oldalak felépítését és használati bemutatását.

A III. szekció a fejlesztői dokumentációt tartalmazza, amelyben, kifejtésre kerülnek a használt architektúra, valamint a fejlesztői környezet leírása, illetve a tesztesetek.

## 1.3. Motiváció

A program létrehozásához való motivációt a felsőoktatásban töltött éveim alatt szerzett tapasztalat indította meg. Mint ahogy sok más egyetemi hallgató, én is arra kényszerültem, hogy diákmunkába kezdjek, ahol meglepően tapasztaltam, mennyire ódivatú és régimódi rendszerekkel adminisztrálják a munkaidő nyilvántartást. Mindig is érdeklődtem az elektronika iránt. Gimnázium éveim alatt megismerkedtem a robotikával, ahol tapasztalatot szereztem a mikrokontrollerek programozásában, ezért nagyon szerettem volna egy kézzel fogható kártyaleolvasót is készíteni a szakdolgozatomhoz. Ez inspirált engem, hogy a webalkalmazás mellé egy kártyaleolvasó modult készítsek, ami lehetővé teszi a kommunikációt a webalkalmazás és az olvasó között.

Létre szerettem volna hozni egy nagyon kényelmes weblapot, ahol a munkáltató, és munkavállaló számára modern és egyszerű körülmények között tudják lebonyolítani a munkaidő nyilvántartását. Egyetemi éveim alatt felidegesített, hogy a jelenléti óráimat az adott cégnél papíron kell vezetni, majd beszkenneálni és sok levelezés után leadni a HR-nek. Ezért létrehoztam egy univerzális platformot, ahol főállású munkavállalók, diákok, gyakornokok könnyedén kitölthetik a jelenléti ívüket, a főállású munkavállalók, kimutatásokat tudnak készíteni a ledolgozott órákról, szabadságokat tudnak beadni.

## 2.Felhasználói dokumentáció

### 2.1 Megvalósítandó feladat

Egy webes alkalmazás létrehozása a munkaidő nyilvántartás egyszerűsítésére és átláthatóvá tételére. Az alkalmazás a munkáltató és dolgozók számára lenne kifejlesztve, hogy segítsen kezelni a gyakran bonyolult folyamatokat.

Célja, hogy könnyebbé tegye a ledolgozott órák rögzítését és a szabadságkérelmek adminisztrációját. Az alkalmazás teljes- és részmunkaidős munkavállalók, valamint gyakornokok számára is hasznos lenne.

Ezen kívül lehetőséget biztosítana a dolgozók munkaidő sértéseinek kiszűrésére és áttekinthetővé tételére, ami hozzájárulna a cég hatékonyságának növeléséhez. Az alkalmazásban lehetőség lenne kimutatások készítésére is, attól függően, hogy milyen jogosultsággal rendelkezik a felhasználó

### 2.2 Hardver- és Szoftverkövetelmények

	Minimum követelmények	Ajánlott követelmények
Eszköz	Személyi számítógép (PC, Notebook)	Személyi számítógép (PC, Notebook)
Processzor	Intel Core i3-3xxx, vagy megegyező teljesítményű processzor	Intel Core i5-8xxx, vagy megegyező teljesítményű processzor
Belső memória	4 GB RAM	8 GB RAM
Kijelző felbontás	HD READY (720*1280) felbontást támogató monitor	HD (1080*1920) felbontást támogató monitor
Operációs rendszer	Windows 10	Windows 11
Böngészők	Google Chrome Mozilla Firefox Microsoft Edge	Google Chrome Mozilla Firefox Microsoft Edge

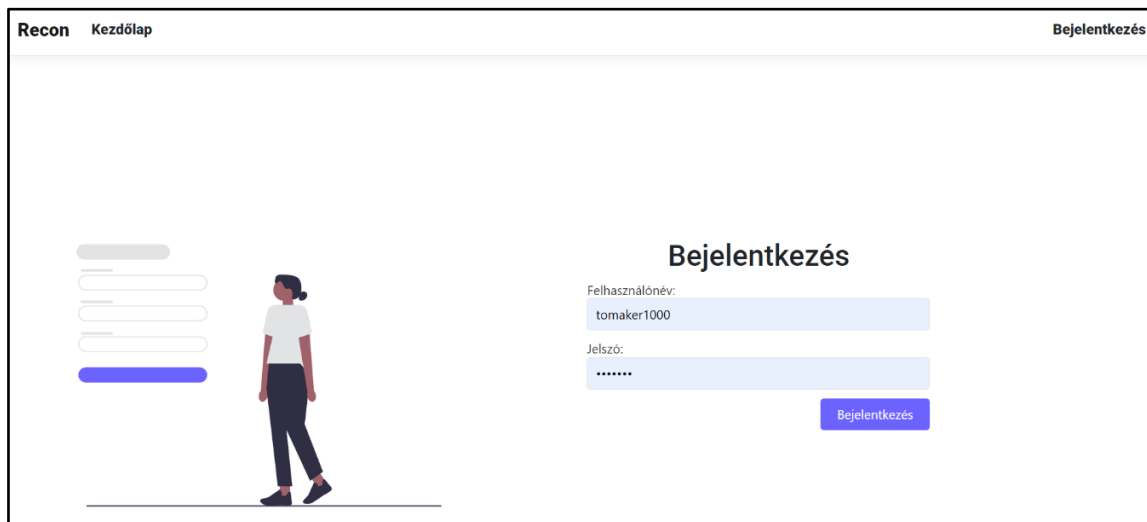
## 2.3 Weboldal felépítése



ábra 1: Kezdőlap

A Weboldalra érkező dolgozók elsősorban egy bejelentkezési felületet láthatnak, amit a 1. ábra szemléltet. A fejlécben megtalálható még egy Kezdőlap gomb, amelyre, ha rákattintunk akkor a kezdőlapra navigál.

A jobb felső sarokban található egy Bejelentkezés gomb, amelyre kattintva a bejelentkező oldalra navigálhatunk, ahol megadhatjuk felhasználónevünket, és jelszaunkat ahhoz, hogy bejelentkezzünk. Ezt a 2. ábrán láthatjuk.



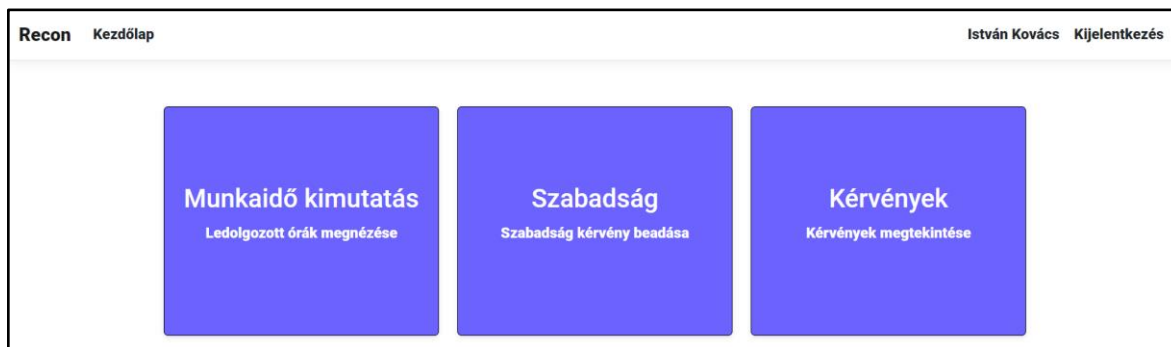
ábra 2: Bejelentkező oldal

Mivel a programot munkavállalók, valamint munkáltatók használják, mindenki saját jogosultsága szerint eltérő kezdőoldallal találkozhat, amit bejelentkezik.

A program tervezésénél alapjáraton 4 felhasználót különböztetek meg, de mivel modulárisra terveztem a weboldalt bármikor bővíthető az oldal.

- Főállású munkavállaló
- Gyakornok
- Munkacsoport vezető
- Adminisztrátor

Ebből fakadóan 4 különböző kezdőoldal találkozhatunk. A 3. ábra mutatja a főállású munkavállaló kezdőképernyőjét.



ábra 3: Főállású munkavállaló kezdőlapja

### 2.3.1 Főállású munkavállaló

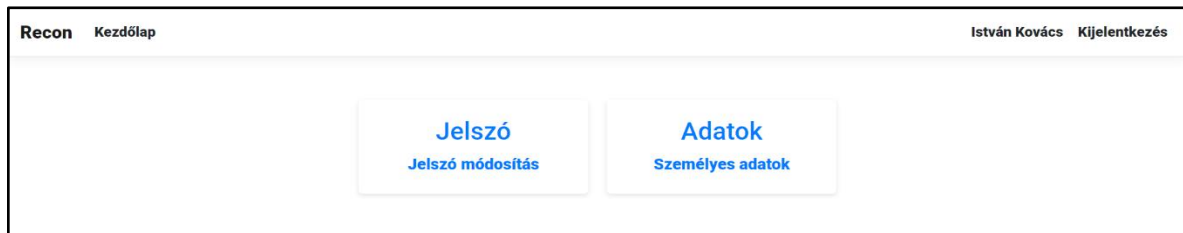
A jobb sarokban a fejlécnél megtalálható egy „Kijelentkezés” gomb, amelyre kattintva kijelentkezünk a weboldalról. Mellette a munkavállaló teljes neve látható, amire, ha rákattintunk, akkor egy irányítópult oldalra navigálunk, ahol két gomb fogad minket. Ezt a 4. ábrán tekinthetjük meg.

Ahhoz, hogy megváltoztassuk a jelszaunkat a „Jelszó” feliratú gombra kell kattintanunk, ami elnavigál minket egy jelszó módosító oldalra, ahol meg kell adnunk a régi jelszaunkat és az új jelszaunkat. Az „Adatok” feliratú gombra kattintva az alábbi személyes adatainkat adhatjuk meg:

- Keresztnév
- Vezetéknév
- Telefonszám



- Cím
- Település
- Megye
- Irányítószám
- Ország
- Életkor
- Nem



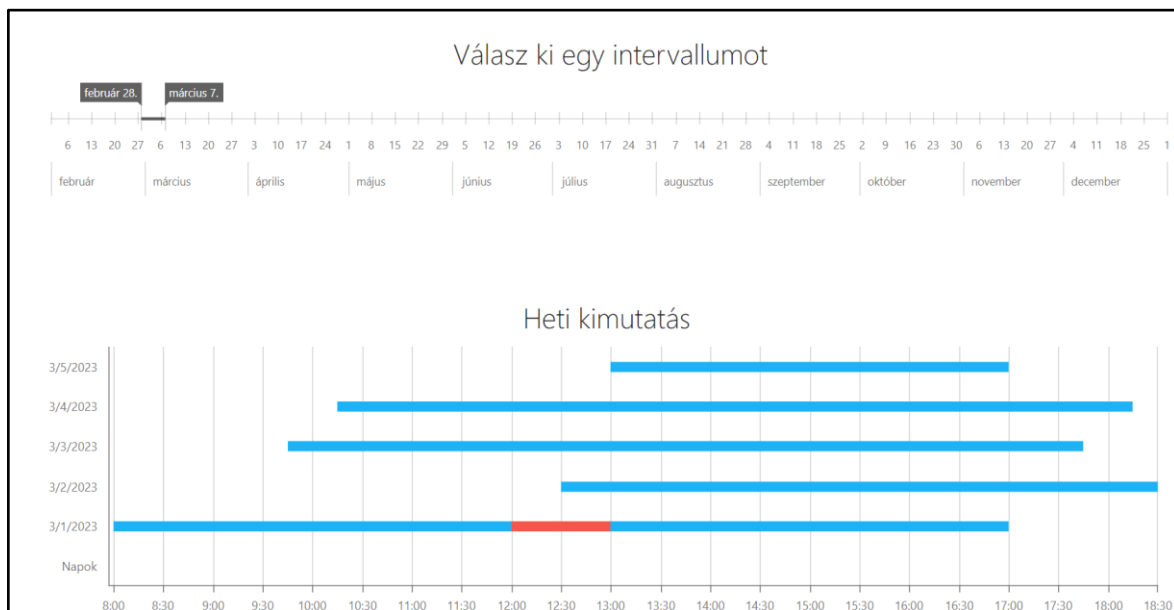
ábra 4: Irányítópult

A 3. ábrán középen 3 gomb található:

- Munkaidő kimutatás felirattal
- Szabadság felirattal
- Kérvények felirattal

Munkaidő kimutatás gombra kattintva megnézhetjük a felhasználó által ledolgozott órákat, illetve a szünettel töltött időt. Az 5. ábrán látható heti lebontás, amely grafikon formájában mutatja ki munkaidő nyilvántartást. A kép felső részén található egy csúszka, amelynek segítségével kiválaszthatjuk, hogy mely adott időperiódus kimutatását szeretnénk megtekinteni. A program kék színnel jelöli az irodában, munkahelyen töltött időszávot (ledolgozott órák), narancssárga színnel pedig a szüneteket.

Az oldalon megtalálható két téglalap, ahol összesítve az adott hétre a ledolgozott órák számát, illetve a szünetek számát tekinthetjük meg.



ábra 5: Dolgozói kimutatás

A szabadság feliratú kártyára kattintva szabadságokat adhatunk le a felettesünknek. Itt meg kell adnunk az alábbi adatokat:

- szabadság címét
- szabadság indokát
- munkacsoportot
- szabadság első napját
- utolsó napját

Mivel egy munkavállaló több csoportba is tarthat, ezért kell kiválasztanunk, hogy melyik munkacsoportba szeretnénk szabadságot kérni. Egy helyesen kitöltött szabadságot a 6.

ábrán

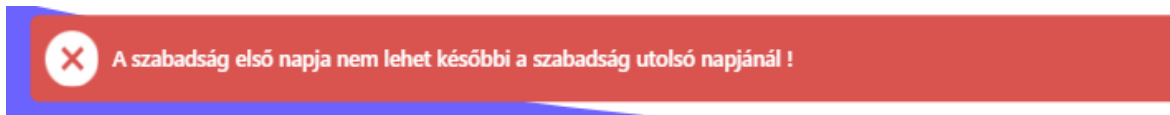
tekinthetjük

meg.

The form is titled 'Cím:' and contains the text 'Nyaralás 2023'. Below it is a 'Leírás:' field with the text 'Kedves Gipsz Jakab szeretnék szabadságot kérni .....'. The 'Szabadság első napja:' field shows '2023. 05. 11.' with a calendar icon. The 'Szabadság utolsó napja:' field shows '2023. 05. 26.' with a calendar icon. The 'Munkacsoport:' field shows 'Architect 1' with a dropdown arrow. A blue 'Mentés' button is at the bottom right.

ábra 6: Helyesen kitöltött szabadságos űrlap

Abban az esetben, ha a szabadság első napja későbbi, mint az utolsó akkor hibaüzenetet kapunk amit, a 7. ábrán látható





ábra 7: Hibaüzenet rossz szabadság beadásakor

„Kérvények” feliratú gombra kattintva a leadott szabadságos kérvényeket tekinthetjük meg. A zöld háttérrel rendelkező kérvények elfogadásra kerültek a felettesünktől, a sárga háttérrel rendelkezők még elfogadásra várnak.

## 2.3.2 Gyakornok


Ha bejelentkeztünk a weboldalra gyakornokként csak egy gomb jelenik meg a kezdőlap, ami a Jelenlétív oldalra navigál.

## Jelenléti ív

Jelenléti ív neve	Jelenléti ív dátuma	Felhasználó azonosítója	
Februári jelenlét	02/16/2023	11	
Junius jelenlet	04/20/2023	11	
Májusi jelenlét ív	04/26/2023	11	

ábra 8: Jelenléti ív listázása

A gombra kattintva megtekinthetjük a lezárt és jelenleg is aktív jelenlétünket egy listába, amit a 8. ábrán láthatunk. Minden jelenléti sorban megtalálható egy kis kék színű ceruza piktogram, amelyre kattintva megnézhetjük az adott hónapra kitöltött jelenléti ívünket, amit a 9. ábrán láthatunk.

 Mentés Excelbe					
Dátum	Óra	Perc	Hétvége	Dolgozó neve	Jóváhagyott
01/04/2023	3	2	<input checked="" type="checkbox"/>	Annamária Ma...	<input type="checkbox"/>
02/04/2023			<input checked="" type="checkbox"/>	Annamária Ma...	<input type="checkbox"/>
03/04/2023			<input type="checkbox"/>	Annamária Ma...	<input type="checkbox"/>
04/04/2023			<input type="checkbox"/>	Annamária Ma...	<input type="checkbox"/>

ábra 9: Egy nem lezárt jelenléti ív

A felhasználó abban az esetben tudja csak módosítani a jelenléti ívét, amennyiben a korábban ledolgozott órák még nem lettek jóváhagyva a felettese által. Amennyiben ez már megtörtént a rendszer megtiltja a felhasználó általi módosítást.

### 2.3.3 Munkacsoport vezető

Mivel egy felhasználó több csoportba is tartozhat ezért, ha munkacsoport vezető beosztással rendelkezik valaki, akkor két képernyőkép fogadhatja a kezdőlapon. Ha csak munkacsoport vezetők vagyunk és nem vagyunk más csoportokban beosztottak, akkor a 10. ábrán látható kép fogad minket bejelentkezéskor.



ábra 10: Munkacsoport vezető kezdőoldala

Viszont, ha munkacsoport vezetők vagyunk, valamint egy másik csoport beosztottjai is, akkor a 11. ábrán látható kezdőlap fogad minket.

A munkacsoport vezető jogosultsága az alábbi öt funkcióval rendelkezik:



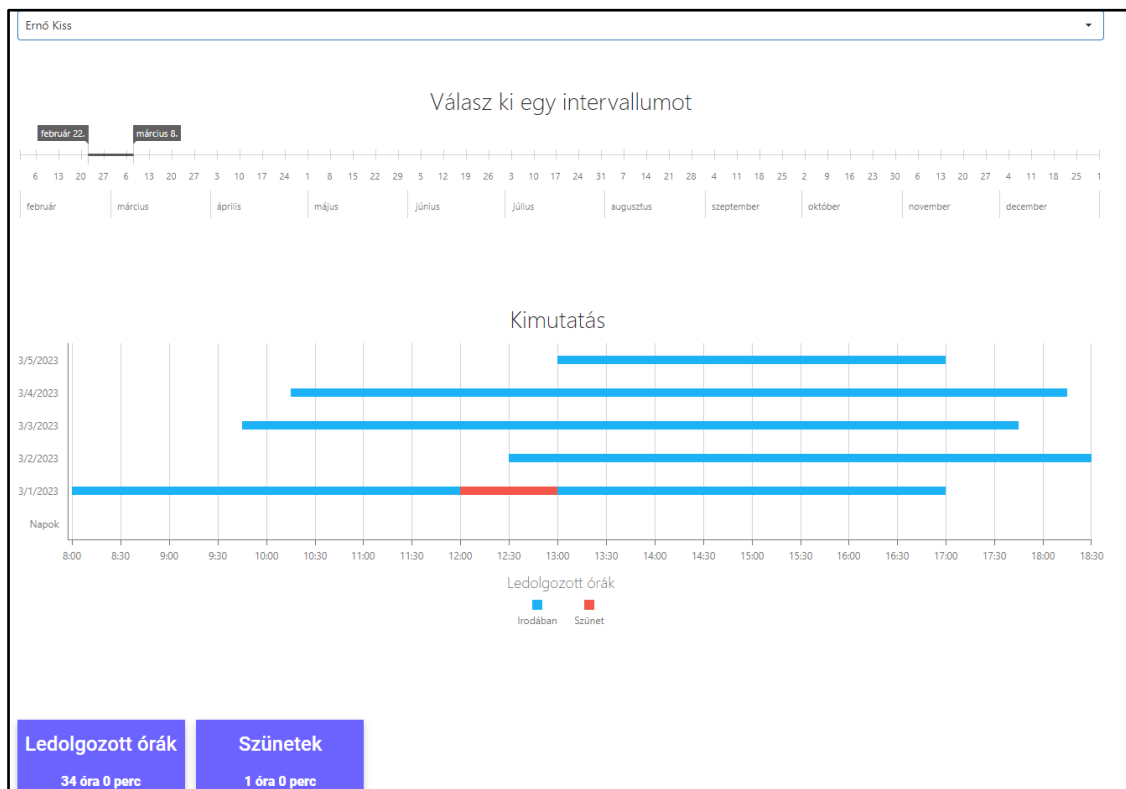
ábra 11: Munkacsoport vezető és beosztott egyben kezdőoldala

- Felhasználókat tud felvenni a munkacsoportokban.
- Új jelenlétiívet tud létrehozni a gyakornokok számára, akik a munkacsoportba tartoznak.

- Kimutatást tud készíteni a dolgozókról.
- Meglévő jelenlétiívet tud engedélyezni.
- Szabadság kérelmeket tud engedélyezni.

A munkacsoportvezető fel tud venni felhasználókat a munkacsoportjába, ahol meg kell adni a felhasználó nevét, illetve, hogy milyen munkacsoportba szeretné felvenni. Amennyiben egy felhasználó nem gyakornok, akkor több munkacsoportba is tartozhat. A munkacsoportvezető jelenléti ívet tud létrehozni azon gyakornokok számára, akik a munkacsoportba tartoznak.

A vezető azon dolgozók számára tud kimutatást készíteni, akik a munkacsoportjába tartoznak. Ahhoz, hogy ezt megtehesse egy legördülő listából választhatja ki melyik dolgozóra szeretne kimutatást készíteni.



ábra 12: Kimutatás oldal

Ezt a 12. ábrán megtekinthetjük.

Az ábrán látható adatok a dolgozó be-és kijárását jelképezik, a kék színnel az irodában töltött időt jelképezik, narancssárga színnel a szüneteket. Megtekinthetjük, a kép

bal sarkában, hogy összesen mennyit dolgozott a munkavállaló az adott intervallumban, és hogy mennyit volt szüneten.

Ahogy már a **2.3.2 fejezetben** kifejtettem a gyakornokok a munkaidejüket egy táblázat formájában vezetik, ezt a munkacsoport vezetőnek engedélyeznie kell. Ezt egy kattintással meg tudja tenni, illetve módosítani is tudja, ha esetleg úgy gondolja, hogy a gyakornok munkaidőt sértett, esetleg elírás történt.

Recon Kezdőlap		Erő Kiss Kijelentkezés	
Jelenlét ív neve	Létrehozás dátuma	Gyakornok neve	
Májusi jelenlét	05/19/2023	Albert Diák	+ 
Áprilisi jelenlét	04/20/2023	Albert Diák	+ 
Márciusi jelenlét	03/23/2023	Albert Diák	+ 

ábra 13: Jelenléti ívek listázása munkacsoport vezetői szemszögéből

A 13. ábrán látható egy lista az aktuális jelenléti ívekről, minden sorban található egy pipa jel amire, ha a felettes rákattint, akkor előugrik egy megerősítő ablak aminek jóváhagyásával az egész havi jelenlétet engedélyezi.

A pipa mellett megtalálható egy ceruza jelkép is, amire rákattintva a felettes módosíthatja az aktuális jelenléti ívet.

Miután engedélyezve lett, a jelenléti ív eltűnik a listából és többé nem módosítható. A gyakornokok megtekinthetik, illetve exportálhatják egy Microsoft Excel táblázatba bármikor a lezárást követően.

A 14. ábrán látható, az az oldal, ahol a vezető szabadságokat tud engedélyezni.



ábra 14: Szabadságolások

Itt megtalálható egy naptár, amiben az adott napra aktuális szabadságok vannak, ha a naptárban egy adott napra kattint a felettes, akkor megtekintheti, hogy milyen emberek, vannak aznap szabadságon. Ha duplán kattintunk a kijelölt napra picit több információt kapunk a szabadságról, mint például, hogy mi a szabadság első, illetve utolsó napja, mi az indoka a illetve a címe.

A csoportvezetőként beosztottak is vagyunk, akkor a **2.3.1-es fejezetben** kifejtett funkciókkal is rendelkezünk.

## 2.3.4 Adminisztrátor

Az adminisztrátor jogosultsággal belépve az 15. ábrán látható felület fogad minket.



ábra 15: Adminisztrátor kezdőlapja



Az adminisztrátor rendelkezik a legtöbb funkcióval, amik az alábbiak:

- Szerepköröket tud létrehozni.
- Felhasználóhoz tud szerepköröket rendelni.
- Munkacsoportokat tud létrehozni.
- Új felhasználót tud létrehozni.
- Belépő kártyát tud létrehozni.
- Felhasználókat tud felvenni munkacsoportokba.
- Új jelenléti ívet tud létrehozni a gyakornokok számára.
- Kimutatást tud csinálni bármely felhasználóról.

Alapértelmezetten 2 szerepet különböztetünk meg az „Admin” illetve a „Intern”, de ha valaki rendelkezik adminisztrátor szerepkörrel akkor létre tud hozni új szerepköröket, ahhoz csak meg kell adnunk a szerepkör nevét. Ha létrehoztunk egy szerepkört akkor tudjuk módosítani a szerepkörök nevét, illetve törölni is tudjuk őket.

A szerepköröket felhasználóhoz kell rendelnünk, hogy az adott felhasználó rendelkezzen az adott jogosultsággal. Ilyenkor a 15.ábrán látható második kártyára kell kattintanunk, ahol megjelenik egy lista ezt a 16. ábrán láthatjuk.

			+
Szerepkör	Felhasználó		
Admin	admin	×	
Intern	Diák Jenő	×	
Intern	Albert Diák	×	
Intern	Annamária Matolcsy	×	

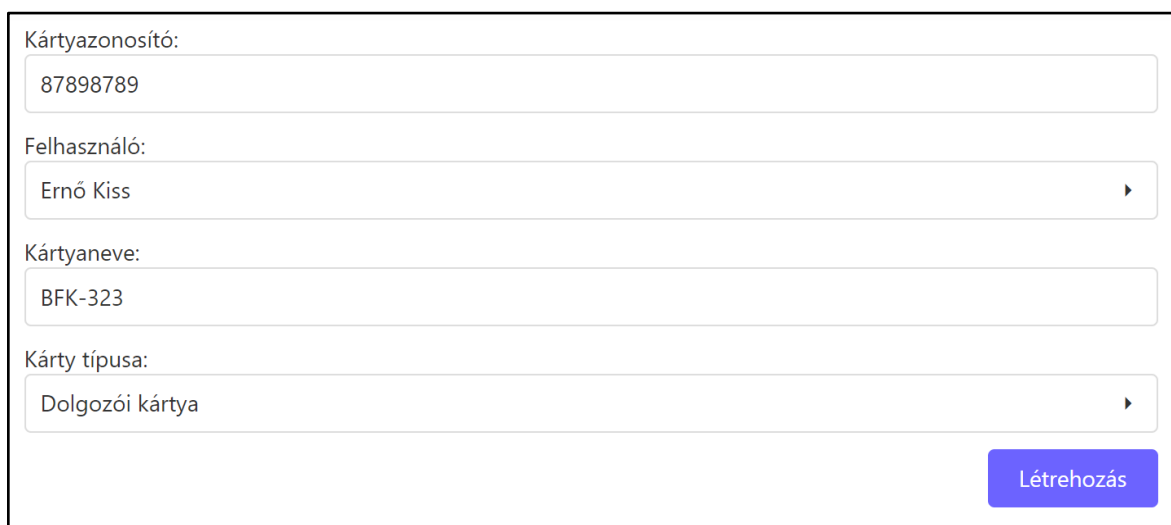
ábra 16: Felhasználók és szerepkörök összekapcsolása

A kép jobb sarkában találhatunk egy plusz gombot, amire rákattintva felhasználóhoz adhatunk szerepköröket, egy felhasználó több szerepkörbe is tartozhat. Minden sorban a listában megtalálunk egy kereszt gombot, amire, ha rákattintunk akkor törölni tudjuk a felhasználót az adott szerepkörből.

Adminisztrátori jogunkkal új munkacsoportot is tudunk létrehozni, ahol meg kell adnunk a munkacsoport nevét, illetve egy listából kiválasztani, hogy melyik felhasználó lesz a felettes az adott munkacsoportban. Miután létrehoztuk a munkacsoportot módosítani is tudjuk őket, módosíthatjuk a munkacsoport nevét, illetve a felettest is. A munkacsoportokat törölni is tudjuk.

Új munkatársat tudunk felvenni, ahhoz, hogy ezt megtegyük meg kell adnunk a munkatárs e-mail címét, illetve a felhasználó nevét. Ilyenkor egy alapértelmezett jelszóval létrejön a felhasználó, amit belépést követően érdemes megváltoztatni.

Ahhoz, hogy kimutatásokat készíthessünk a munkavállalóról, rendelkeznie kell egy mágneskártyával. A mágneskártya azonosítóval belépő kártyát tudunk létrehozni. Ahhoz, hogy ezt megtegyük meg kell adnunk a kártyaazonosítóját, a felhasználót, akihez rendelni szeretnénk a kártyát, a kártya nevét, illetve a kártya típusát, ezt láthatjuk a 17. ábrán.



Kártyazonosító:
87898789
Felhasználó:
Ernő Kiss
Kártyaneve:
BFK-323
Kártya típusa:
Dolgozói kártya
Létrehozás

ábra 17: Kártya létrehozás

Az alábbi három funkció:

- Felhasználókat tud felvenni munkacsoportokba.
- Új jelenlétiívet tud létrehozni a gyakornokok számára.
- Kimutatást tud csinálni bármely felhasználóról.



ábra 18: Nem elérhető oldal hibaüzenet

adminisztrátor



ábra 19: Nem található oldal hibaüzenet

A munkacsoport vezető funkcióit bővíti ki mivel, ha rendelkezünk adminisztrátori joggal akkor az összes munkacsoportba tud felhasználót felvenni nem kell, hogy vezetője legyen a munkacsoportnak. Az összes gyakornok számára tud jelenlétet létrehozni, úgyszintén nem kell, hogy az adott gyakornok a munkacsoportjába tartozzon. Az összes munkavállalóról tud kimutatást készíteni, nem kell, hogy munkacsoportba tartozzon az

A szoftvernek biztonságosnak kell lennie ezért, ha esetleg olyan oldalt szeretnénk elérni, amihez nincs jogosultságunk akkor a 18. ábrán látható hiba oldal jelenik meg. Esetleg rossz oldalra keverednénk vagy talán rossz azonosítót írunk be az oldal böngészőjébe és az nem létezik, akkor a 19. ábrán látható hibaoldal fogad minket.

## 2.4 Weboldal telepítése

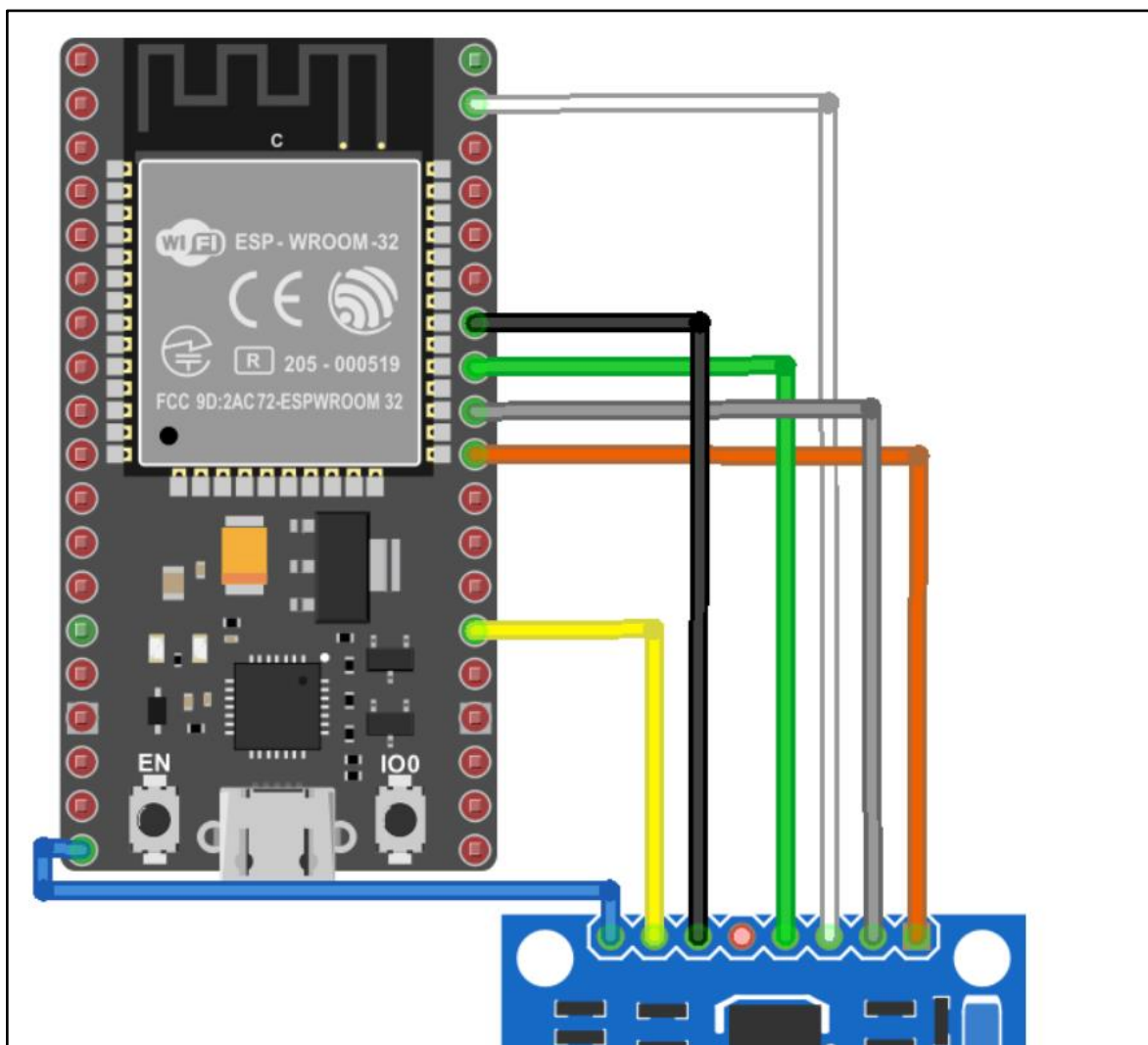
A Weboldal telepítéséhez csomagoljuk ki a „Recon.zip” fájlt ahhoz, hogy lokálisan tudjuk futtatni a webalkalmazást szükségünk van egy MS SQL szerverre. Ha sikerült telepítenünk az MS SQL szervert, ezek után futtassuk le az SQL parancssorokat, amit a „Data-Migration” könyvtárban találunk. Létrejöttek az adatbázisban a táblák a lokális számítógépen, ha ezzel készek vagyunk már csak össze kell kapcsolnunk az SQL adatbázisunkat és a webalkalmazásunkat.

A forrásfájlok között találunk egy appsettings.json fájlt ahol találunk egy „ConnectionStrings”- nevű objektumot, aminek van egy „DataDbContextConnection” paramétere, itt meg kell adnunk a saját SQL adatbázisunk elérését a Server attribútumnál, majd meg kell adnunk az adatbázis nevét, illetve egy UserID (felhasználónév), illetve Password (jelszó) attribútumot amivel csatlakozunk az adatbázishoz. Ha ezzel készen vagyunk akkor, indítsuk el a Recon.exe fájlt, és a <https://localhost:5000/> címen elérhetjük a weboldalt. Alapértelmezetten egy adminisztrátor felhasználóval léphetünk be melynek felhasználóneve : *admin*, jelszava : *Asd123!*

## 2.5 Kártya olvasó telepítése

Az alkalmazás elkészítésénél nagyon szerettem volna egy fizikai kártyaolvasót is megvalósítani. A kártyaolvasó, amit használtam az egy RFID RC522 modul, ami az Arduino mikrokontrollereket támogatja, továbbá szükségünk lesz egy esp wroom 32

mikrokontrollerre.



ábra 20: ESP32 mikrokontroller és RFID kártyaolvasó kábel kapcsolata

Az 20. ábra<sup>1</sup> szerint kössük össze a RFID modult a mikrokontrollerrel.

Ha ezzel készen vagyunk szükségünk lesz egy Micro USB csatlakozóra és egy számítógépre, mivel be kell konfigurálnunk a kontrollernek, hogy milyen hálózatra csatlakozzon. Ahhoz, hogy ezt megtegyük szükségünk lesz egy Visual Studio Code szerkesztőre. A PlatformIO bővítményt telepítenünk kell a szerkesztőbe, hogy fel tudjuk tölteni a forráskódunkat a mikrokontrollerre. Ha ezzel is kész vagyunk már csak a forráskód módosítása van hátra „Card-Reader/ESP32Wifi/src” mappában találunk egy main.cpp, ahol az 21. ábrán látható módon meg kell adnunk a hálózat nevét és jelszavát, ahol szeretnénk,

<sup>1</sup> <https://www.aranacorp.com/en/using-an-rfid-module-with-an-esp32/>

hogy a kártyaolvasó fusson, illetve a webalkalmazás elérési vonalát. Fontos, ha a saját számítógépünkön futtattjuk a webalkalmazást, akkor a mikrokontrollernek is azon a hálózaton kell, hogy fusson, illetve meg kell adnunk egy api kulcsot, alapértelmezetten ez a „TEST”- szöveg az api kulcs.

```
#define WIFI_SSID "Rozsak"
#define WIFI_PASSWORD "1064rozsa"

const char* serverName = "https://192.168.0.139:44378/api/MagneticCard";
```

ábra 21: Programkód a kártyaolvasó konfiguráláshoz

Miután be konfiguráltuk a mikrokontrollert, a 22. ábrán látható, egy pipa, amivel a



ábra 22: A mikrokontrollerre történő adatküldésről

forráskódot a

mikrokontrollerre tudjuk

írni. Ha mindent rendben konfiguráltunk be, akkor áramforrás alá kell helyezni a mikrokontrollert továbbá létre kell hozni egy kártyát a webalkalmazásban, és ha beregisztráltuk, akkor le ellenőrizhetjük, hogy ha a kártyát az olvasóhoz érintjük akkor az adatbázis „Checks” táblájában látnunk kell egy új sort az adott kártyaazonosítóval és a jelenlegi dátummal.

# 3.Fejlesztői dokumentáció

## 3.1 Architektúra

A szoftver 3 rétegű architektúrában készült, melyek a következők:

- UI: user interface (felhasználói felület)
- BL: business logic (üzleti logika)
- DB: database (adatbázis).

Ennek megvalósításához az ASP.NET Core keretrendszert használtam. Az MVC (.NET Model-View-Controller) az egyik leggyakrabban használt architektúra a modern webalkalmazások fejlesztéséhez. Az MVC egy szoftvertervezési minta, amely megkönnyíti az alkalmazások felépítését moduláris és könnyen karbantartható komponensekre bontja.

Az MVC architektúra három fő komponensből áll: Model, View és Controller.

- Model: Az alkalmazás adatmodelljét, az adatok kezelését és az üzleti logikát tartalmazza. A Model felelős az adatok lekérdezéséért, kezeléséért és frissítéséért. Ez lehet például adatbázis-kezelés, API-hívások, fájlkezelés vagy bármilyen más adatforrás kezelése. A Model szolgáltatja az adatokat a Controllernek a megjelenítéshez és feldolgozáshoz.
- View: Az alkalmazás felhasználói felületét (UI) tartalmazza. A View felelős az adatok megjelenítéséért és a felhasználóval való interakció kezeléséért. A View passzív komponens, azaz nem tartalmaz üzleti logikát. Csak a megjelenítésért felelős, és az eseményeket továbbítja a Controller felé.
- Controller: Az alkalmazás vezérlését és a felhasználói interakciók kezelését végzi. A Controller fogadja az eseményeket a Viewből, feldolgozza azokat, és megfelelő választ ad a Modell és a View közötti kommunikáció koordinálásával. A Controller felelős az alkalmazás állapotának frissítéséért, az adatok feldolgozásáért és az eredmények visszajuttatásáért a View felé.

Az MVC architektúra előnyei közé tartozik a szétválasztott felelősségi körök, ami lehetővé teszi a könnyű karbantarthatóságot és bővíthetőséget. Az alkalmazások különböző részei függetlenül fejleszthetők, mivel az egyes komponensek jól elkülönítettek egymástól.

Az MVC architektúra a .NET fejlesztésben az ASP.NET MVC keretrendszer alapjául szolgál, amely lehetővé teszi a könnyű és strukturált webalkalmazások fejlesztését a Model-View-Controller minta alapján.

Mivel a szakdolgozatomnál szerettem volna javítani a felhasználói élményt a felhasználói felületen az alábbi két könyvtárat használtam, a DevExtreme egy JavaScript alapú fejlesztői eszközkészlet, amely lehetővé teszi a modern és responszív webes alkalmazások fejlesztését. A DevExtreme kiterjedt funkciókkal rendelkezik és számos kész UI komponenst tartalmaz, amelyek lehetővé teszik a könnyű és gyors alkalmazásfejlesztést. A DevExtreme használata rendkívül egyszerű. Az eszközkészlet számos előre elkészített és testre szabható UI komponenst tartalmaz, például gombokat, űrlapokat, listákat, táblázatokat, grafikonokat és sok másat, amelyek segítségével könnyedén építhetünk fel modern és vonzó felhasználói felületeket.

A DevExtreme további előnye, hogy erős adatkezelési lehetőségeket kínál. Könnyen kezelhetjük az adatokat, például lekérhetjük, szűrhetjük, rendezhetjük, szerkeszthetjük és törölhetjük azokat a beépített adatforrásokkal vagy egyéb adatszolgáltatókkal, például REST API-kkal.

Az adatok vizuális megjelenítése is egyszerű a DevExtreme grafikon komponenseivel, amelyek különböző típusú grafikonokat támogatnak, például vonalgrafikonokat, oszlopdiagramokat, kördiagramokat és sok másat.

A második könyvtár pedig a Toast könyvtárra esik. A Toast egy könnyen kezelhető, JavaScript alapú értesítési könyvtár, amely lehetővé teszi a felhasználók számára rövid, időleges üzenetek megjelenítését a webes alkalmazásokban. A Toast segítségével könnyen és elegánsan jeleníthetünk meg értesítéseket, figyelmeztetéseket vagy információkat a felhasználó számára az alkalmazásban történő tevékenységekkel kapcsolatban. Felhasználói felületen szerettem volna vonzó és esztétikus kinézetet létrehozni, ezért használtam a Bootstrap könyvtárat, mely segítségével egy modern kinézetet tudtam készíteni a weblapomnak.

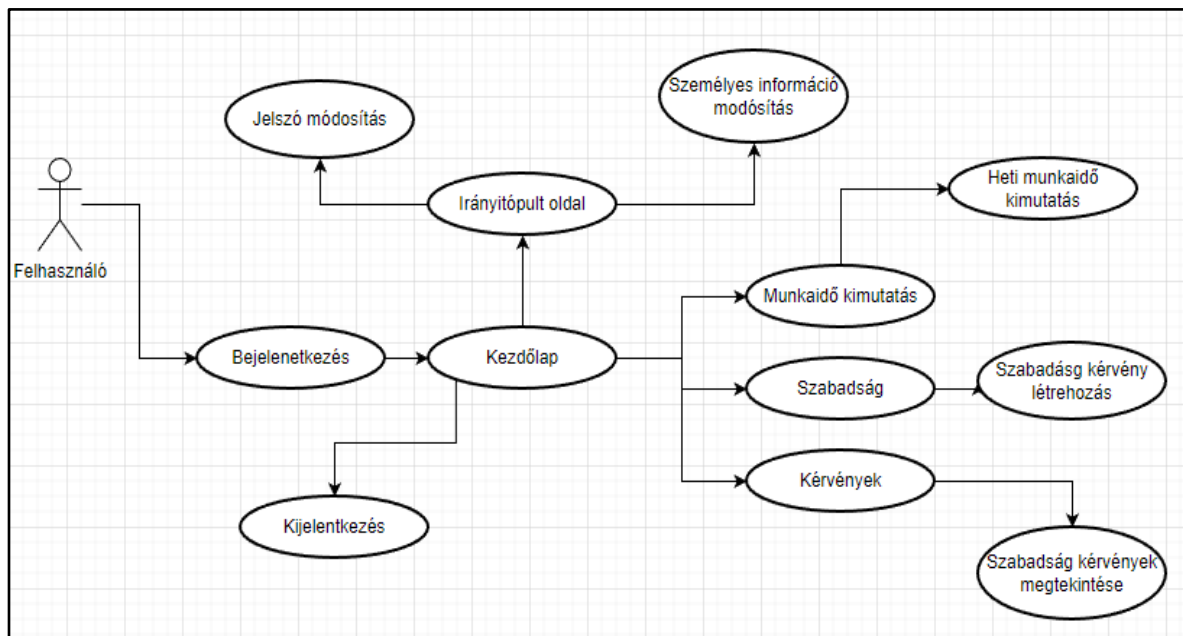
Mivel az alkalmazásomban adatokat is kell tárolnom az adatbázis választáskor egyértelműen adódott, hogy a MS SQL-t választom, mivel jól integrálható más Microsoft termékekkel és technológiákkal ezáltal sokkal könnyebben és gyorsabban tudtam fejleszteni az alkalmazásomat.



## 3.2 Használati eset diagram

A weboldalon 1 személy számít „felhasználónak”, aki több tevékenységet végezhet. Mivel több különböző jogosultsággal lehet rendelkezni az oldalon, ezáltal bemutatom a 5 különböző ezek a leggyakoribb esetet diagramok.

- 23. ábrán a főállású munkavállaló eset diagramja
- 24. ábrán a gyakornok eset diagramja
- 25. ábrán a munkacsoport vezető eset diagramja
- 26. ábrán a munkacsoport vezető, egyben beosztott eset diagramja
- 27. ábrán az adminisztrátor eset diagramja



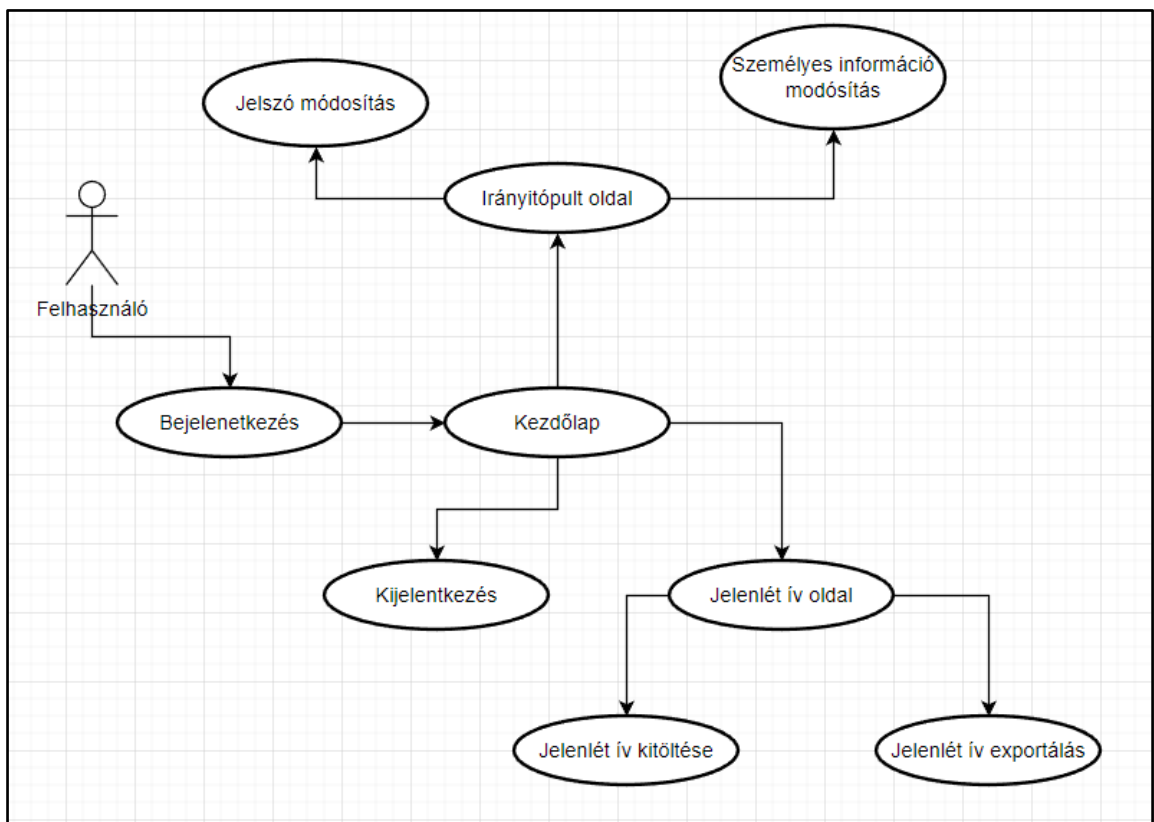
ábra 23: Főállású munkavállaló eset diagramja

Az lentebb látható user story táblázatban a főállású munkavállaló használati eset diagramjához a kiegészítése.

1.0.0	Sikeres bejelentkezés	GIVEN	Felhasználó bejelentkező oldalon van
		WHEN	Felhasználó beírja a helyes felhasználó nevét, illetve jelszavát.
		THEN	Sikeresen bejelentkezik
1.0.1	Sikertelen bejelentkezés	GIVEN	Felhasználó bejelentkező oldalon van
		WHEN	Felhasználó rossz felhasználó nevét, illetve jelszavát ad meg a bejelentkező

			fülön
		THEN	Hibaüzenetet kap a bejelentkező oldalon
1.0.2	Sikeres jelszó változtatás	GIVEN	Felhasználó jelszóváltoztatás oldalon van
		WHEN	Felhasználó bevan jelentkezve és helyesen megadta a régi, illetve az új jelszót
		THEN	Felhasználó sikeresen megváltoztatta a jelszavát és kap egy értesítő üzenetet erről
1.0.3	Sikertelen jelszó változtatás	GIVEN	Felhasználó jelszóváltoztatás oldalon van
		WHEN	Felhasználó bevan jelentkezve és rosszul adta meg a régi, illetve az új jelszót
		THEN	Felhasználó egy értesítő üzenetet kap, hogy sikertelen volt a jelszóváltoztatás
1.0.4	Felhasználó személyes adatokat szeretne feltölteni	GIVEN	Felhasználó személyes információ oldalon van
		WHEN	Felhasználó bevan jelentkezve és helyesen megadta a személyes információkat
		THEN	Felhasználó egy értesítő üzenetet kap, hogy Sikeres módosítottad a személyes adataidat
1.0.5	Felhasználó védett oldalt szeretne elérni	GIVEN	Felhasználó bevan jelentkezve
		WHEN	Felhasználó olyan weboldalt akar elérni, amelyhez nincs jogosultsága
		THEN	Hibaüzenetet kap, azzal a szöveggel, hogy sajnos nincs jogosultságod a weblap megtekintésére
1.0.6	Nem bejelentkezett felhasználó szeretne védett oldalakat elérni	GIVEN	Felhasználó nincs bejelentkezve
		WHEN	Felhasználó olyan weboldalt akar elérni, amelyhez be kell, hogy legyen jelentkezve
		THEN	Bejelentkező oldalra irányítja a felhasználót
1.0.7	Felhasználó be van jelentkezve, de semmilyen munkacsoporthoz nincs hozzárendelve	GIVEN	Felhasználó be van jelentkezve
		WHEN	Felhasználó olyan weboldalt akar elérni, amelyhez nincs jogosultsága
		THEN	Hibaüzenetet kap, azzal a szöveggel, hogy sajnos nincs jogosultságod a weblap megtekintésére
1.0.8	Kijelentkezés	GIVEN	Felhasználó be van jelentkezve
		WHEN	Kijelentkező gombra kattint
		THEN	Kijelentkezik a felhasználó
1.0.9	Felhasználó szabadsági	GIVEN	Szabadság kérvény beadás oldalon van
		WHEN	Felhasználó helyesen kitölti az adatokat

	kérvényt szeretne beadni	THEN	Sikeresen beadta a szabadság kérvényét
1.0.10	Felhasználó szabadsági kérvényt szeretne beadni	GIVEN	Szabadság kérvény beadás oldalon van
		WHEN	Felhasználó helytelenül tölti ki az adatokat
		THEN	Hibaüzenete kap, hogy sikertelen volt a szabadság kérelmed létrehozása
1.0.11	Múltbeli kérvényeket szeretné listázni	GIVEN	Kérvények oldalon van
		WHEN	A felhasználó adott már le szabadság kérelmet
		THEN	Ki listázva kapja meg a felhasználó az kérvényeinek előzményeit
1.0.12	Múltbeli kérvényeket szeretné listázni	GIVEN	Kérvények oldalon van
		WHEN	A felhasználó nem adott le szabadság kérelmet
		THEN	Egy üres weblapot kap, ahol azt a üzenetet kapja hogy Nincsenek szabadság kérelmeid
1.0.13	Munkaidő kimutatást szeretné megtekinteni	GIVEN	Munkaidő kimutatás oldalon van
		WHEN	Felhasználó rendelkezik belépési adatokkal
		THEN	A grafikonon megjelennek az adatok
1.0.14	Munkaidő kimutatást szeretné megtekinteni	GIVEN	Munkaidő kimutatás oldalon van és a felhasználó rendelkezik belépési adatokkal
		WHEN	Felhasználó a csúszkán kiválaszt egy hetet
		THEN	A grafikonon megjelennek az adott hétre az adatok



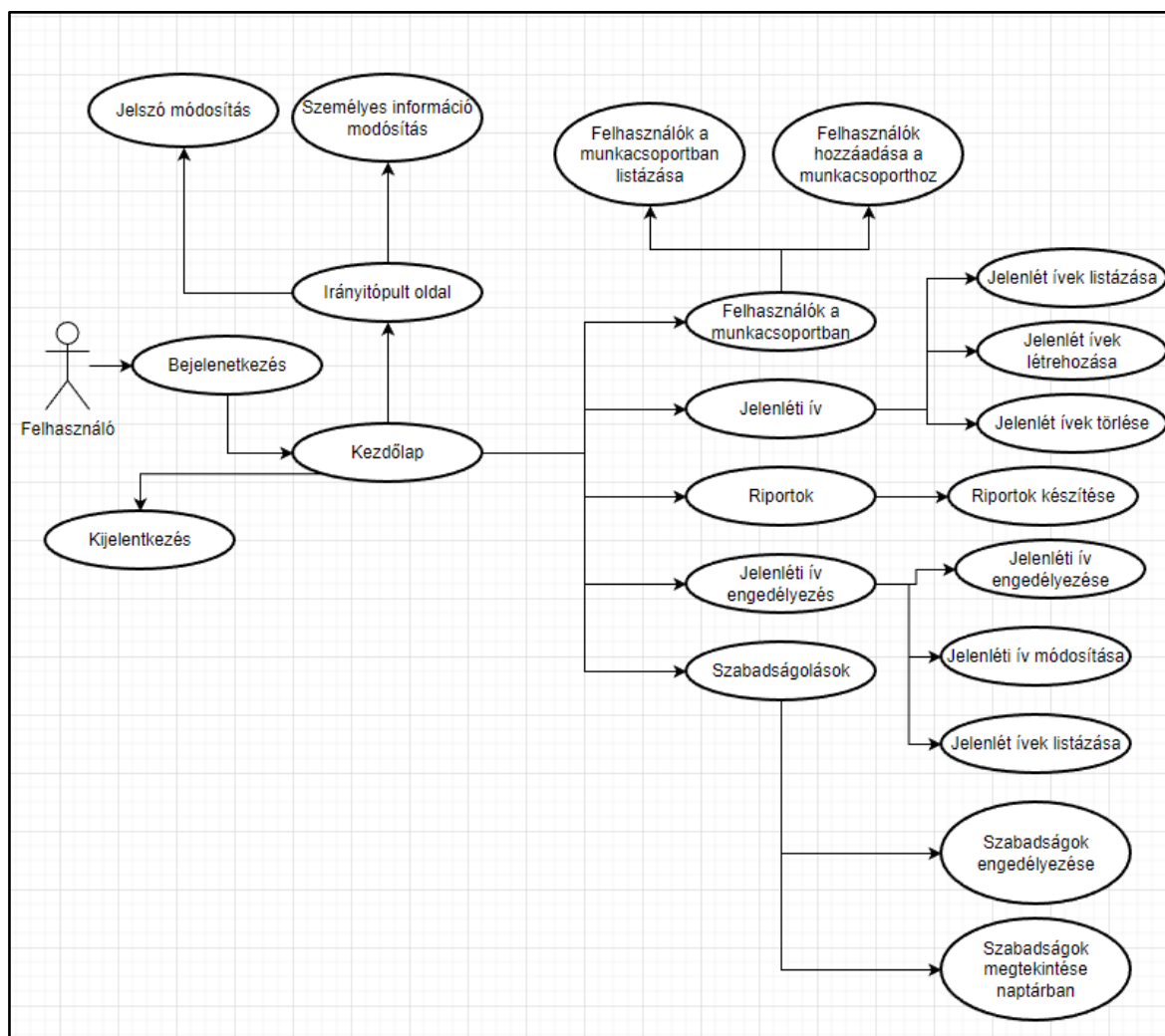
ábra 24: Gyakornok eset diagramja

Az lentebb látható user story táblázatban a gyakornoki használati eset diagram kiegészítése. Az alábbi user story táblázatban nem tüntettem fel, mivel ez általánosságban minden felhasználóra megegyezik a user story tábla 1.0.0-1.0.8-ig

2.0.0	Jelenlét ívet szeretné megnézni	GIVEN	Kezdőoldalon van a felhasználó
		WHEN	Jelenlét ív gombra kattint
		THEN	Egy listában megjelennek a jelenlétívek
2.0.1	Jelenlét ívet szeretné módosítani	GIVEN	Jelenlét ív oldalon van és kiválasztotta az aktuális jelenlétívet
		WHEN	A listában rákattint egy ceruza jelképre
		THEN	Sikeresen módosul a jelenlét ív adat
2.0.2	Jelenlét ívet szeretné módosítani	GIVEN	Jelenlét ív oldalon van és kiválasztotta az aktuális nem lezárt jelenlétívet
		WHEN	Nem létező jelenlétívre kattintott

		THEN	Hiba oldal töltődik be, hogy a keresett oldal nem található
2.0.3	Jelenlét ív órát szeretne módosítani nem lezárt jelenlét ívnél	GIVEN	Jelenlét ív táblázat meg van nyitva
		WHEN	Óra oszlopba 0-23 közötti számot ír be
		THEN	Helyesen módosul a jelenlétív óraszám
2.0.4	Jelenlét ív órát szeretne módosítani nem lezárt jelenlét ívnél	GIVEN	Jelenlét ív táblázat meg van nyitva
		WHEN	Óra oszlopba nem 0 és 23 közötti számot ír be
		THEN	Hibaüzenetet kap, hogy jelenlétív óraszám 0 és 23 között kell, hogy legyen
2.0.5	Jelenlét ív percet szeretne módosítani nem lezárt jelenlét ívnél	GIVEN	Jelenlét ív táblázat meg van nyitva
		WHEN	Perc oszlopba 0-59 közötti számot ír be
		THEN	Helyesen módosul a jelenlétív perc
2.0.6	Jelenlét ív percet szeretne módosítani nem lezárt jelenlét ívnél	GIVEN	Jelenlét ív táblázat meg van nyitva
		WHEN	Perc oszlopba nem 0-59 közötti számot ír be
		THEN	Hibaüzenetet kap, hogy jelenlétív perc szám 0 és 59 között kell, hogy legyen
2.0.7	Jelenlét ívet szeretné módosítani	GIVEN	Jelenlét ív oldalon van és kiválasztotta az aktuális jelenlét ívet, ami lezárt
		WHEN	Óra cellában kattint
		THEN	Nem történik semmi
2.0.8	Jelenlét ívet szeretné exportálni	GIVEN	Kiválasztotta az aktuális jelenlét ívet, ami nem lezárt
		WHEN	Mentés Excelbe gombra kattintott
		THEN	El mentődnek az adatok az Excelbe
2.0.9	Jelenlét ívet szeretné exportálni	GIVEN	Kiválasztotta az aktuális jelenlét ívet, ami lezárt
		WHEN	Mentés Excelbe gombra kattintott

		THEN	El mentődnek az adatok az Excelbe
--	--	------	-----------------------------------



ábra 25: Munkacsoport vezető eset diagramja

A lentebb látható user story táblázat a Munkacsoport vezető használati eset diagramját egészíti ki.

3.0.0	Felhasználók a munkacsoportban oldal navigáció	GIVEN	Kezdőoldalon van a felhasználó
		WHEN	Felhasználók a munkacsoportban gombra kattint
		THEN	Felhasználók a munkacsoportban oldalra navigál
3.0.1	Új felhasználó	GIVEN	Felhasználók a munkacsoportban oldalon

	hozzáadása a munkacsoport hoz		van
		WHEN	A plusz gombra kattint a felhasználó
		THEN	Egy űrlap oldalra navigál
3.0.2	Új felhasználó hozzáadása a munkacsoport hoz	GIVEN	Felhasználók a munkacsoport létrehozás oldalon van
		WHEN	Helyesen kitöltötte az űrlapot
		THEN	Üzenetet kapunk, hogy sikeresen hozzárendelted a felhasználót a munkacsoporthoz
3.0.3	Új felhasználó hozzáadása a munkacsoport hoz	GIVEN	Felhasználók a munkacsoport létrehozás oldalon van
		WHEN	A kiválasztott felhasználó már a munkacsoport tagja
		THEN	Üzenetet kapunk, hogy nem Sikerült hozzárendelni a felhasználót a munkacsoporthoz
3.0.4	Gyakornok hozzáadása a munkacsoport hoz	GIVEN	Felhasználók a munkacsoport létrehozás oldalon van
		WHEN	A kiválasztott gyakornok már más munkacsoportba tartozik
		THEN	Üzenetet kapunk, hogy nem Sikerült hozzárendelni a felhasználót a munkacsoporthoz
3.0.5	Felhasználó törlése a munkacsoport ból	GIVEN	Felhasználók a munkacsoport oldalon van
		WHEN	Listában az x gombra kattint
		THEN	Üzenetet kapunk, hogy nem rendelkezel a megfelelő jogosultsággal
3.0.6	Jelenlétív oldal	GIVEN	A felhasználó a kezdőlapen található
		WHEN	Jelenlét ív gombra kattint
		THEN	Jelenlét ív oldalra navigál
3.0.7	Jelenlétív létrehozás	GIVEN	A felhasználó a jelenlét ív oldalon található
		WHEN	Plusz gombra kattint
		THEN	Jelenlét ív létrehozás oldalra navigál
3.0.8	Jelenlétív létrehozás	GIVEN	A felhasználó a Jelenlét ív létrehozás oldalon található
		WHEN	Helyesen kitöltött az űrlapot küld be a felhasználó
		THEN	Üzenetet kap, hogy helyesen létrehoztad a jelenlét ívet
3.0.9	Jelenlétív létrehozás	GIVEN	A felhasználó a Jelenlét ív létrehozás oldalon található

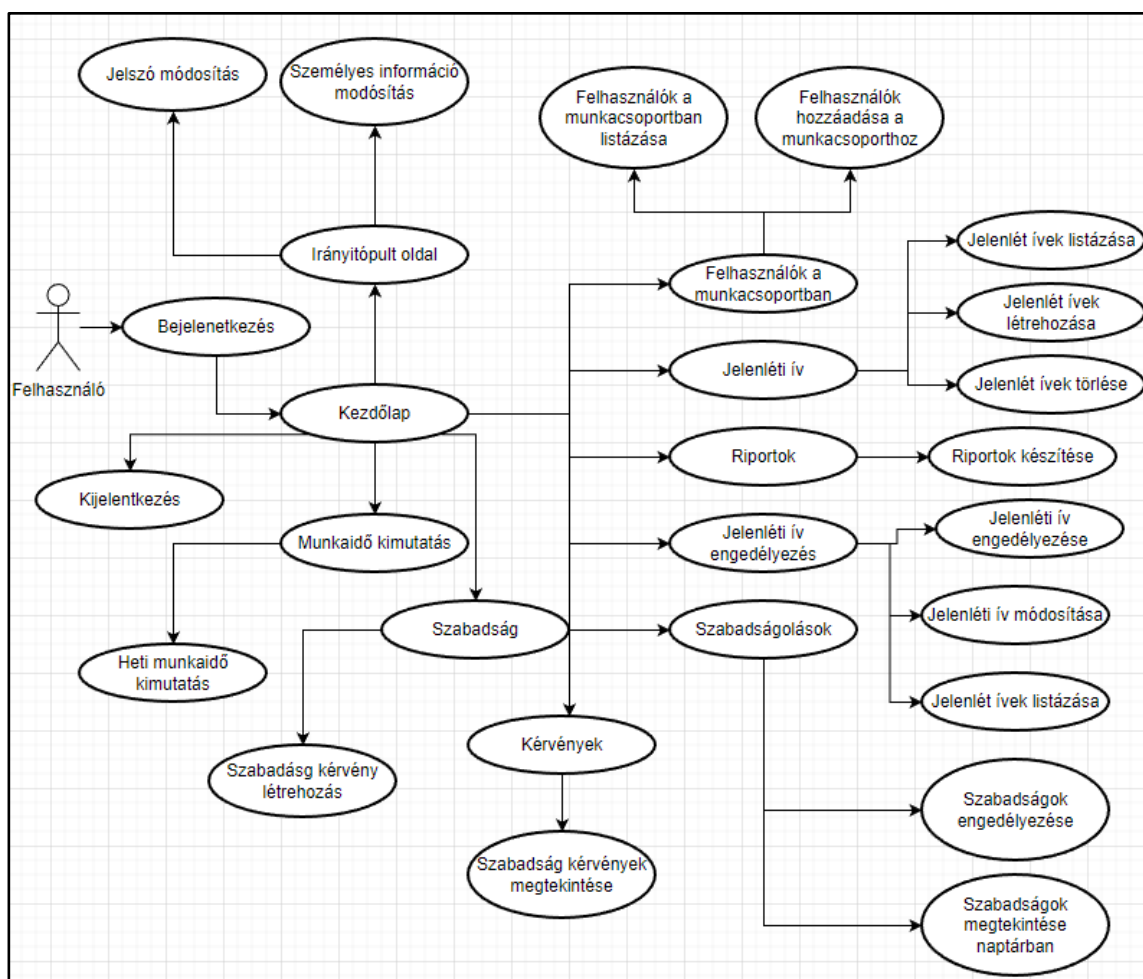
		WHEN	Nem töltötte ki az űrlapot helyesen
		THEN	Üzenetet kap, hogy Hibás adatokat adtál meg
3.0.10	Jelenlétív törlés	GIVEN	A felhasználó a Jelenlét ív lista oldalon található
		WHEN	A sorok végén található x gombra kattint
		THEN	Üzenetet kap, hogy sikeresen törölted a jelenléti ívet
3.0.11	Riportok oldalra navigálás	GIVEN	A felhasználó a kezdőoldalon található
		WHEN	Riportok feliratú gombra kattint
		THEN	Riportok oldal jelenik meg
3.0.12	Riportok készítése egy felhasználóról	GIVEN	A felhasználó a riportok oldalon található
		WHEN	A legördülő listából kiválasztja a felhasználót, aki rendelkezik belépési adatokkal az adott intervallumon
		THEN	A megjelenik egy grafikonon, ahol az adatok a felhasználóról láthatóak
3.0.13	Riportok készítése egy felhasználóról	GIVEN	A felhasználó a riportok oldalon található és kiválasztott egy felhasználót
		WHEN	A felhasználó az intervallumot változtatja és rendelkezik a felhasználó belépési adatokkal az adott intervallumhoz
		THEN	A Grafikonon megjelennek az adatok a felhasználóról az adott intervallumra
3.0.14	Riportok készítése egy felhasználóról	GIVEN	A felhasználó a riportok oldalon található és kiválasztott egy felhasználót a legördülő listából
		WHEN	A felhasználó nem rendelkezik belépési adatokkal
		THEN	Üzenetet kapunk, hogy nem található belépési adatok a felhasználóhoz
3.0.15	Riportok készítése egy felhasználóról	GIVEN	A felhasználó a riportok oldalon található és kiválasztott egy felhasználót
		WHEN	A felhasználó az intervallumot változtatja és nem rendelkezik a felhasználó belépési adatokkal az adott intervallumhoz
		THEN	Üzenetet kapunk, hogy nem található belépési adatok a felhasználóhoz az adott intervallumra
3.0.16	Jelenlétív engedélyezés oldalra	GIVEN	A felhasználó a kezdőoldalon található
		WHEN	A felhasználó a Jelenléti ív engedélyezés gombra kattint



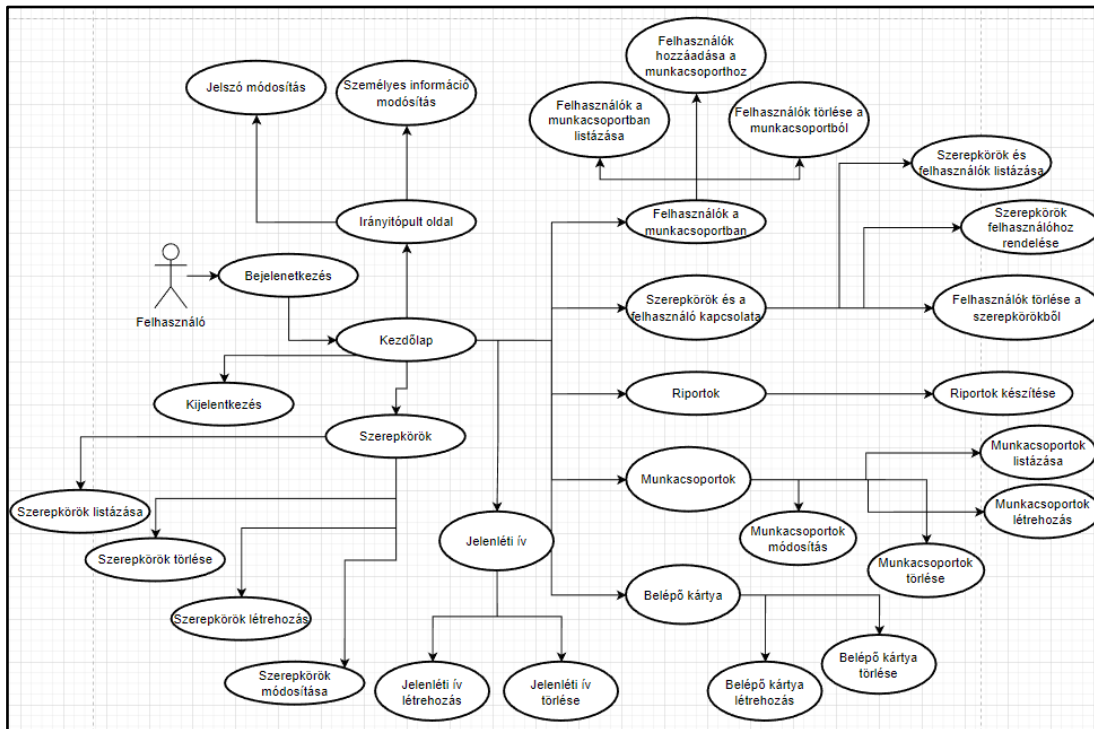
	navigálás	THEN	Jelenléti ív engedélyezés oldalra navigálunk
3.0.17	Jelenléti ív lezárása	GIVEN	A felhasználó a jelenléti ív engedélyezés oldalon található
		WHEN	A felhasználó a lista soraiban a + gombra kattint
		THEN	Megjelenik egy dialógus ablak
3.0.18	Jelenléti ív lezárása	GIVEN	Megjelent a dialógus ablak
		WHEN	A felhasználó az igen gombra kattint
		THEN	Eltűnik a dialógus ablak és üzenetet kap a felhasználó, hogy sikeresen engedélyezted a jelenléti ívet és eltűnik a listából a jelenléti ív
3.0.19	Jelenléti ív lezárása	GIVEN	Megjelent a dialógus ablak
		WHEN	A felhasználó az nem gombra kattint
		THEN	Eltűnik a dialógus ablak és nem történik semmi
3.0.20	Jelenléti ív lezárása	GIVEN	Megjelent a dialógus ablak
		WHEN	A felhasználó az nem gombra kattint
		THEN	Eltűnik a dialógus ablak és nem történik semmi
3.0.21	Jelenléti ív módosítás	GIVEN	A felhasználó a jelenléti ív engedélyezés oldalon található
		WHEN	A felhasználó a lista soraiban a ceruza gombra kattint
		THEN	Megjelenik a jelenléti ív napokra lebontva
3.0.22	Jelenléti ív módosítás	GIVEN	A felhasználó a jelenléti ív módosítás oldalon van
		WHEN	A felhasználó az óra oszlopba nem 0-23 közötti számot ír be
		THEN	A felhasználó hibaüzenetet kap, hogy az értéknek 0 és 23 között kell lennie
3.0.23	Jelenléti ív módosítás	GIVEN	A felhasználó a jelenléti ív módosítás oldalon van
		WHEN	A felhasználó az óra oszlopba 0-23 közötti számot ír be
		THEN	Az óra oszlopban megváltozik az adott óra mennyisége
3.0.24	Jelenléti ív módosítás	GIVEN	A felhasználó a jelenléti ív módosítás oldalon van
		WHEN	A felhasználó az perc oszlopba nem 0-59 közötti számot ír be

		THEN	A felhasználó hibaüzenetet kap, hogy az értéknek 0 és 59 között kell lennie
3.0.25	Jelenléti ív módosítás	GIVEN	A felhasználó a jelenléti ív módosítás oldalon van
		WHEN	A felhasználó az perc oszlopba 0-59 közötti számot ír be
		THEN	Az perc oszlopban megváltozik az adott perc mennyisége
3.0.26	Szabadság oldalra navigálás	GIVEN	A felhasználó a kezdőlapra található
		WHEN	A felhasználó a szabadságolások gombra kattint
		THEN	A felhasználó a szabadságolások oldalra navigál
3.0.27	Szabadságok listázása	GIVEN	A felhasználó a szabadságolások oldalon található
		WHEN	Az adott munkacsoportban vannak szabadság kérvények
		THEN	A kalendárium alatt sárga színnel megjelennek az adott szabadság kérvények
3.0.28	Szabadságok engedélyezés	GIVEN	A felhasználó a szabadságolások oldalon található
		WHEN	A felhasználó az engedélyezés gombra kattint az adott szabadság kérvényen
		THEN	Egy dialógus ablak jelenik meg
3.0.29	Szabadságok engedélyezés dialógus ablak	GIVEN	A felhasználó az engedélyezés gombra megjelent a dialógus ablak
		WHEN	A dialógus ablak igen gombra kattintott
		THEN	Eltűnik a dialógus ablak és a szabadság kérvény a kalendáriumban pedig megjelenik az adott intervallumra a szabadság
3.0.30	Szabadságok engedélyezés dialógus ablak	GIVEN	A felhasználó az engedélyezés gombra megjelent a dialógus ablak
		WHEN	A dialógus ablak nem gombra kattintott
		THEN	Eltűnik a dialógus ablak és nem történik semmi
3.0.31	Engedélyezett szabadságok megtekintése	GIVEN	A felhasználó a szabadságolások oldalon található és a kalendáriumban kiválaszt egy adott napot, ahol csak egy szabadság van
		WHEN	Duplán kattint az adott napra

		THEN	Egy ablakban megjelenik az információ az adott szabadságról
3.0.32	Engedélyezett szabadságok megtekintése	GIVEN	A felhasználó a szabadságolások oldalon található és a kalendáriumban kiválaszt egy adott napot, ahol nincs szabadság
		WHEN	Duplán kattint az adott napra
		THEN	Nem történik semmi
3.0.33	Engedélyezett szabadságok megtekintése	GIVEN	A felhasználó a szabadságolások oldalon található és a kalendáriumban kiválaszt egy adott napot, ahol több szabadság is van
		WHEN	Egyszer kattint az adott napra
		THEN	Megjelenik egy listában az adott napra az összes szabadság



ábra 26: Munkacsoport vezető és beosztott eset diagramja



ábra 27: Adminisztrátor eset diagramja

A 26. ábrán látható munkacsoport vezető és beosztott használati eset diagramja ez a munkacsoportvezető és főállású munkavállaló user story táblázat uniója. Mivel egy felettes lehet beosztott egy másik csoportban

4.0.1	Szerepkörök oldalra navigálás	GIVEN	Kezdőoldalon van a felhasználó
		WHEN	Felhasználók a szerepkörök gombra kattint
		THEN	Szerepkörök oldalra navigál
4.0.2	Szerepkörök törlése	GIVEN	Szerepkörök oldalon van a felhasználó
		WHEN	A x gombra kattint a lista egyik sorában
		THEN	Egy dialógus ablak jelenik meg azzal az üzenettel, hogy Biztosan törölni szeretné a szerepkört?
4.0.3	Szerepkörök törlése	GIVEN	Biztosan törölni szeretné a szerepkört dialógus ablak megjelent
		WHEN	Igen gombra kattintunk
		THEN	Üzenetet kapunk, hogy sikeresen törölted a szerepkört
4.0.4	Szerepkörök törlése	GIVEN	Biztosan törölni szeretné a szerepkört dialógus ablak megjelent

		WHEN	Nem gombra kattintunk
		THEN	Nem történik semmi
4.0.5	Szerepkörök módosítása oldalra navigálás	GIVEN	Felhasználó a szerepkörök oldalon van
		WHEN	A listában a ceruza jelre kattintunk
		THEN	Módosító oldalra navigálunk
4.0.6	Szerepkörök módosítása	GIVEN	Felhasználó a szerepkörök módosító oldalon van
		WHEN	Módosítja a felhasználó a szerepkör nevét és mentés gombra kattint
		THEN	Szerepkörök oldalra navigálunk
4.0.7	Szerepkörök létrehozása	GIVEN	Felhasználó a szerepkörök létrehozás oldalon van
		WHEN	Helyesen kitöltött űrlapot küld be a felhasználó
		THEN	üzenetet kapunk, hogy sikeresen létrehoztad a szerepkört
4.0.8	Szerepkörök felhasználóhoz rendelése	GIVEN	Felhasználó a Szerepkörök felhasználóhoz létrehozás oldalon van
		WHEN	Helyesen kitöltött űrlapot küld be a felhasználó
		THEN	üzenetet kapunk, hogy sikeresen hozzárendelted a felhasználót a szerepkörhöz
4.0.9	Szerepkörök felhasználóhoz rendelése, amelyikhez már hozzátartozik	GIVEN	Felhasználó a Szerepkörök felhasználóhoz létrehozás oldalon van
		WHEN	Helytelen űrlapot küld be a felhasználó
		THEN	üzenetet kapunk, hogy sikertelen volt a hozzárendelés a felhasználó már a szerepkörhöz tartozik
4.0.10	Szerepkörök felhasználóhoz törlés	GIVEN	Felhasználó a Szerepkörök felhasználóhoz oldalon van
		WHEN	A listában az x gombra kattint
		THEN	Megjelenik egy dialógus ablak
4.0.11	Szerepkörök felhasználóhoz törlés	GIVEN	Biztosan törölni szeretné a felhasználót a szerepkörből dialógus ablak megjelent
		WHEN	Igen gombra kattintunk
		THEN	Üzenetet kapunk, hogy sikeresen törölted a felhasználót a csoportból
4.0.12	Szerepkörök felhasználóhoz törlés	GIVEN	Biztosan törölni szeretné a felhasználót a szerepkörből dialógus ablak megjelent
		WHEN	Nem gombra kattintunk

		THEN	Nem történik semmi
4.0.13	Munkacsoport létrehozás	GIVEN	Felhasználó a munkacsoport létrehozás oldalon van
		WHEN	Helyesen kitöltött űrlapot küld be a felhasználó
		THEN	Üzenetet kapunk, hogy sikeresen létrehoztad a munkacsoportot
4.0.14	Munkacsoport törlése	GIVEN	Felhasználó a munkacsoport oldalon van
		WHEN	Listában az adott sorban x gombra kattintunk
		THEN	Megjelenik egy dialógus ablak az alábbi felirattal biztosan törölni szeretné a csoportot?
4.0.15	Munkacsoport törlése	GIVEN	Biztosan törölni szeretné a csoportot dialógus ablak megjelent
		WHEN	Igen gombra kattintunk
		THEN	Üzenetet kapunk, hogy sikeresen törölted a munkacsoportot és eltűnik a listából a csoport
4.0.16	Munkacsoport törlése	GIVEN	Biztosan törölni szeretné a felhasználót a szerepkörből dialógus ablak megjelent
		WHEN	Nem gombra kattintunk
		THEN	Nem történik semmi
4.0.17	Munkacsoport módosítása	GIVEN	Munkacsoport módosítás oldalon van a felhasználó
		WHEN	Helyesen kitöltött űrlapot küld be a felhasználó
		THEN	Megváltoznak a munkacsoport adatai
4.0.18	Munkatárs regisztrációja	GIVEN	Új munkatárs oldalon van a felhasználó
		WHEN	Helyesen kitöltött űrlapot küld be a felhasználó
		THEN	Üzenetet kapunk, hogy sikeresen létrehoztad az új fiókot
4.0.19	Munkatárs regisztrációja	GIVEN	Új munkatárs oldalon van a felhasználó
		WHEN	Helytelen űrlapot küld be a felhasználó
		THEN	Üzenetet kapunk, hogy sikertelen volt a regisztrálás
4.0.20	Belépő kártya létrehozás	GIVEN	Belépő kártya létrehozás oldalon van a felhasználó
		WHEN	Helyes űrlapot küld be a felhasználó és felhasználónak még nincs kártyája
		THEN	Üzenetet kapunk, hogy sikeresen hozzá lett rendelve a kártya a felhasználóhoz

4.0.21	Belépő kártya létrehozás	GIVEN	Belépő kártya létrehozás oldalon van a felhasználó
		WHEN	Helyes űrlapot küld be a felhasználó, de a kiválasztott felhasználónak már van kártyája
		THEN	Üzenetet kapunk, hogy sikertelen a kártya hozzárendelés mivel a felhasználó már rendelkezik kártyával
4.0.22	Belépő kártya létrehozás	GIVEN	Belépő kártya létrehozás oldalon van a felhasználó
		WHEN	Helytelen űrlapot küld be a felhasználó
		THEN	Üzenetet kapunk, hogy sikertelen a kártya hozzárendelés a felhasználóhoz
4.0.23	Belépő kártya törlés	GIVEN	Belépő kártyák oldalon van a felhasználó
		WHEN	A kiválasztott sorban a listában az x gombra kattintunk
		THEN	Megjelenik egy dialógus ablak az alábbi felirattal biztosan törölni szeretné a kártyát?
4.0.24	Belépő kártya törlés	GIVEN	Dialógus ablak megjelent
		WHEN	Nem gombra kattintunk
		THEN	Nem történik semmi
4.0.25	Felhasználók törlése a munkacsoportból	GIVEN	Felhasználók a munkacsoportban oldalon vagyunk
		WHEN	A kiválasztott sorban a listában az x gombra kattintunk
		THEN	Megjelenik egy dialógus ablak az alábbi felirattal biztosan törölni szeretnéd a dolgozót a csoportból?
4.0.26	Felhasználók törlése a munkacsoportból	GIVEN	Dialógus ablak megjelent
		WHEN	Igen gombra kattintunk
		THEN	Eltűnik a listából a felhasználó és üzenetet kapunk, hogy sikeresen törölted a felhasználót a munkacsoportból
4.0.27	Felhasználók törlése a munkacsoportból	GIVEN	Dialógus ablak megjelent
		WHEN	Nem gombra kattintunk
		THEN	Nem történik semmi

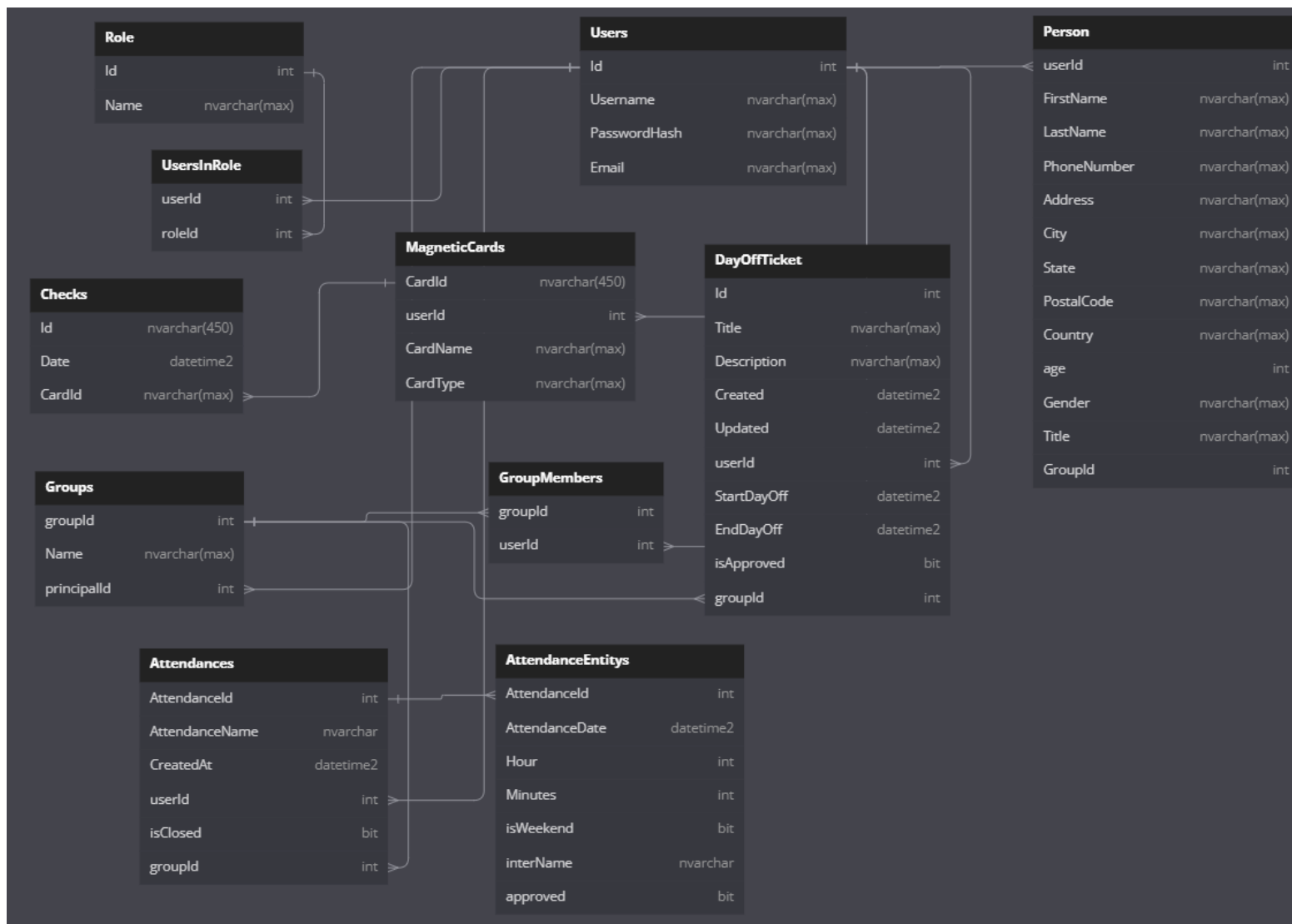
### 3.3 Adatbázis táblák

A 28. ábrán láthatóak az adatbázis táblák és kapcsolatai, amik az alábbi táblák:

- Role
- UsersInRole
- Users
- Person
- Checks
- MagneticCards
- DayOffTickets
- Groups
- GroupMembers
- Attendances
- AttendanceEntitys

A fentebb felsorolt adatbázis táblák Code First megközelítéssel lettek létrehozva az Entity Framework segítségével. A Code First megközelítésben az adatbázis struktúrája a modell osztályok alapján kerül létrehozásra. Ez azt jelenti, hogy a táblák, oszlopok és kapcsolatok definíciója az osztályok és azok tulajdonságai alapján történik.





ábra 28: Adatbázis táblák és kapcsolatai

### A főbb táblák oszlopainak rövid leírása

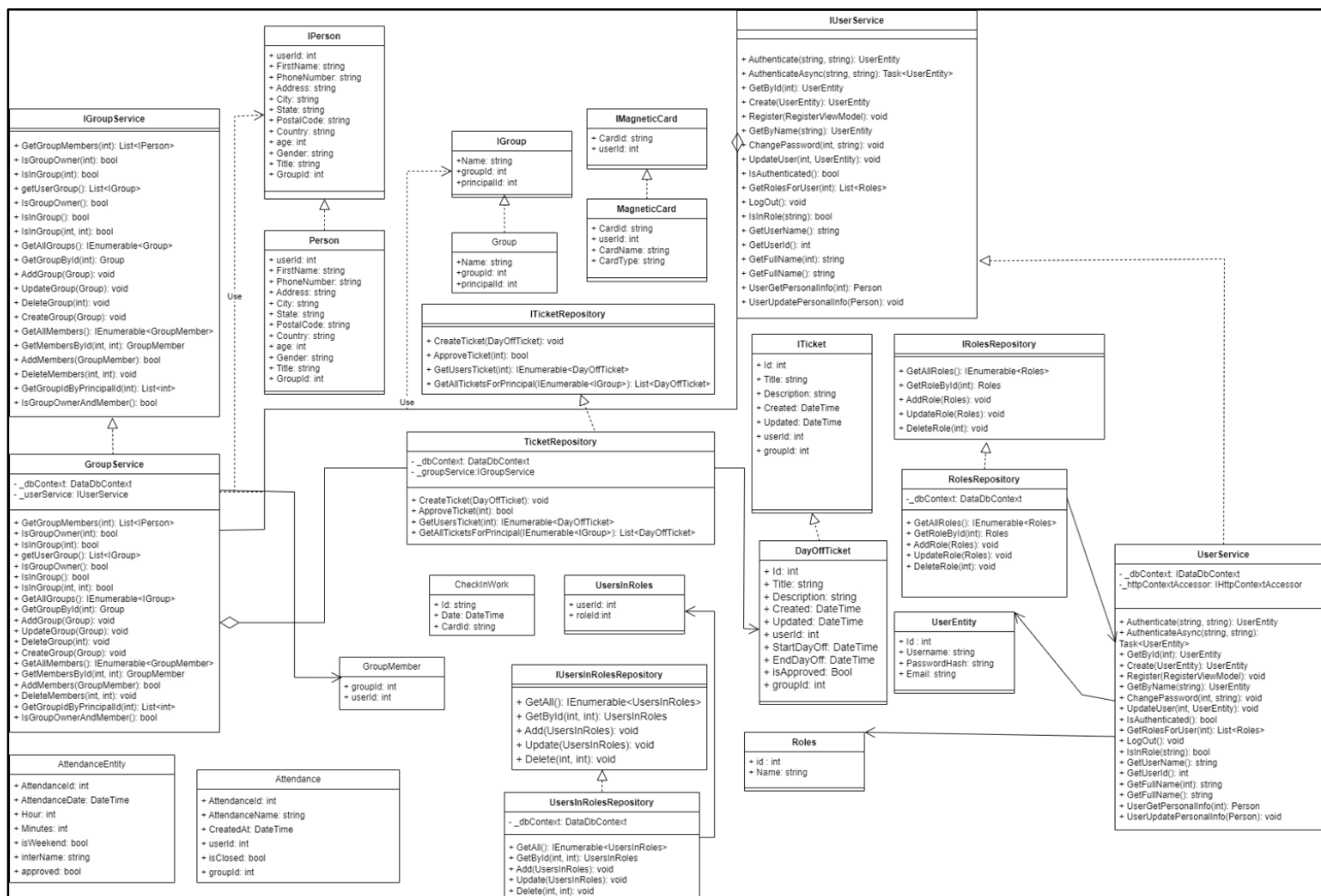
- Users - ebben a táblában tárolom a felhasználó bejelentkezéséhez szükséges információkat
  - Id - a felhasználó egyedi azonosítója
  - Username - a felhasználó neve, amivel bejelentkezik a weboldalra
  - PasswordHash – a felhasználó titkosított jelszava
  - Email – a felhasználó emailcíme
- Groups – ebben a táblában tárolom a létező munkacsoportokat
  - groupId – a munkacsoport egyedi azonosítója
  - Name – a munkacsoport neve
  - principalId – a munkacsoport vezetőjének egyedi azonosítója
- Role – ebben a táblában tárolom a létező szerepeköröket
  - Id – egyedi azonosítója a szerepkörnek
  - Name – az adott szerepkör neve
- UsersInRole – ez egy összerendelő tábla, amiben a felhasználókat szerepkörhöz rendelem
  - userId – a felhasználó egyedi azonosítója
  - roleId – a szerepkör egyedi azonosítója
- GroupMembers – ez egy összerendelő tábla, amiben a felhasználókat a munkacsoportokhoz rendelem
  - groupId – a munkacsoport egyedi azonosítója
  - userId – a felhasználó egyedi azonosítója
- Checks – ebben a táblában található minden be és ki jelentkezés az irodából
  - Id – egyedi azonosító az adott
  - Date – pontos dátum amikor a kártyaolvasóhoz érintettük a kártyát
  - CardId – a kártya azonosítója
- MagneticCard – ebben a táblában találhatóak meg a beregisztrált kártyák
  - CardId – egyedi azonosítója a kártyának
  - userId – a kártyához tartozó felhasználó egyedi azonosítója
  - CardName – kártya neve
  - CardType – kártya típusa

## 3.4 UML osztálydiagram

A 29. ábrán láthatóak a főbb osztályok UML diagramja. A program alapját az alábbi 4 osztály adja:

- UserService
- GroupService
- RoleRepository
- UsersInRoleRepository

Ezeket nélkül nem volna lehetséges a felhasználó hitelesítése, felhasználó szereposztása, illetve a csoport hierarchia.



ábra 29: Osztályok UML diagramja

### 3.4.1 UserService

A **UserService** osztály az *IUserService* interfészt implementálja. Az osztály felelős a felhasználókhoz kapcsolódó műveletek végrehajtásáért, mint például a felhasználónév lekérése, bejelentkezés, jelszóváltás, felhasználó adatainak frissítése.

- **GetFullName** metódus visszaadja a bejelentkezett felhasználó teljes nevét. Ehhez megvizsgálja, hogy a felhasználó be van-e jelentkezve (**IsAuthenticated** metódus), majd lekéri a felhasználó azonosítóját (**GetUserId** metódus) és a Person táblából megkeresi a megfelelő rekordot, majd visszaadja a kereszt- és vezetéknévét.
- **IsInRole** metódus ellenőrzi, hogy a bejelentkezett felhasználó rendelkezik-e a megadott szereppel. Ehhez is megvizsgálja, hogy a felhasználó be van-e jelentkezve (**IsAuthenticated** metódus), majd lekéri a felhasználó azonosítóját (**GetUserId** metódus) és lekérdezi a felhasználó szerepeinek neveit a UsersInRole és Role táblákból. Visszaadja, hogy a felhasználó szerepei tartalmazzák-e a megadott szerepet.
- **IsAuthenticated** metódus ellenőrzi, hogy a felhasználó be van-e jelentkezve. Ehhez megvizsgálja, hogy a UserId munkamenet kulcs létezik-e az HttpContextAccessor.HttpContext.Session objektumban.
- **ChangePassword** metódus frissíti a felhasználó jelszavát a megadott új jelszóval. Ehhez lekéri a felhasználót az azonosító alapján (**GetById** metódus), majd a PasswordHash tulajdonságot frissíti az újonnan titkosított jelszóval.
- **UpdateUser** metódus frissíti a felhasználó adatait a megadott frissített felhasználó objektum alapján. Ehhez lekéri a felhasználót az azonosító alapján (**GetById** metódus), majd frissíti a felhasználónévét és az e-mail címet.
- **AuthenticateAsync** és **Authenticate** metódusok bejelentkeztetik a felhasználót az egyik szinkronban a másik aszinkron bejelentkeztetés
- **GetById** metódus a felhasználó azonosítója alapján lekéri és visszaadja a megfelelő felhasználót az adatbázisból
- **GetUserId** metódus visszaadja a bejelentkezett felhasználó azonosítóját. Először ellenőrzi, hogy a felhasználó be van-e jelentkezve (**IsAuthenticated** metódus)

### 3.4.2 GroupService

A **GroupService** osztály implementálja az *IGroupService* interfészt és feladata a csoportokkal és csoporttagokkal kapcsolatos műveletek végrehajtása. Ez az osztály felelős a csoportok létrehozásáért, csoporttagok lekérdezéséért és a csoportvezetők azonosításáért.

- **GetGroupMembers:** Ez a metódus lekéri és visszaadja a megadott csoport tagjait. A csoporttagokat az adatbázisból olvassa ki a groupId alapján, majd visszaadja azokat az IPerson listában.
- **IsInGroup:** Ez a metódus ellenőrzi, hogy egy adott felhasználó (userid) a megadott csoport (groupId) tagja-e. Az adatbázisban megvizsgálja, hogy a felhasználó és a csoport azonosító páros szerepel-e a GroupMembers táblában.
- **getUserGroup:** Ez a metódus lekéri és visszaadja az aktuális felhasználóhoz tartozó csoportokat. Először ellenőrzi, hogy a felhasználó be van-e jelentkezve (**IsAuthenticated** metódus), majd az adatbázisból lekéri a felhasználóhoz tartozó csoporttagságokat, majd visszaadja azokat az IGroup listában
- **IsGroupOwner** Ez a metódus ellenőrzi, hogy a felhasználó, aki be van jelentkezve, tulajdonosa-e a megadott csoportnak (groupId). Az adatbázisban megvizsgálja, hogy a csoport tulajdonosának (principalId) azonosítója megegyezik-e a bejelentkezett felhasználó azonosítójával.
- **AddGroup:** Ez a metódus hozzáad egy új csoportot az adatbázishoz.

### 3.4.3 RoleRepository

Az **RolesRepository** osztály az *IRolesRepository* interfészt implementálja. Ez az osztály felelős a szerepekkel kapcsolatos műveletek végrehajtásáért, mint például az összes szerep lekérdezése, szerep azonosítása azonosító alapján, szerep hozzáadása, szerep frissítése és szerep törlése.

- **GetAllRoles:** Ez a metódus visszaadja az összes szerepet, amelyek tárolva vannak az adatbázisban.
- **GetRoleById:** Ez a metódus azonosító alapján keresi és visszaadja a megfelelő szerepet az adatbázisból.
- **AddRole:** Ez a metódus hozzáad egy új szerepet az adatbázishoz. A role paraméter egy Roles objektum, amelyet hozzá kívánunk adni az adatbázishoz.

- **UpdateRole:** Ez a metódus frissíti egy meglévő szerep adatait az adatbázisban.
- **DeleteRole:** Ez a metódus törli a megadott azonosítóval rendelkező szerepet az adatbázisból.

### 3.4.4 UsersInRoleRepository

A **UsersInRolesRepository** osztály az *IUsersInRolesRepository* interfészt implementálja. Az osztály lehetővé teszi a felhasználók és szerepek közötti kapcsolatok kezelését az alkalmazásban, beleértve az új szerepkapcsolatok hozzáadását, meglévő szerepkapcsolatok lekérdezését, frissítését és törlését.

- **GetAll:** Ez a metódus visszaadja az összes felhasználót és szerepet tartalmazó rekordot az adatbázisból.
- **GetById:** Ez a metódus az adott szerep azonosítója és felhasználó azonosítója alapján keresi és visszaadja a megfelelő UsersInRoles rekordot az adatbázisból. A `roleId` és `userId` paraméterek meghatározzák a keresett rekord azonosítóit. Ha a rekord nem található, akkor null értéket ad vissza.
- **Add:** Ez a metódus hozzáad egy új rekordot a felhasználók és szerepek közötti kapcsolatok táblájához az adatbázisban.
- **Delete:** Ez a metódus törli a megadott szerep és felhasználó közötti kapcsolatot az adatbázisból. A `roleId` és `userId` paraméterek azonosítják a törlendő rekordot.

### 3.4.5 Kártya olvasó

Az arduino fejlesztői környezetében két fő függvényt használunk:

- A **setup** függvény az Arduino programjának kezdőállapotát konfigurálja. Ez a függvény csak egyszer fut le, amikor az Arduino bekapcsol vagy újraindul. Itt konfigurálom be a programomban a hálózati csatlakozáshoz, illetve a kártya olvasáshoz szükséges dolgokat.
- A **loop** függvény az Arduino programjának fő ciklusát tartalmazza. Ez a függvény folyamatosan ismétlődik, miután a **setup** függvény egyszer lefutott. Ebben a függvényben mindig meghívom a **readRFID** függvényt

A kártya olvasást több komponensre szedtem így átláthatóbb, hogy pontosan melyik függvény mit csinál:

- A **readRFID** függvény az RFID olvasóval kommunikál, azért felelős, hogy a kártyák sorozatszámát beolvassa, észrevegye, hogy új kártyát érintettek a beolvasóhoz, illetve a memóriájába elmentse a kártyasorozatszámát, amit tovább adjon a **ConvertByteToDec** függvénynek.
- A **ConvertByteToDec** nevű függvény arra szolgál, hogy átalakítsa a byte tömbben tárolt adatokat decimális számokká, majd át adja a **callApi** függvénynek szöveggént.
- A **callApi** függvény egy api hívást indít a megkapott kártyaazonosítóval a programom belül bekonfigurált web elérési helyre.

## 3.5 Alkalmazás részei

Lentebb látható a főbb könyvtárak rövid leírása:

- **Attributes:** ebben a könyvtárban találhatóak az általam írt egyéni attribútumok. Két fő attribútum található meg.
  - **Authenticated:** attribútum, amely azért felel, hogy megnézzé, hogy a felhasználó be van-e jelentkezve, ha nincs akkor a bejelentkező oldalra dobja.
  - **CustomRole:** attribútum, amely egy dinamikus attribútum, a zárojelek közé megadható több szerepkör neve, és ha felhasználó nem rendelkezik az adott szerepkörrel akkor egy hibaoldalra navigálja.
- **Controllers:** ebben a könyvtárban találhatóak meg a kontrollerek
  - **Api:** ebben a könyvtárban az api kontrollerek találhatóak meg
- **Data:** az alkalmazás adatelérési rétegével (data access layer) kapcsolatos fájlokat és osztályokat tartalmazza
- **Migrations:** ebben a könyvtárban az Entity Framework által készített adatbázis sémák találhatóak meg. Amelyek segítségével C#-os osztályokból adatbázis táblákat hozunk létre.
- **Models:** tartalmazza az alkalmazás modelljének (adatstruktúráinak) osztályait
  - **Interface** itt találhatóak meg interfész modellek
  - **Model** itt találhatóak meg az implementált osztályok



- **Views:** a webes felhasználói felületünk különböző oldalainak (view-knak) fájljait tartalmazza

## 3.6 Tesztelés

A program teszteléshez a Xunit tesztelő keretrendszert használtam, mivel ez könnyen integrálható az ASP.NET projectekhez. A tesztelés során a legfőbb osztályokat teszteltem, mivel egy összetett programról beszélünk ezért több fajta tesztet is készítettem, megtalálható unit tesztek, integrációs teszt, api tesztesetek, illetve kézi tesztek is mivel a fizikai kártya olvasót is teszteltem. A lentebb látható táblázatban találhatók meg a tesztek bővebb leírása.

Teszt neve	Teszt leírása	Teszt típusa	Teszt elvárt eredménye	Teszt eredménye
Integration_GroupServiceTest	GroupService osztály tesztelése, mókolt UserService osztállyal együtt	Integrációs teszt	A GroupService metódusainak tesztelése attól függően, hogy a felhasználó milyen jogosultsággal rendelkezik	Sikeres
Api_Get_ReturnsListOfGroups	A ListGroupController az adatbázisban lévő csoportokat (Group) listázzuk	Api teszt	Meghívjuk a ListGroupController-t, amikor bevigyunk bejelentkezve az összes csoportot meg kell kapnunk	Sikeres
Api_Get_ReturnsListOfRoles	A ListRoleController meghívásakor vissza kell, hogy adjon az adatbázisban lévő szerepeket, ha bevigyunk jelentkezve	Api teszt	Meghívjuk a ListRoleController-t, amikor bevigyunk bejelentkezve az összes szerepkört meg kell kapnunk	Sikeres
Api_Get_ReturnsListOfRolesNotAuthenticated	A ListRoleController meghívásakor nem bejelentkezett felhasználónak semmilyen adatot nem kell, hogy visszaadjon	Api teszt	Meghívjuk a ListRoleController-t, amikor nem vagyunk bejelentkezve semmilyen adatot nem kell kapnunk	Sikeres
Unit_IsGroupOwner_UsingGroupService	A GroupService IsGroupOwner egy igaz hamis értéket ad vissza attól függően, hogy a felhasználó munkacsoportvezető-e	Unit teszt	Munkacsoportvezetők vagyunk igaz értéket kapunk	Sikeres

Unit_IsAuthenticated_ReturnsFalse_WhenSessionIsNull	A UserService IsAuthenticated metódusa egy igaz hamis értéket ad vissza attól függően, hogy bevan a jelentkezve a felhasználó vagy sem	Unit teszt	Ha nem található süti, akkor hamis értéket kapunk	Sikeres
Unit_Create_AddsUserToDatabase	A UserService Create metódusa létrehoz egy felhasználót az adatbázisban	Unit teszt	Adatbázisban megjelenik az új felhasználó	Sikeres
Unit_GetUserName_ReturnsNull_WhenNotAuthenticated	A UserService GetUserName metódusa visszaadja a felhasználó felhasználónevét	Unit teszt	Semmit nem ad vissza, ha nincs bejelentkezve a felhasználó	Sikeres
Unit_RolesTest_WhenAuthenticated	A UserService GetRoles metódusa visszaadja a felhasználó szerepkörét	Unit teszt	Felhasználó szerepkörét kapjuk meg amihez hozzá van rendelve	Sikeres
Unit_IsInRole_ReturnsFalse_WhenUserIsNotInRole	A UserService IsInRole metódusa visszaad egy igaz hamis értéket attól függően, hogy a felhasználó benne van -e a szerepkörben	Unit teszt	Felhasználó nem rendelkezik a megfelelő szerepkörrel	Sikeres
Unit_IsAuthenticated_ReturnsTrue_WhenLoggedIn	A UserService IsAuthenticated metódusa egy igaz hamis értéket ad vissza attól függően, hogy bevan a jelentkezve a felhasználó vagy sem	Unit teszt	Igaz értékkel térünk vissza, mivel be vagyunk jelentkezve	Sikeres
Unit_LogOut_ClearsSession	A UserService LogOut metódusa kijelentkezteti a felhasználót	Unit teszt	Kijelentkezünk	Sikeres
Unit_GetUserId_ReturnsCorrectUserId_WhenAuthenticated	A UserService GetUserId metódusa visszaadja a felhasználó azonosítóját	Unit teszt	Felhasználó azonosítóját kapjuk vissza helyesen	Sikeres
Unit_GetFullName_ReturnsCorrectName_WhenUserExists	A UserService GetFullName metódusa visszaadja a felhasználó teljes nevét	Unit teszt	Felhasználó teljes nevét kapjuk vissza, ha bevan jelentkezve	Sikeres

Unit_GetFullName_WhenNotAuthenticated	A UserService Authenticate bejelentkezeti felhasználót	Unit teszt	Semmit nem kapunk vissza, mivel a felhasználó nincs bejelentkezve	Sikeres
Unit_AddRoleTest	A RoleRepository osztály Add metódusa arra szolgál, hogy egy Role Entitást adjon az adatbázishoz	Unit teszt	Ha meghívjuk az Add metódust, aminek átadunk egy Role entitást az adatbázisba bekerül a Role entitás	Sikeres
Unit_DeleteRoleTest	A RoleRepository osztály Delete metódusa arra szolgál, hogy egy Role Entitást töröljön az adatbázisból	Unit teszt	Ha meghívjuk az Delete metódust, aminek átadunk egy Role azonosítót az adatbázisból törölve lesz az adott azonosítójú entitás	Sikeres
Unit_GetAllRolesTest	A RoleRepository osztály GetAllRoles metódusa arra szolgál, hogy egy Role Entitást töröljön az adatbázisból	Unit teszt	Ha meghívjuk az GetAllRoles metódust az adatbázisban lévő összes Role entitást visszkapjuk	Sikeres
Unit_UpdateRoleTest	A RoleRepository osztály UpdateRole metódusa arra szolgál, hogy egy Role Entitást módosítson az adatbázisban	Unit teszt	Ha meghívjuk az Update metódust az adatbázisban egy Role Entitással akkor módosítjuk a meglévő entitást az adatbázisban	Sikeres
Unit_AddUserInRoleRepository	A UsersInRoleRepository osztály Add metódusa arra szolgál, hogy egy UsersInRole Entitást hozzon létre az adatbázisban	Unit teszt	Az Add metódusnak átadunk egy UsersInRole entitást akkor lementi az adatbázisba	Sikeres
Unit_DeleteUsersInRoleRepository	A UsersInRoleRepository osztály Delete metódusa arra szolgál, hogy egy UsersInRole Entitást hozzon töröljön az adatbázisban	Unit teszt	Az Delete metódusnak átadunk két azonosítót, akkor kitörli az adatbázisból az adott UsersInRole azonosítójú entitást	Sikeres
Unit_GetByIdUsersInRoleRepository	A UsersInRoleRepository osztály GetById metódusa arra szolgál, hogy egy UsersInRole Entitást kérjen le az adatbázisban	Unit teszt	Az GetById metódusnak átadunk két azonosítót, és lekéri a UsersInRole adatbázisból az adott azonosítójú elemet	Sikeres

Nem regisztrált kártya, helytelen api kulcs megadásával	Rosszul konfigurált kártyaolvasóhoz érintünk egy nem regisztrált kártyát	Kézi teszt	Nem történik semmit, a kártyaolvasón BadRequest szöveget kapunk vissza	Sikeres
Nem regisztrált kártya, helyes api kulcs megadásával	Jól konfigurált kártyaolvasóhoz érintünk egy nem regisztrált kártyát	Kézi teszt	Nem történik semmit, a kártyaolvasón Invalid Card szöveget kapunk vissza	Sikeres
Beregisztrált kártya, helyes api kulcs megadásával	Jól konfigurált kártyaolvasóhoz érintünk egy regisztrált kártyát	Kézi teszt	Az adatbázisban létrejön egy Check entitás és Valid Card üzenetet kapunk vissza	Sikeres
Beregisztrált kártya, helytelen api kulcs megadásával	Rosszul konfigurált kártyaolvasóhoz érintünk egy regisztrált kártyát	Kézi teszt	Nem történik semmit, a kártyaolvasón BadRequest szöveget kapunk vissza	Sikeres

### 3.7 Fejlesztési lehetőség

Véleményem szerint a program kis és közép vállalatok számára már így is hasznos lehet kimutatás szempontjából, illetve gyorsabb adminisztrációk ügyintézésben. Az elkészült funkciókon felül még az alábbi funkciókkal lehetne fejleszteni az alkalmazást.

- Egy blog felületet, ahol a munkáltató az újdonságokat kitudná publikálni
- A dolgozói hierarchia megtekintése
- Üzenet küldésre alkalmas platform kialakítása
- SMTP szerver készítése.

Ezek a funkciók mind implementálhatóak az alkalmazásba, mivel könnyen skálázható és nagyon modulárisra terveztem, de sajnos a szakdolgozatom belül ezekre az idő hiányában nem adódott lehetőségem.

## 4.Hivatkozások

- [1] Adatbázis táblák és kapcsolatai diagramhoz használt szerkesztő, [Online].  
Elérhető: <https://dbdiagram.io/>.  
[Hozzáférés dátuma: 2023. 05. 15.]
- [2] UML diagram szerkesztő, [Online].  
Elérhető: <https://app.diagrams.net/>.  
[Hozzáférés dátuma: 2023. 05. 15.]
- [3] Ábrák az alkalmazásban, [Online].  
Elérhető: <https://undraw.co/illustrations> .  
[Hozzáférés dátuma: 2023. 05. 15.]
- [4] Űrlapok, táblák, Ikonokhoz használt könyvtár, [Online].  
Elérhető: <https://js.devexpress.com/>  
[Hozzáférés dátuma: 2023. 05. 15.]
- [5] Üzenetek megjelenítésére használt könyvtár, [Online].  
Elérhető: <https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/toastr.min.js>.  
[Hozzáférés dátuma: 2023. 05. 15.]
- [6] Betűtípushoz használt könyvtár , [Online].  
Elérhető: <https://fonts.googleapis.com/css2?family=Roboto>.  
[Hozzáférés dátuma: 2023. 05. 15.]
- [7] Felhasználói kinézethez használt könyvtár , [Online].  
Elérhető: <https://getbootstrap.com/>.  
[Hozzáférés dátuma: 2023. 05. 15.]
- [8] Kártya olvasó kép [Online]  
Elérhető: <https://www.aranacorp.com/en/using-an-rfid-module-with-an-esp32/>  
[Hozzáférés dátuma: 2023. 05. 15 ]
- [9] Mikrokontroller fejlesztői könyvtár [Online]  
Elérhető: <https://www.platformio.org/>  
[Hozzáférés dátuma: 2023. 05. 15 ]
- [10] Kártya olvasóhoz használt könyvtár JSON konvertálás [Online]

- Elérhető: <https://arduinojson.org/>  
[Hozzáférés dátuma: 2023. 05. 15 ]
- [11] RFID kártya olvasáshoz használt könyvtár [Online]  
Elérhető: <https://github.com/miguelbalboa/rfid.git>  
[Hozzáférés dátuma: 2023. 05. 15 ]
- [12] HTTP kérés indításhoz használt könyvtár [Online]  
Elérhető: <https://github.com/espressif/arduino-esp32/blob/master/libraries/HTTPClient/src/HTTPClient.cpp>  
[Hozzáférés dátuma: 2023. 05. 15 ]
- [13] Internet kapcsolat létesítéséhez használt könyvtár [Online]  
Elérhető: <https://github.com/arduino-libraries/WiFi>  
[Hozzáférés dátuma: 2023. 05. 15 ]
- [14] SPI kommunikációhoz használt könyvtár [Online]  
Elérhető: <https://reference.arduino.cc/reference/en/language/functions/communication/spi/>  
[Hozzáférés dátuma: 2023. 05. 15 ]
- [15] Titkosításhoz használt könyvtár [Online]  
Elérhető: <https://github.com/neoKushan/BCrypt.Net-Core>  
[Hozzáférés dátuma: 2023. 05. 15 ]
- [16] Teszteléshez használt könyvtár [Online]  
Elérhető: <https://github.com/xunit/visualstudio.xunit>  
[Hozzáférés dátuma: 2023. 05. 15 ]
- [17] Mókoláshoz használt könyvtár [Online]  
Elérhető: <https://github.com/moq/moq4>  
[Hozzáférés dátuma: 2023. 05. 15 ]
- [18] Láblécben látható kép [Online]  
Elérhető: <https://www.shapedivider.app/>  
[Hozzáférés dátuma: 2023. 05. 15 ]

## Szakirodalom

- [19] Arduino Esp32 alapjai [Online]  
Elérhető:<https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>  
[Hozzáférés dátuma: 2023. 05. 15 ]
- [20] ASP.NET Core alapjai [Online]  
Elérhető:<https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-7.0>  
[Hozzáférés dátuma: 2023. 05. 15 ]
- [21] MS SQL alapjai [Online]  
Elérhető:<https://learn.microsoft.com/en-us/sql/sql-server/tutorials-for-sql-server-2016?view=sql-server-ver16>  
[Hozzáférés dátuma: 2023. 05. 15 ]

## Mellékletek

A mellékelt tömörített fájl „ZSURAVSZKIJ\_TAMÁS\_CAALUO\_SZAKDOLGOZAT.zip” tartalma:

- „Recon.zip”: A webalkalmazás futtatható példánya (saját számítógépre konfigurálva).
- „Recon-develop”: A webalkalmazás projekt mappája
- „Card-Reader”: A kártya olvasó projekt mappája
- „Data-Migration”: A telepítéshez szükséges adat migrációs szkriptfájl
- „Témabejelentő.pdf”: A témabejelentő
- „Eredetiség nyilatkozat.doc” Az eredetiségi nyilatkozatom
- „Szakdolgozat Dokumentáció.pdf” A szakdolgozat dokumentációja PDF formátumban
- „Szakdolgozat Dokumentáció.docx” A szakdolgozat dokumentációja Word dokumentum formátumban