



EÖTVÖS LORÁND TUDOMÁNYEGYETEM
INFORMATIKAI KAR
PROGRAMOZÁSELMÉLET ÉS SZOFTVERTECHNOLÓGIAI
TANSZÉK

Vállalati munkaerő nyilvántartó alkalmazás

Témavezető:

Varga Dániel
egyetemi adjunktus, PhD

Szerző:

Buchsbaum Miklós
programtervező informatikus BSc

Budapest, 2025

Tartalomjegyzék

1. Bevezetés	5
2. Felhasználói dokumentáció	6
2.1. Az alkalmazás rövid ismertetése	6
2.2. Rendszerkövetelmények	7
2.3. Telepítés	9
2.3.1. Előzetes lépések	9
2.3.2. Konfiguráció beállítása	9
2.3.3. Az alkalmazás kihelyezése	12
2.4. Az alkalmazás általános használata	14
2.4.1. A felület	14
2.4.1.1. Általános felhasználói felület	15
2.4.1.2. Owner felület	16
2.4.2. Visszajelzések	17
2.4.2.1. Mezővalidáció	18
2.4.2.2. Műveletek megerősítése	19
2.4.2.3. Felhasználói fiók törlése	19
2.4.2.4. Értesítések a rendszerben	20
2.4.2.5. Email értesítések	20
2.5. Általános felhasználói felület	21
2.5.1. Naptár funkciók	21
2.5.1.1. Beosztás megtekintése	21
2.5.1.2. Beosztás kezelése	25
2.5.1.3. Összevont beosztás	28
2.5.2. Felhasználók kezelése	30
2.5.3. Felhasználók kezelése	30
2.5.3.1. Szerepkörök és jogosultságok	30

TARTALOMJEGYZÉK

2.5.3.2. Felhasználói felületkezelés és interakciók	31
2.5.3.3. Felhasználói profilok kezelése	32
2.5.3.4. Szabadságnapok kezelése	34
2.5.4. Kimutatások	35
2.5.5. Projektek	40
2.5.6. Felhasználói profil	47
2.5.7. Csevegés Funkció	49
2.5.7.1. Áttekintés	49
2.5.7.2. Csevegés Felület Felépítése	49
2.5.7.3. Egyéni Beszélgetések	50
2.5.7.4. Csoportos Beszélgetések	51
2.5.7.5. Online Állapot Jelzése	52
2.5.8. Munkaidő Nyilvántartás	53
2.5.8.1. Felület Felépítése	53
2.5.8.2. Személyes Munkaidő Követés	53
2.5.8.3. Csapat Munkaidő Áttekintés	53
2.5.8.4. Részletes Felhasználói Nézet	55
2.5.8.5. Munkaidő Szerkesztés	55
2.5.9. Videóhívás Rendszer	55
2.5.9.1. Felület Felépítése	56
2.5.9.2. Fő Funkcionalitás	56
2.5.9.3. Értesítések és Kapcsolatkezelés	57
2.6. Owner felület	58
2.6.1. Irányítópult (Dashboard)	59
2.6.2. Vállalatok Kezelése	59
2.6.3. Vállalati Részletek	60
2.6.4. Licenckulcsok Kezelése	60
2.6.5. Tevékenységnapló	61
3. Fejlesztői dokumentáció	64
3.1. A megoldandó feladat	64
3.1.1. A felhasználói felületre vonatkozó elvárások	64
3.1.2. A szerveroldali komponensre vonatkozó elvárások	65
3.2. Fejlesztői környezet	65

3.3.	Az alkalmazás rétegei és felelősségi köreik	66
3.3.1.	A modell réteg (Model Layer)	67
3.3.2.	Az adatkezelési réteg (Data Access Layer)	68
3.3.3.	A szolgáltatási réteg (Service Layer)	68
3.3.4.	A prezentációs réteg (Presentation Layer)	69
3.4.	Perzisztencia	70
3.4.1.	Adatkezelési réteg	70
3.4.2.	Entity Framework Core használata	71
3.4.3.	Adatbázis konfiguráció	71
3.4.4.	Migrációk	72
3.4.5.	Kapcsolatok és kezelésük	72
3.4.6.	Lekérdezési optimalizáció	74
3.5.	Autentikáció	74
3.5.1.	Felhasználói fiókok és szerepkörök	74
3.5.2.	Jogosultságkezelés	75
3.5.3.	Hitelesítési folyamat	75
3.6.	Parancsok és lekérdezések	75
3.6.1.	Vezérlő-alapú kéréskezelés	76
3.6.2.	Hibakezelés és eredmények kezelése	76
3.6.3.	DTO-modellek használata	76
3.6.4.	Az MVC architektúra alkalmazása	77
3.6.5.	Aszinkron műveletek	78
3.6.6.	HTTP kérések és válaszok	79
3.6.7.	Kérések formátuma	79
3.6.8.	Válaszok formátuma	80
3.6.9.	Validáció és hibaüzenetek	80
3.6.10.	Valós idejű kommunikáció	80
3.7.	A szerveroldal belépési pontja	82
3.7.1.	Vezérlők	83
3.7.2.	HTTP kérések és válaszok felépítése	88
3.8.	Kliens	89
3.8.1.	A felület tervezése	89
3.8.2.	Állapotkezelés	90
3.8.3.	Oldalak	90

TARTALOMJEGYZÉK

3.8.4. Közös komponensek	91
3.8.5. HTTP kérések kezelése	91
3.9. Tesztelés	92
3.9.1. Automatikus egységtesztek	92
3.9.2. Alkalmazás manuális tesztelése	93
4. Összegzés	95
Köszönnetnyilvánítás	97
Irodalomjegyzék	97
Ábrajegyzék	98

1. fejezet

Bevezetés

A szoftverfejlesztés során a tervezési szakasz kulcsfontosságú szerepet játszik. Mivel egy alkalmazás életciklusa során jelentős időt és energiát igényel annak karbantartása és továbbfejlesztése, elengedhetetlen, hogy már a kezdeti fázisban gondosan ügyeljünk a modularitásra és a bővíthetőségre. Különösen igaz ez nagyobb rendszerek esetében, ahol több fejlesztő együttműködése szükséges – ilyenkor még fontosabbá válik az átlátható, jól szervezett architektúra.

Az utóbbi években egyre nagyobb figyelmet kap a clean architecture alapelvein nyugvó alkalmazásfejlesztés, ami számomra is felkeltette az érdeklődést. Ez az architektúra útmutatást ad arra vonatkozóan, hogyan érdemes egy összetettebb rendszert komponensekre bontani, különféle felelősségi körök mentén elkülönítve az egyes rétegeket. Ennek eredményeképpen könnyebben karbantartható, tesztelhető, és újrahasznosítható kód jön létre. A clean architecture elvei különösen népszerűek az ASP.NET alapú alkalmazások fejlesztésekor.

Mivel régóta szerettem volna megismерkedni az ASP.NET keretrendszerrel, illetve érdekelt, hogyan lehet modern, objektumorientált szemléletben fejleszteni a .NET ökoszisztemában C# nyelven, úgy döntöttem, hogy ezt a technológiát választom szakdolgozatom alapjául. A frontend fejlesztéséhez az APN.NET által támogatott Razor view-kat használom, ami jól illeszkedik az ASP.NET backendhez.

Szakdolgozatom témája egy olyan webalkalmazás tervezése és megvalósítása, amely a vállalaton belüli munkaerő nyilvántartásával, kezelésével és projektmenedzsment feladatokkal foglalkozik. A rendszer kialakításánál a clean architecture alapelveit követem, hogy a végeredmény egy jól strukturált, rugalmasan bővíthető alkalmazás legyen.

2. fejezet

Felhasználói dokumentáció

2.1. Az alkalmazás rövid ismertetése

A **Managly** egy komplex, modulárisan felépített webalkalmazás, amely vállalatirányítási és projektmenedzsment feladatokat lát el. A rendszer célja, hogy hatékonyan támogassa a közepes és nagyvállalatok működését azáltal, hogy átlátható felületet biztosít a felhasználók, projektek, feladatok és munkaidő kezelésére.

Az alkalmazás ASP.NET Core MVC keretrendszerre épül, modern front-end technológiák (pl. jQuery, Bootstrap, Chart.js) és valós idejű kommunikációs megoldások (SignalR) felhasználásával. A backend és frontend technológiák együttesen biztosítják a rendszer reszponzív, felhasználóbarát működését.

A rendszer többféle szerepkör kezelésére képes: a felhasználók hozzáférési szintjeik alapján különböző jogosultságokat kapnak (pl. *Owner*, *Admin*, *Manager*, *Employee*), így biztosítva a vállalati adatok védelmét és a megfelelő munkafolyamatokat.

A **Managly** különböző modulokból épül fel, melyek közül a legfontosabbak:

- **Projektmenedzsment modul:** Lehetővé teszi új projektek létrehozását, feladatak kezelését, csapattagok hozzárendelését, valamint a projektekhez kapcsolódó aktivitások és időráfordítások nyomon követését.
- **Munkaidő-nyilvántartás:** A felhasználók be- és kijelentkezése, szabadságok kezelése, illetve ledolgozott munkaórák és túlórák elemzése.
- **Jelentéskészítés és analitika:** Részletes kimutatások, vizualizációk és exportálható riportok támogatják a vezetői döntéshozatalt.

- **Kommunikációs modul:** Valós idejű üzenetküldés és videókonferencia lehetőségek segítik a csapatmunkát.
- **Naptár és ütemezés:** Beosztások kezelése, megtekintése, illetve a vakációs kérelmek benyújtására szolgál.
- **Adminisztrációs funkciók:** Felhasználók és jogosultságok kezelése, vállalati beállítások módosítása, licenszek nyilvántartása.

A rendszer biztonsági szempontból is megbízható: támogatja a biztonságos bejelentkezést, szerepköralapú jogosultságkezelést, valamint a vállalati adatok elkülönített és védett tárolását.

A **Managly** továbbá *könnyen bővíthető*, így a jövőben lehetőség van új modulok és funkciók – például mobilalkalmazás, külső API-integrációk vagy automatizált riportkészítő eszközök – beépítésére. A rendszer skálázhatósága és testreszabhatósága révén kiválóan alkalmas különböző méretű és profilú vállalatok számára.

2.2. Rendszerkövetelmények

Az alkalmazás futtatásához, fejlesztéséhez szükséges szoftvereket és csomagokat az alábbi felsorolás tartalmazza:

- .NET SDK 8.0 verzió
- Visual Studio 2022 17.5.2 vagy újabb
- NuGet csomagok:
 - BCrypt.Net-Next 4.0.3
 - MailKit 4.10.0
 - Microsoft.AspNetCore.Authentication.Cookies 2.3.0
 - Microsoft.AspNetCore.Authentication.JwtBearer 8.0.14
 - Microsoft.AspNetCore.Cryptography.KeyDerivation 8.0.1
 - Microsoft.AspNetCore.Identity 2.3.0
 - Microsoft.AspNetCore.Identity.EntityFrameworkCore 8.0.0
 - Microsoft.AspNetCore.SignalR 1.2.0

- Microsoft.AspNetCore.SignalR.Client 9.0.1
 - Microsoft.EntityFrameworkCore 8.0.2
 - Microsoft.EntityFrameworkCore.Design 8.0.2
 - Pomelo.EntityFrameworkCore.MySql 8.0.2
 - SendGrid 9.29.3
- Adatbázis:
 - MySQL Server 8.0 vagy újabb
 - MySQL Workbench (opcionális)
 - Fejlesztői eszközök:
 - Git verziókezelő rendszer
 - Postman vagy hasonló API tesztelő eszköz
 - Hardverkövetelmények:
 - Legalább 1 GB RAM
 - Legalább 2 GB szabad tárhely
 - Internet kapcsolat
 - Audió- és videó bemeneti eszközök (videókonferencia funkcióhoz)

Az alkalmazás fejlesztéséhez és teszteléséhez a saját, személyi számítógépemet használtam. Ennek specifikációi:

- Operációs rendszer: macOS 14.2.0 (Darwin)
- Processzor: Apple Silicon M1
- Memória: 8 GB RAM

Az alkalmazást Safari és Google Chrome böngészőt használva teszteltem.

2.3. Telepítés

2.3.1. Előzetes lépések

Telepítsük a rendszerkövetelményekben felsorolt programokat a megadott linkeken található telepítővel. Az alábbiakban részletezzük a telepítési lépéseket:

1. Telepítsük a .NET SDK 8.0 verziót¹.
2. Indítsuk újra a számítógépet.
3. Telepítsük a MySQL Server 8.0 adatbázis-kezelőt².
4. Telepítsük a Node.js 18.14.2 verziót vagy újabbat³.
5. Telepítsük az Entity Framework Core Tool-t a következő parancs kiadásával:
`dotnet tool install -global dotnet-ef`
6. Fejlesztéshez generálunk HTTPS tanúsítványt a következő parancssal: `dotnet dev-certs https -trust`

Az összes telepítés elvégzése után szükség lehet a számítógép újraindítására.

Hozzunk létre egy mappát a számítógép egy tetszőleges könyvtárában! Az alkalmazást ebbe a mappába fogjuk telepíteni. Például: C:/Managly.

2.3.2. Konfiguráció beállítása

A telepítéshez és helyes működéshez szükséges konfigurációkat az alábbiakban részletezzük:

appsettings.json fájl beállítása Végezzük el a Managly/appsettings.json fájl kitöltését:

- A ConnectionStrings alatti DefaultConnection kulcsú beállítás értéke a MySQL adatbázis kapcsolati sztringjét tartalmazza. Például:

```
"Server=localhost;Database=Managly;User=root;Password=your_password;"
```

¹<https://dotnet.microsoft.com/en-us/download/dotnet/8.0>

²<https://dev.mysql.com/downloads/mysql/>

³<https://nodejs.org/en>

Az adatbázis a telepítés során fog létrejönni. Fontos, hogy a megadott felhasználónév és jelszó rendelkezzen megfelelő jogosultságokkal az adatbázishoz.

- Az `EmailSettings` alatt az email küldéshez szükséges beállításokat adhatjuk meg:
 - `SenderEmail`: Küldő email címe
 - `SenderName`: Küldő neve
 - `SmtpHost`: SMTP szerver címe (pl. `smtp.gmail.com`)
 - `SmtpPort`: SMTP port száma (általában 587)
 - `SmtpUser`: SMTP felhasználónév
 - `SmtpPass`: SMTP jelszó vagy alkalmazásjelszó
- Amennyiben saját Gmail fiókot használunk a levelek küldéséhez, be kell állítani az alkalmazásjelszót a Google fiókban⁴ a kétlépcsős hitelesítés bekapcsolása után.
- Opcionálisan beállíthatjuk a HTTP és HTTPS portokat a `Urls` beállítással:

```
"Urls": "http://*:5050;https://*:5051"
```

SSL tanúsítvány beállítása A biztonságos HTTPS kapcsolathoz SSL tanúsítványt kell beállítani:

- **Fejlesztési környezetben:** Az ASP.NET Core fejlesztői tanúsítvanya automatikusan használatba kerül, ha a `dotnet dev-certs https -trust` parancsot lefuttattuk.
- **Éles környezetben:** A `localhost.crt` és `localhost.key` fájlok másolása a telepítési mappába, vagy hivatalos, megbízható tanúsítvány beszerzése (pl. Let's Encrypt).

⁴<https://myaccount.google.com/security>

Program.cs konfigurációk A Program.cs fájlban már több fontos beállítás található, amelyeket érdemes ismerni:

- **Dinamikus portok:** Az alkalmazás automatikusan kezeli a portok beállítását a következő prioritással:
 - Az appsettings.json fájlban megadottUrls beállítás
 - A HTTP_PORT és HTTPS_PORT környezeti változók
 - Az alapértelmezett 5050 (HTTP) és 5051 (HTTPS) portok
- **SSL tanúsítvány kezelése:** A Program.cs automatikusan betölti a localhost.crt és localhost.key fájlokat a HTTPS kapcsolat biztosításához.
- **MySQL verzió beállítása:** A MySQL verzió a new MySqlServerVersion(new Version(8, 0, 0)) sorban van beállítva, ami a 8.0.0 verzióra van konfigurálva a legjobb kompatibilitás érdekében.
- **Automatikus HTTPS átirányítás:** Az alkalmazás automatikusan átirányítja a HTTP kéréseket HTTPS-re, anélkül hogy ehhez külön konfigurációra lenne szükség.
- **Munkamenet és cookie beállítások:** A builder.Services.AddSession és builder.Services.ConfigureApplicationCookie blokkokban vannak beállítva. Fontos tudni, hogy a bejelentkezés csak HTTPS kapcsolaton keresztül fog működni a CookieSecurePolicy.Always beállítás miatt.

Alapértelmezett felhasználó és szerepkörök Az alkalmazás indításakor automatikusan létrejön egy alapértelmezett tulajdonosi (Owner) felhasználó, amennyiben még nincs ilyen felhasználó az adatbázisban. Az Owner felhasználó alapértelmezett adatai:

- Email: owner@managly.com
- Jelszó: Owner123!
- Név: System Owner

Emellett az alkalmazás a következő szerepköröket hozza létre az első indításkor:

- Owner (Tulajdonos)
- Admin (Adminisztrátor)
- Manager (Menedzser)
- Employee (Alkalmazott)

Fontos kiemelni, hogy az Owner felhasználó beállításai az inicializálásra vonatkoznak. A fentebb megadott adatokkal az alkalmazás első indításakor fog létrejönni a felhasználó. A létrejött Owner felhasználó jelszavát biztonsági okokból az első belépés után feltétlenül változtassa meg az alkalmazás felületén.

2.3.3. Az alkalmazás kihelyezése

Az előzetes lépések elvégzése után megkezdhetjük az alkalmazás kihelyezését a korábban erre a cérla létrehozott mappába. A kihelyezés során különös figyelmet kell fordítani az alkalmazás biztonsági beállításaira és az adatbáziskapcsolat megfelelő konfigurációjára.

1. A forráskód gyökérmapjából (ahol a Managly.sln található) adjuk ki a következő parancsot a szükséges függőségek letöltéséhez:

```
dotnet restore
```

2. Módosítsuk a Managly/appsettings.json fájlt a telepítési környezetnek megfelelően, különösen a kapcsolati sztringet és az e-mail beállításokat.
3. A gyökérmapában maradva adjuk ki a következő parancsot az adatbázis létrehozásához és a migrációk futtatásához:

```
dotnet ef database update
```

Ez létrehozza a MySQL adatbázist az appsettings.json fájlban megadott kapcsolati adatok alapján. Győződjünk meg róla, hogy a MySQL szerver fut és elérhető a telepítés során.

4. A Managly mappából adjuk ki a következő parancsot az alkalmazás kiadásához:

```
dotnet publish -c Release  
-o [a létrehozott mappa útvonala]  
-self-contained
```

A `-self-contained` kapcsolóval az alkalmazás tartalmazza a .NET runtimeot, így a futtatáshoz nem szükséges a célgépen .NET telepítése.

Például, ha a C:/Managly mappába telepítjük az alkalmazást, a kiadott parancs:

```
dotnet publish -c Release -o C:\Managly --self-contained
```

5. Másoljuk át a `localhost.crt` és `localhost.key` fájlokat a közzétett mappába, ha szükséges. Ezek a fájlok biztosítják a HTTPS kapcsolat működését.

Portok beállítása és HTTPS konfigurálása Az alkalmazás alapértelmezetten az 5050-es (HTTP) és 5051-es (HTTPS) portokon indul el. Ha ezek a portok már foglaltak, vagy más portokat szeretnénk használni, több lehetőségünk is van:

- **Környezeti változókkal:** A közzétett alkalmazást indíthatjuk egyedi portokkal:

```
HTTP_PORT=8080 HTTPS_PORT=8443 dotnet Managly.dll
```

- **Urls paraméterrel:** Közvetlenül parancssori argumentumként adjuk meg:

```
dotnet Managly.dll --urls="http://*:8080;https://*:8443"
```

- **appsettings.json módosításával:** A közzétett mappában módosítunk:

```
"Urls": "http://*:8080;https://*:8443"
```

Az alkalmazás az SSL tanúsítványt automatikusan kezeli. A fejlesztői környezetben a `dotnet dev-certs` parancssal generált, a közzétett verzióban pedig a `localhost.crt` és `localhost.key` fájlokat használja. Éles környezetben javasolt hivatalos, megbízható tanúsítvány használata, például a Let's Encrypt szolgáltatás-tól.

Az alkalmazás futtatása A fenti lépések elvégzése után a megadott mappában létrejött a Managly futtatható állomány. Az alkalmazás futtatásához a telepítési mappában lévő `Managly.exe` fájlt vagy Linux/macOS rendszeren a Managly binárist indítsuk el. Alternatív megoldásként a telepítési mappában kiadhatjuk a következő parancsot is:

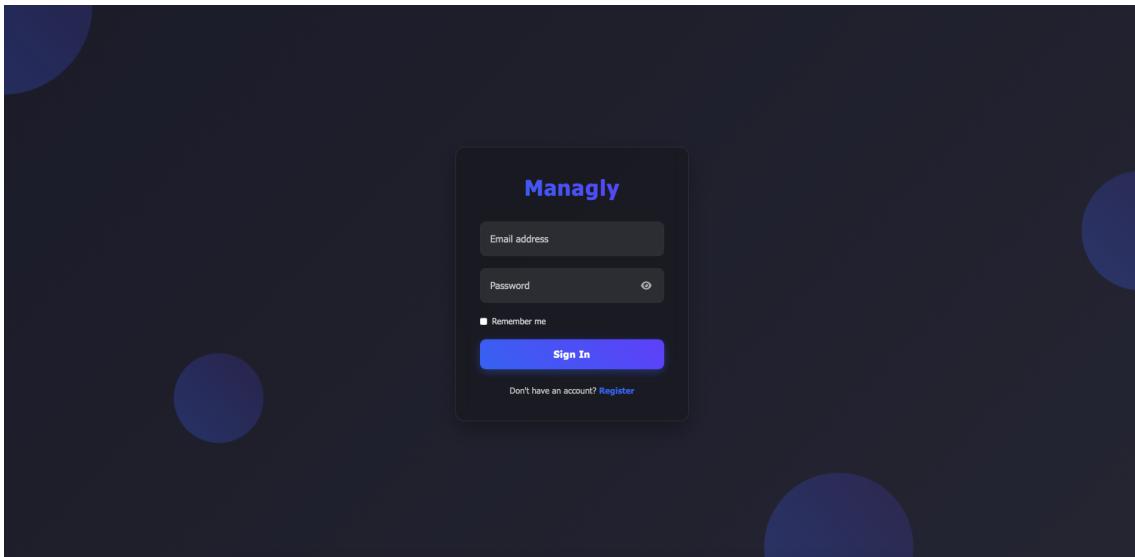
```
dotnet Managly.dll
```

A program elindulásakor a konzolablakban megjelenik (???. ábra) az alkalmazás elérési útvonala, ami alapértelmezetten `https://localhost:5051` vagy a konfigurált egyedi URL. Ezt az URL-t megnyitva egy böngészőben elérhetjük az alkalmazás felületét. Az alkalmazás leállítása a `CTRL+C` billentyűkombinációval történik.

2.4. Az alkalmazás általános használata

2.4.1. A felület

Az alkalmazás felhasználói felülete a modern webalkalmazások elveinek megfelelően lett kialakítva. Az oldal felkeresése során az első felület, amivel a felhasználó szembetalálkozik, a bejelentkező oldal (2.1. ábra).



2.1. ábra. A Managly bejelentkező felülete

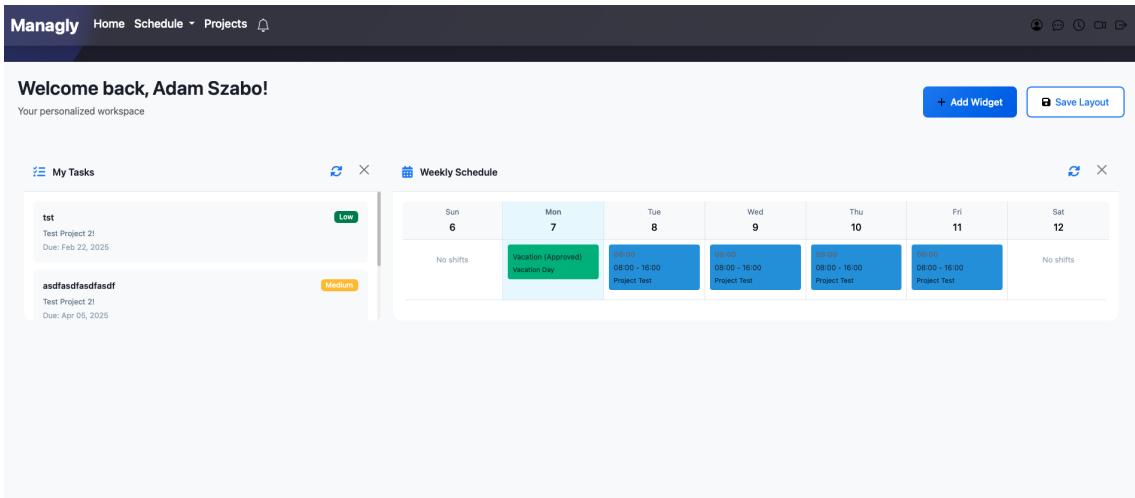
A bejelentkezést követően a rendszer automatikusan a felhasználó jogosultságának megfelelő felületre irányítja át. A Managly alkalmazásban hierarchikus jogosultsági rendszer van jelen, ahol minden felhasználó szerepkörétől függően különböző funkcionálitást és megjelenést kap.

2.4.1.1. Általános felhasználói felület

A legtöbb felhasználó (Admin, Manager, Employee) hasonló alapfelületet kap, amely a következő fő elemeket tartalmazza:

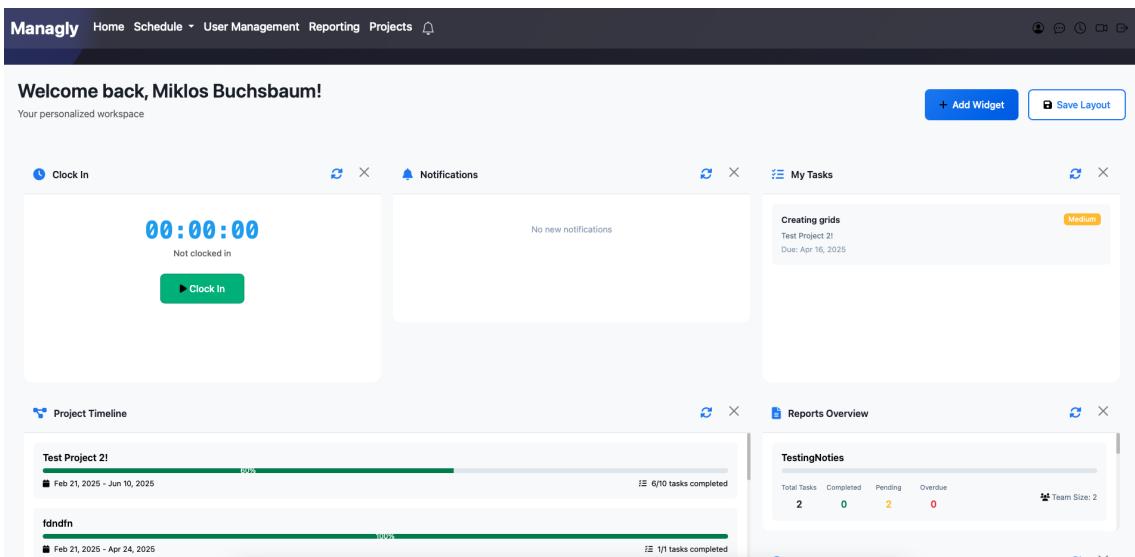
- **Navigációs sáv:** Az oldal tetején található, tartalmazza a különböző modulok (Főoldal, Naptár, Projektek, Értesítések, Profil, Chat, Jelenlét, Videókonferencia, Kijelentkezés) elérési lehetőségeit.
- **Dashboard:** Személyre szabható kezdőképernyő, ahova a felhasználók különböző widgeteket helyezhetnek el (2.2. ábra). Ezek a widgetek gyors áttekintést nyújtanak a projektek állapotáról, a közelgő határidőkről, és más fontos információkról.

2. Felhasználói dokumentáció



2.2. ábra. A Dashboard felülete a widgetekkel Epmployee felhasználók számára

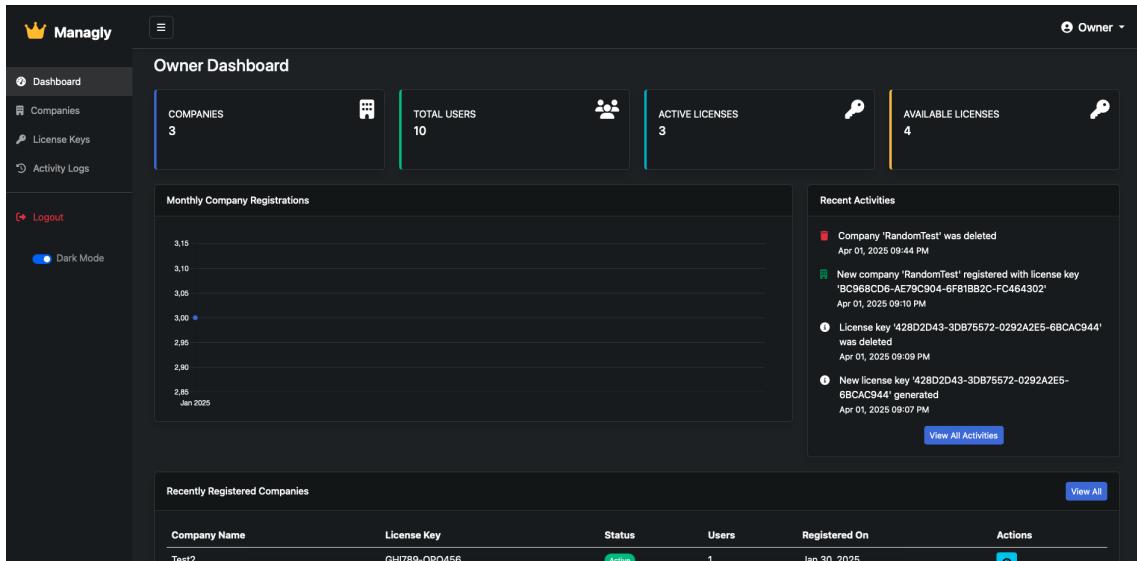
Az Admin jogosultsággal rendelkező felhasználók további modulat érhetnek el a felületen, mint például a felhasználók kezelése és a kimutatások. (2.3. ábra)



2.3. ábra. Az Admin felhasználói felület többletfunkciói

2.4.1.2. Owner felület

Az Owner (tulajdonos) jogosultsággal rendelkező felhasználó számára egy teljesen egyedi felület került kifejlesztésre (2.4. ábra). Ez a speciális felület a cég átfogó kezelésére összpontosít, és a következő fő elemeket tartalmazza: Dashboard, Companies, License Keys, Activity Logs.



2.4. ábra. Az Owner speciális felhasználói felülete

Az alkalmazásban található felületek minden eszközön is megfelelően jelennek meg. A felhasználói felületek tervezésénél fontos szempont volt az átláthatóság és az ergonomikus használat, ami megkönnyíti a felhasználók számára a különböző funkciók gyors elérését.

2.4.2. Visszajelzések

Az alkalmazás az egyes műveletek végrehajtásának sikerességéről visszajelzést ad értesítés formájában. Az értesítések a jobb felső sarokban jelennek meg toast üzenetek formájában, erre példát a 2.5 ábrán láthatunk.



2.5. ábra. Példa felugró értesítésre

Hiba esetén is azonos módon kapunk értesítést, például, ha egy olyan műveletet szeretnénk végrehajtani, amelyre az adott pillanatban nincsen jogunk. Természetesen - a megfelelő felhasználói élmény biztosítása érdekében - a felületen csak olyan műveletek végrehajtását kezdeményező vezérlők jelennek meg, amelyek ténylegesen végre is hajthatók az oldal betöltésének pillanatában. Azonban előfordulhat olyan eset, hogy - például - egy másik felhasználó egy olyan műveletet végez a rendszerben, amely következtében az általunk látott - már korábban betöltött - felület nem a naprakész jogosultságainknak megfelelően jelenik meg. Így amennyiben

hasonló esetet tapasztalunk, gondoljunk arra, hogy a megnyitott oldal frissítésére lehet szükség.

Felmerülhet a kérdés, hogy a frissítés ilyen esetekben miért nem automatikusan történik. A felhasználói élmény szempontjából zavaró lenne az, hogy hibaüzenetet látunk (jelezve, hogy az adott művelet nem hajtható végre), de közben az oldal állapota megváltozik (mintha mégis végrehajtottunk volna egy műveletet). Továbbá, ha a hibaüzenet megjelenítése után elrejtenénk az oldal korábbi állapotát, a felhasználónak nem lenne lehetősége felmérni, mi váltotta ki a hibaüzenetet.

2.4.2.1. Mezővalidáció

Kaphatunk hibaüzenetet validációs problémák miatt is, például ha az általunk megadott bemenet formátuma nem megfelelő. Ilyen esetekben a felugró értesítés mellett a konkrét hibáról is információt kapunk az érintett bemeneti mező(k)nél, ahogy ez a 2.6 ábrán is látható.

The image contains two side-by-side screenshots of a 'Create New Profile' interface. Both screenshots have a dark background and a blue circular icon with a person and a plus sign at the top.

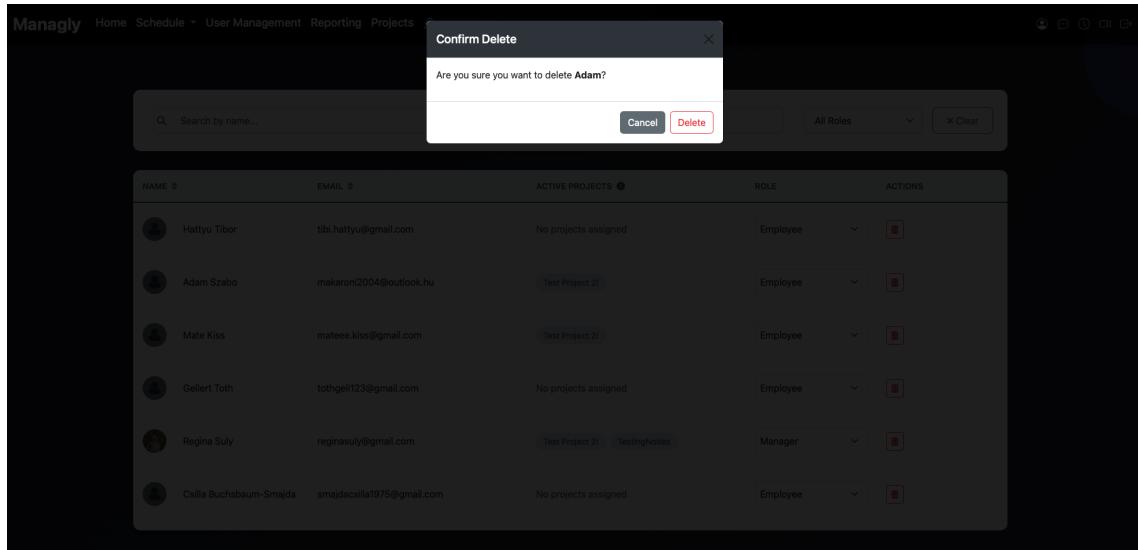
(a) Használatban lévő Email cím regisztrálása: This screenshot shows an email input field containing 'buchsbaum.miki2004@gmail.com'. Below it, a red error message says 'This email address is already registered.' There are other fields for 'First Name' (Buchsbaum), 'Last Name' (Miklos), and a dropdown for 'Employee' status. The 'Create Profile' button is at the bottom.

(b) Üresen hagyott mezők: This screenshot shows an empty 'First Name' input field. A red error message above it says 'First name is required'. Other fields like 'Last Name' (empty), 'Email' (example@example.com), and 'Employee' (dropdown open) are also shown. The 'Create Profile' button is at the bottom.

2.6. ábra. Validációs hibaüzenetek

2.4.2.2. Műveletek megerősítése

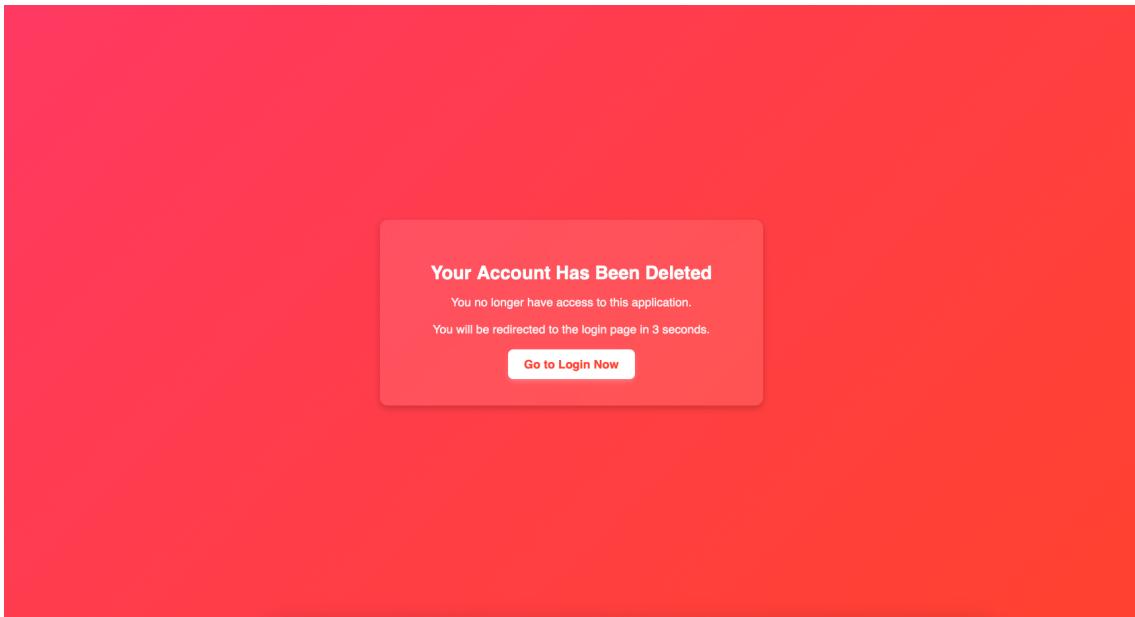
Egyes műveletek végrehajtása előtt a rendszer megerősítést kér. Ezek nagyobb hatású, nem (vagy nem azonnal) visszavonható műveletek esetén jellemzőek. Ilyenkor egy felugró párbeszédablakban (2.7 ábra) kell szándékunkat megerősíteneünk.



2.7. ábra. Példa megerősítésére szolgáló párbeszédablakra

2.4.2.3. Felhasználói fiók törlése

Amennyiben egy felhasználó fiókja törlésre kerül, és a felhasználó éppen be van jelentkezve, a rendszer automatikusan kijelentkezteti és átirányítja a "Fiók törölve" oldalra (2.8 ábra). Innét a felhasználó már csak a bejelentkező oldalra navigálhat, ahol természetesen a törölt fiókadatokkal már nem tud bejelentkezni.



2.8. ábra. Fiók törölve oldal

2.4.2.4. Értesítések a rendszerben

A rendszer emellett értesítéseket küld a felhasználóknak a következő esetekben:

- Új projekt létrehozása esetén a projekthez rendelt felhasználók értesítést kapnak
- Új feladat létrehozásakor a feladathoz rendelt felhasználók értesítést kapnak
- Feladat befejezéskor az Admin jogosultsággal rendelkező felhasználók értesítést kapnak
- Chat üzenetek érkezéskor, ha a felhasználó nincs a chat felületen
- Videókonferencia hívás érkezéskor értesítést kap a hívott fél, illetve nem fogadott hívás esetén is

2.4.2.5. Email értesítések

A rendszer továbbá Email értesítést küld minden Admin jogosultsággal rendelkező felhasználó számára az adott szervezeten belül, ha egy felhasználó vakációt kérvényezett, hogy azt minél előbb engedélyezni vagy adott esetben elutasítani tudják.

Most, hogy megismerkedtünk az alkalmazás felületének általános jellemzőivel, a következő fejezetekben rátérünk konkrét funkcionálisok ismertetésére, felhasználói történetek által meghatározott egy lehetséges sorrendben.

2.5. Általános felhasználói felület

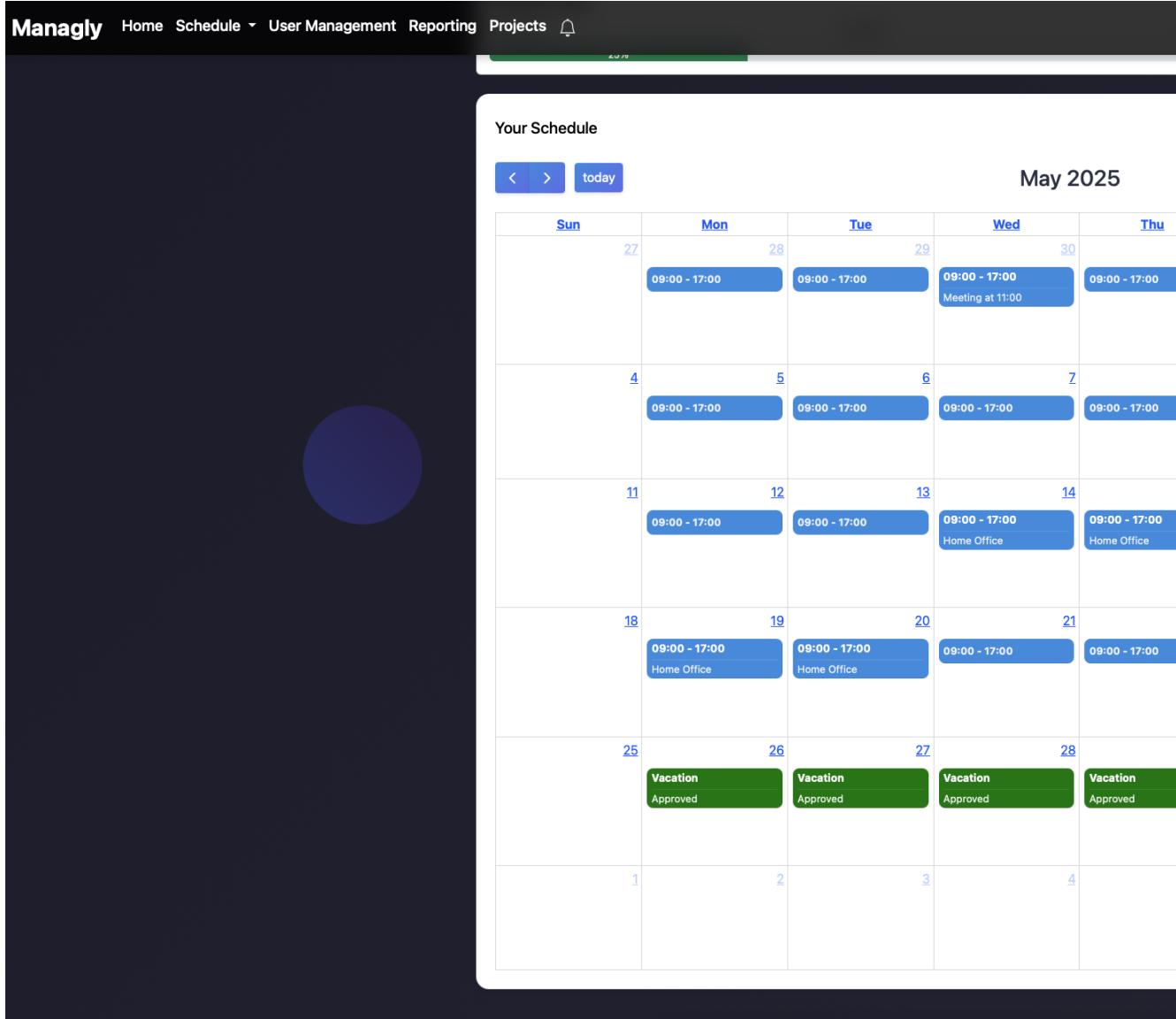
Először az általános felhasználói felületet kerül bemutatásra, mivel az Owner jogosultsággal rendelkező felhasználó számára egy teljesen egyedi felület került fejlesztésre.

2.5.1. Naptár funkciók

A Managly alkalmazás naptár funkciói komplex munkaidő- és szabadságkezelési rendszert biztosítanak a felhasználók számára. A rendszer különböző jogosultsági szintekkel rendelkező felhasználók számára nyújt megfelelő felületeket, amelyek segítségével hatékonyan koordinálhatják a vállalati munkarendet, szabadságokat és távollétéket.

2.5.1.1. Beosztás megtekintése

A beosztás megtekintése funkcióval az összes felhasználó egyszerűen nyomon követhetik saját munkarendjüket és szabadságaikat egy könnyen átlátható naptár felületen.



2.9. ábra. Felhasználói beosztás megtekintése

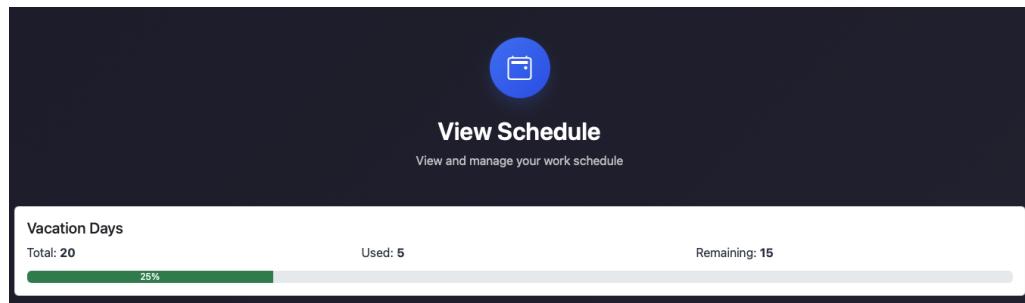
Fő funkciók

- Személyes naptár:** A felhasználók FullCalendar alapú naptár nézetén tekintetik meg beosztásukat, amely színkódjal különbözteti meg a munkanapot és a szabadságot.
- Esemény részletek:** A naptárbejegyzésekre kattintva részletes információkat kaphatnak a műszakról vagy szabadságról, beleértve a pontos időpontokat és a hozzáfűzött megjegyzésekét.



2.10. ábra. Esemény részletei

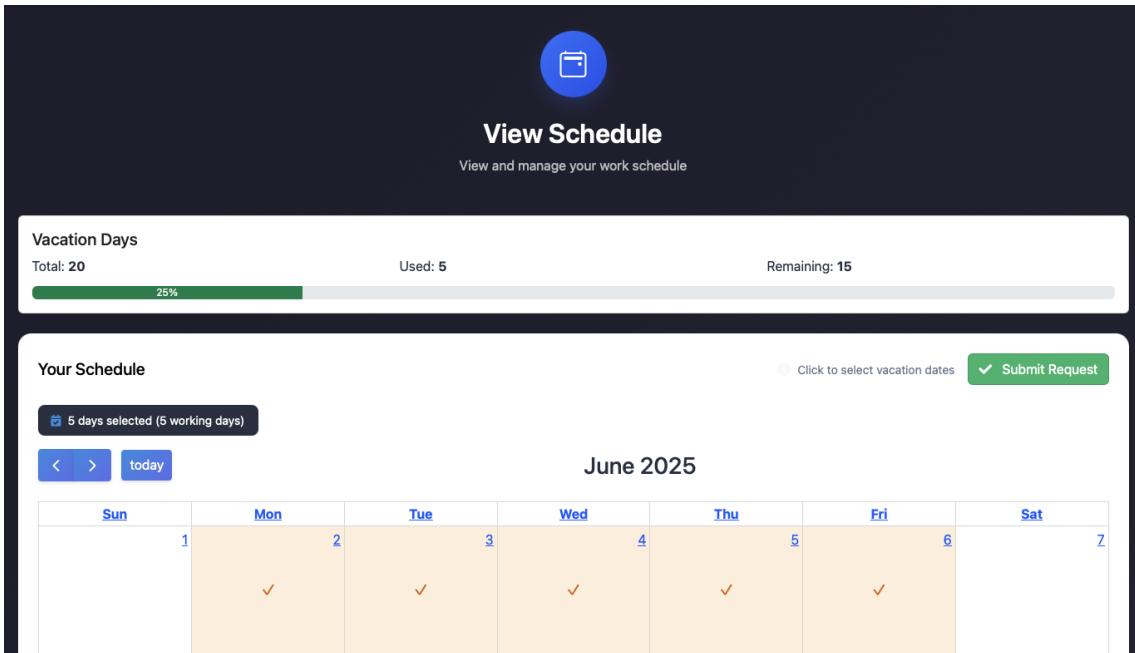
- **Szabadságkérés:** A felhasználók közvetlenül a naptárban választhatják ki a kívánt szabadság időpontját, és egyszerűen kérelmezhetik annak jóváhagyását.
- **Szabadságkeret követése:** Az alkalmazás valós időben mutatja a felhasználó rendelkezésre álló, felhasznált és hátralévő szabadságnapjainak számát, százalékos vizualizációval kiegészítve.



2.11. ábra. Szabadságkeret követése

Szabadságigénylés folyamata A felhasználók az alábbi lépésekkel igényelhetnek szabadságot:

1. A "Request Vacation" gombra kattintva aktiválódik a szabadságkérési mód.
2. A felhasználó kiválasztja a kívánt napokat a naptárban, amely azonnal vissza-jelzést ad a kiválasztott napok számáról és a munkanapokon belüli arányáról.



2.12. ábra. Szabadságigénylés folyamata

3. Az alkalmazás automatikusan ellenőrzi, hogy a választott napokon nincs-e már munkaműszak vagy jóváhagyott szabadság, és figyelmeztet, ha átfedés van.
4. A rendszer valós időben ellenőrzi, hogy a felhasználónak van-e elég szabadság-napja a kérelem teljesítéséhez, és vizuális figyelmeztetést jelenít meg, ha nincs elegendő nap.
5. A "Submit Request" gombra kattintva a kérelem jóváhagyásra kerül a rendszerben, és a felhasználó azonnali visszajelzést kap a sikeres beküldésről. Az Admin jogosultsággal rendelkező felhasználók Email értesítést kapnak a leadott igénylésről.

A rendszer intelligensen kezeli a hétvégéket, és csak a munkanapokat számolja bele a felhasznált szabadságnapok számába. Az igénylés után a szabadságkérelmek "Pending" (Függőben) státusszal jelennek meg a naptárban narancssárga színnel, jelzve, hogy adminisztrátori jóváhagyásra várnak. Jóváhagyást követően zöld színnel jelennek meg a kiválasztott szabadságok.



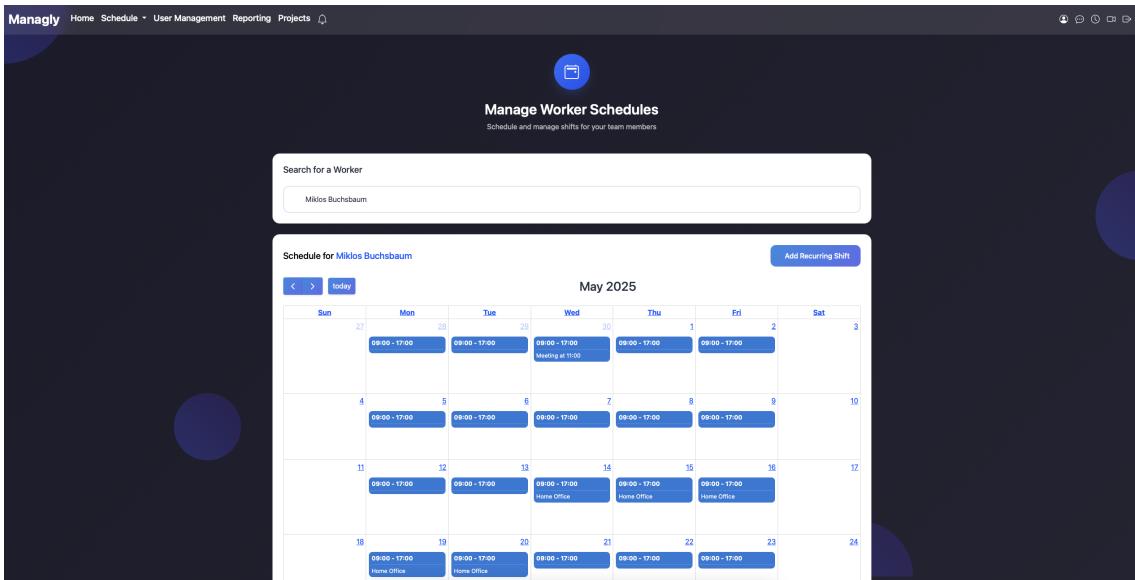
(a) Függő szabadságkérelem

(b) Jóváhagyott szabadságkérelem

2.13. ábra. Szabadságkérelemek állapota

2.5.1.2. Beosztás kezelése

Az Admin és Manager jogosultsággal rendelkező felhasználók számára elérhető a beosztás kezelése funkció, amely lehetővé teszi a dolgozók munkarendjének és szabadságkérélmeinek teljes körű adminisztrációját.



2.14. ábra. Beosztás kezelése adminisztrátori nézet

Fő funkciók

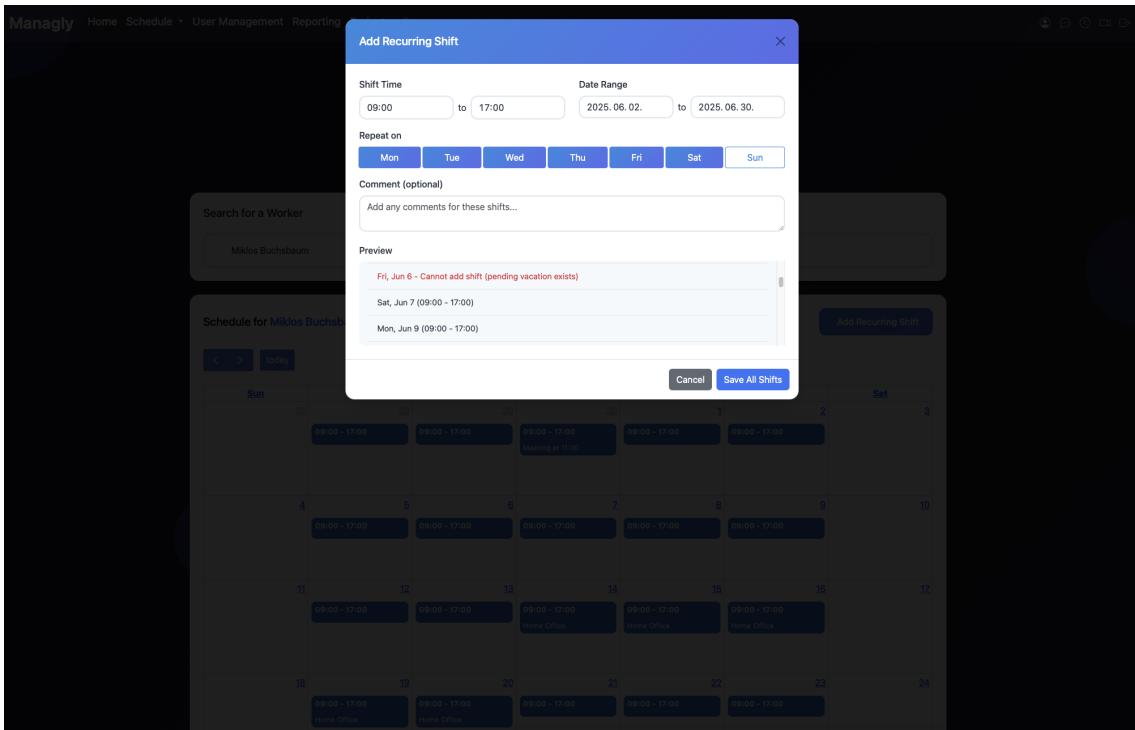
- Dolgozók keresése:** Az adminisztrátorok gyors keresőfunkció segítségével választhatják ki a kezeltet kívánt dolgozó munkarendjét.
- Műszakok létrehozása:** Lehetőség van egyéni műszakok létrehozására, megadva a kezdő- és befejezási időpontot, valamint esetleges megjegyzéseket.
- Ismétlődő műszakok:** Az adminisztrátorok ismétlődő műszakokat állíthatnak be különböző napokon, időszakokon belül, ami jelentősen megkönnyíti a rendszeres beosztások készítését.
- Műszakok szerkesztése:** A létrehozott műszakok utólag módosíthatók, ami lehetővé teszi a rugalmas időbeosztás kezelését.
- Szabadságkérelmek kezelése:** Az adminisztrátorok jóváhagyhatják, elutasíthatják vagy törölhetik a dolgozók szabadságkérélmeit.

- **Elérhetőség ellenőrzése:** A rendszer automatikusan ellenőrzi, hogy van-e ütközés a tervezett műszakok és a meglévő szabadságok vagy más műszakok között.

Az összes funkció felugró ablakok segítségével hajtható végre.

Ismétlődő műszakok kezelése A rendszer különösen hasznos funkciója az ismétlődő műszakok létrehozása (2.15. ábra):

1. Az "Add Recurring Shift" gombra kattintva megnyílik az ismétlődő műszakok kezelőfelülete.
2. Az adminisztrátor beállíthatja a műszak időpontját (kezdés és befejezés).
3. Kiválaszthatja a dátumtartományt, amelyen belül az ismétlődést alkalmazni szeretné.
4. Megjelölheti a hét napjait, amelyeken a műszakot létrehozza.
5. Valós idejű előnézet mutatja a létrehozandó műszakok listáját.
6. A rendszer automatikusan ellenőrzi az elérhetőséget, és figyelmezteti az adminisztrátort az esetleges ütközésekre. Ütközés esetén is lehetőség van a beosztást létrehozni. Ebben az esetben az ütköző napok kihagyásra kerülnek.



2.15. ábra. Ismétlődő műszakok kezelése

Ez a funkció rendkívül hatékony módszer a rendszeres munkarendek gyors létrehozására, különösen nagyobb létszámú csapatok esetén.

Szabadságkérelmek kezelése Az adminisztrátorok és menedzserek az alábbi lehetőségekkel kezelhetik a beosztott dolgozók szabadságkérelmeit:

- **Jóváhagyás:** A kérelem elfogadása, amely után a szabadság zöld színnel jelent meg a naptárban, és automatikusan levonásra kerül a dolgozó szabadságkeretéből.
- **Elutasítás:** A kérelem elutasítása, amely nem terheli a dolgozó szabadságkeretét.
- **Törlés:** Korábban jóváhagyott szabadság törlése, amely visszaállítja a dolgozó szabadságkeretét.
- **Áthelyezés:** Lehetőség van a már jóváhagyott szabadság másik időpontra történő áthelyezésére is.

Minden műveletnél a rendszer automatikusan frissíti a dolgozó szabadságkeretét, és csak a munkanapot számolja bele a felhasznált napok közé.

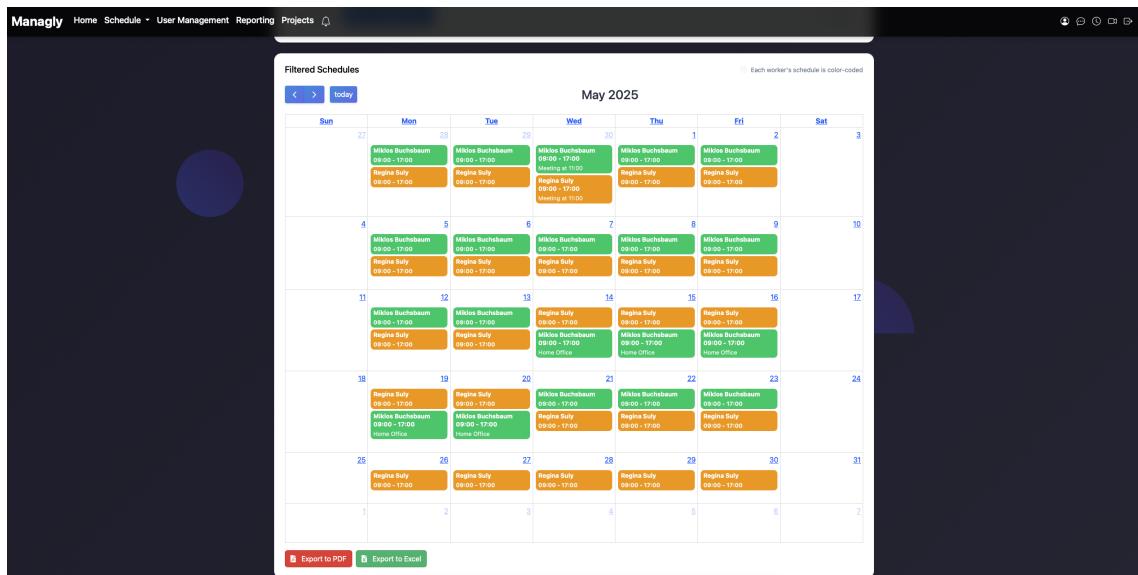
Korlátozások A beosztás kezelése során a rendszer számos korlátozást alkalmaz a hibák elkerülése érdekében:

- Nem hozható létre műszak olyan napra, amelyen a dolgozónak már jóváhagyott szabadsága van.
- Nem módosítható vagy törölhető múltbeli műszak vagy szabadság.
- Nem hozható létre átfedő műszak ugyanazon dolgozónak ugyanarra a napra.
- Automatikus ellenőrzés történik az ismétlődő műszakok létrehozásakor, hogy ne ütközzenek meglévő műszakokkal vagy szabadságokkal.

Ezek a korlátozások biztosítják a beosztási rendszer integritását és megakadályozzák az ellentmondásos bejegyzések létrehozását. Kényelmi funkcióként egy áthúzásban alapuló (drag and drop) mechanizmus került implementálásra, amely lehetőséget biztosít a beosztások teljes körű, gyors és rugalmas átrendezésére. Ebben az esetben is a felugró értesítések gondoskodnak arról, hogy a felhasználó visszajelzést kapjon műveleteiről.

2.5.1.3. Összevont beosztás

Az összevont beosztás nézet lehetővé teszi az Admin és Manager jogosultsággal rendelkező felhasználók számára, hogy átfogó képet kapjanak a vállalat összes dolgozójának munkarendjéről és szabadságairól egy közös naptárban.



2.16. ábra. Összevont beosztás nézet

Fő funkciók

- **Dolgozók szerinti szűrés:** Az adminisztrátorok kiválaszthatják azokat a dolgozókat, akiknek a beosztását meg szeretnék jeleníteni az összevont naptárban. Itt egy vagy akár több felhasználóra is lehetőség van szűrni név alapján.
- **Projektek szerinti szűrés:** Lehetőség van a projektek alapján történő szűrére, amely megjeleníti az adott projektben résztvevő összes dolgozó beosztását.
- **Színkódolás:** minden dolgozó beosztása egyedi színkóddal jelenik meg, amely segíti a könnyű azonosítást a közös naptárban.
- **Aktív szűrők követése:** A felület jelzi az aktuálisan aktív szűrőket, és lehetőséget biztosít azok egyenkénti vagy együttes törlésére.
- **Részletes eseménymegjelenítés:** Az egyes naptárbejegyzések a dolgozó nevét, a műszak időpontját és esetleges megjegyzéseket is tartalmaznak.
- **Exportálási lehetőségek:** Az összevont beosztás exportálható PDF vagy Excel formátumban, amely megkönnyíti a jelentések készítését és a beosztás megosztását.

Szűrési lehetőségek Az összevont beosztás nézet két fő szűrési lehetőséget kínál, amelyek nem kombinálhatók egymással (???. ábra):

1. **Dolgozók szerinti szűrés:** Az adminisztrátor kereshet és kiválaszthat egy vagy több dolgozót, akiknek a beosztását együttesen szeretné látni. minden dolgozó beosztása egyedi színnel jelenik meg.
2. **Projektek szerinti szűrés:** Az adminisztrátor kiválaszthat egy vagy több projektet, és a rendszer automatikusan megjeleníti az adott projektekhez rendelt összes dolgozó beosztását.

A dolgozók szerinti szűrés különösen hasznos lehet konkrét csapatok vagy osztályok munkarendjének követésére, míg a projektek szerinti szűrés átfogó képet nyújt egy adott projekt résztvevőinek rendelkezésre állásáról.

Exportálási funkciók Az összevont beosztás kétféle formátumban exportálható:

- **PDF export:** A rendszer egy több oldalas PDF dokumentumot generál, amelyben dolgozónként külön oldalon jeleníti meg a beosztást táblázatos formában. Az exportban szerepel egy összefoglaló oldal is, amely tartalmazza az alkalmazott szűrőket és az exportálás időpontját.
- **Excel export:** A rendszer egy Excel táblázatot generál, amelyben minden dolgozó beosztása külön munkalapon jelenik meg, valamint egy összefoglaló munkalap is készül a szűrési feltételekkel.

Az exportált állományok névkonvenciója tartalmazza a hónapot, az évet és az alkalmazott szűrőket (pl. "schedule-april-2025-example-project.xlsx"), amely segíti a dokumentumok könnyű azonosítását és rendszerezését.

2.5.2. Felhasználók kezelése

2.5.3. Felhasználók kezelése

A Managly rendszer egyik alapvető funkciója a felhasználók kezelése, amely a vállalati hierarchia és munkafolyamatok hatékony működtetéséhez elengedhetetlen. Ez a modul a rendszeradminisztrátorok számára biztosít eszközöket a cég alkalmazottainak kezeléséhez, beleértve a profilk létrehozását, profilk törlését, a szerepkörök kiosztását, valamint a szabadságnapok nyilvántartását. A felhasználókezelés rendszerét szerepköralapú jogosultságkezelés szabályozza, biztosítva, hogy minden felhasználó csak a számára engedélyezett műveleteket végezhesse el.

2.5.3.1. Szerepkörök és jogosultságok

A rendszerben definiált szerepkörök hierarchikus struktúrát alkotnak, amely szabályozza a felhasználók hozzáférését a különböző funkciókhöz:

- **Admin:** Teljes körű hozzáféréssel rendelkezik a rendszerhez. Képes felhasználókat létrehozni, törölni, szerepköröket módosítani, és az összes alkalmazott szabadságnajait kezelní.
- **Manager:** Részleges adminisztrációs jogosultságokkal rendelkezik. Megtekintheti az összes felhasználót, módosíthatja a szabadságnapot,

de nem változtathatja meg a szerepköröket és nem hozhat létre vagy törlhet felhasználókat.

2.5.3.2. Felhasználói felületkezelés és interakciók

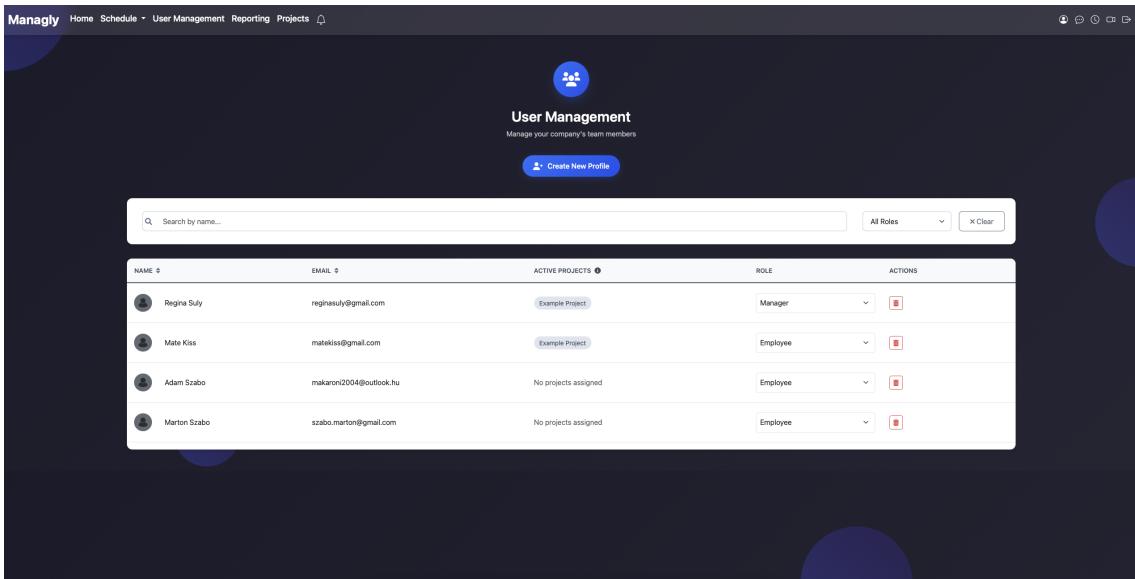
Felhasználók áttekintése A felhasználókezelési felület központi eleme a felhasználók listáját megjelenítő táblázat. A táblázat tartalmazza az összes felhasználót, aki az adott szervezethez hozzá van rendelve. minden sorban egy felhasználó jelenik meg és a megjelenített adatok tartalmazzák a felhasználó profilképét, teljes nevét, e-mail címét, a hozzájuk rendelt aktív projekteket és jogosultági körüket. Szervezeten kívüli felhasználók természetesen nem elérhetőek. A táblázat alábbi funkciókat biztosítja (2.17. ábra):

1. **Keresés és szűrés:** Az alkalmazottak kereshetők név szerint, valamint szűrhetők szerepkör alapján.
2. **Rendezés:** A lista rendezhető nevek és e-mail címek szerint, növekvő vagy csökkenő sorrendben.
3. **Felhasználó eltávolítása:** Csak az Admin jogosultsággal rendelkező felhasználók távolíthatnak el felhasználókat a szervezetből.
4. **Jogosultsági szerepkör módosítása:** Csak az Admin jogosultsággal rendelkező felhasználók tudják a felhasználók jogosultsági szerepkörét módosítani.
5. **Felhasználó profiladatainkat megtekintése:** Az Admin, illetve Manager jogosultsággal rendelkező felhasználók megjeleníthetik a kiválasztott felhasználó összes profil adatát.

Felhasználói profilok részletei minden felhasználóhoz részletes profil tartozik a táblázatban a nevükre kattintva és ez tartalmazza:

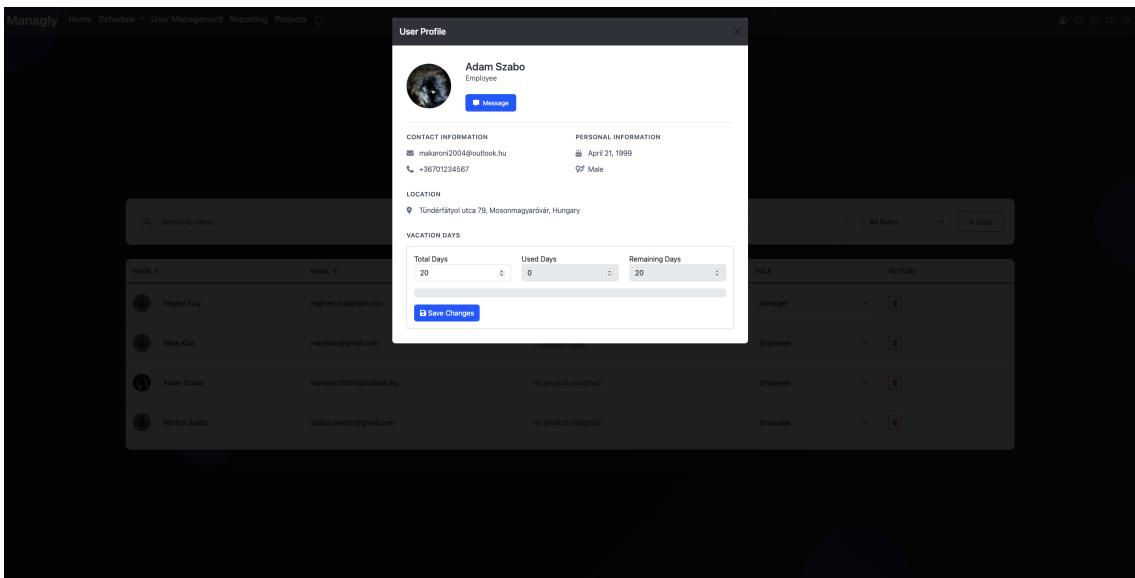
- Alapvető személyes adatokat (név, e-mail, telefonszám)
- Demográfiai információkat (ország, város, cím, születési dátum, nem)
- Projekttagságokat és szerepköröket
- Szabadságnapok nyilvántartását

2. Felhasználói dokumentáció



2.17. ábra. Felhasználókezelési felület áttekintése

A rendszer modális ablakokban jeleníti meg a részletes profil információkat (2.18. ábra).



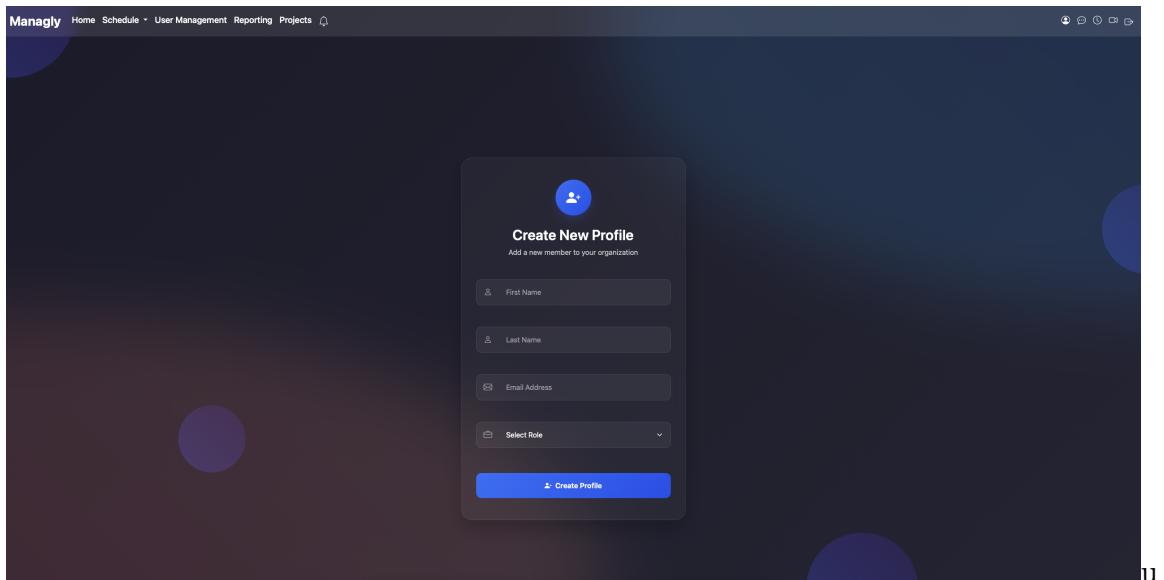
2.18. ábra. Felhasználó profil részletei

2.5.3.3. Felhasználói profilk kezelése

Új felhasználó létrehozása Az Admin jogosultsággal rendelkező felhasználók képesek új alkalmazottat felvenni a rendszerbe. A folyamat a következő lépésekkel áll (2.19. ábra):

1. Az adminisztrátor megadja az új felhasználó alapvető adatait

- Keresztnév és vezetéknév
 - E-mail cím (amely egyben a felhasználónévként is szolgál)
 - Szerepkör (Employee vagy Manager)
2. A rendszer generál egy véletlenszerű, biztonságos, egyszer használatos jelszót az újonnan létrehozott felhasználó számára
 3. Az új felhasználót létrehozza az adatbázisban, alapértelmezett beállításokkal:
 - 20 szabadságnap évente
 - Alapértelmezett profilkép
 - Előre meghatározott alapvető adatok
 4. A rendszer e-mailt küld az új felhasználónak a hozzáférési adatokkal
 5. A felhasználó első bejelentkezéskor köteles megváltoztatni a jelszavát



2.19. ábra. Új felhasználó létrehozása

Új felhasználó létrehozása természetesen csak szervezeten belül lehetséges. Validáció során a rendszer ellenőrzi, hogy minden szükséges megadásra kerüljön, illetve a megadott cím egyedi legyen. Ez azt jelenti, hogy ugyanazzal az e-mail címmel két felhasználó nem rendelkezhet egyidejűleg a szervezeten belül. Természetesen, ha egy felhasználó több szervezethez is tartozik, akkor neki ugyanazzal az e-mail címmel a másik szervezetben is létrehozhaztól egy profil.

Szerepkörök módosítása A felhasználók szerepköreinek módosítása egy kulcsfontosságú adminisztrátori funkció, amely lehetővé teszi a jogosultságok dinamikus kezelését:

1. Az adminisztrátor kiválaszt egy új szerepkört a felhasználónak
2. A rendszer megerősítést kér a változtatásról
3. Az adatbázis sikeres frissítése után a felhasználói felület is azonnal frissül

Ez a funkciót csak az Admin jogosultsággal rendelkező felhasználók érhetik el. A megváltozott jogosultságot a felhasználó az első oldalfrissítést követően élvezheti. A folyamatos felhasználói élmény érdekében került ez így megvalósításra.

Felhasználó törlése Az Admin jogosultsággal rendelkező felhasználók törölhetnek felhasználókat a rendszerből. Ez a művelet több lépésből áll:

1. Az Admin kiválasztja a törölni kívánt felhasználót
2. A rendszer megerősítést kér a törlés előtt egy felugró ablakban, mivel ez egy vissza nem vonható folyamat
3. Majd a felhasználó teljesen törölve lesz

Ez egy gondosan megtervezett folyamat, mivel törlés következtével a felhasználóhoz tartozó összes adat törlésre kerül. Ez azt jelenti, hogy minden nyilvántartásból elvész, mint például a projektek. A törlött felhasználó a következő lapfrissítésig tudja az oldalt használni. Utána egy üzenet jelenik meg számára, ami közli vele, hogy profilja törlésre került a szervezeten belül. Lehetősége van visszanavigálni a bejelentkező felületre, de ha nem teszi, akkor a rendszer 10 másodpercen belül ezt megteszi helyette.

2.5.3.4. Szabadságnapok kezelése

Szabadságnapok nyilvántartása A rendszer nyilvántartja minden felhasználó szabadságnapjait:

- **Összes szabadságnap:** Az évente rendelkezésre álló napok száma (alapértelmezetten 20 nap)

- **Felhasznált szabadságnapok:** Az adott évben már igénybe vett napok száma
- **Fennmaradó szabadságnapok:** Az még rendelkezésre álló napok száma
- **Szabadság év:** Az aktuális évet nyilvántartó mező, amely az év váltásakor automatikusan frissül

Szabadságnapok módosítása Az Admin és Manager jogosultsággal rendelkező felhasználók módosíthatják a felhasználók szabadságnapjainak számát (2.18. ábra):

1. A felhasználó profilján módosítható az összes szabadságnap
2. A rendszer validálja, hogy a megadott érték legalább 20 legyen (törvényi minimum)
3. Az új értékek alapján frissül a fennmaradó napok száma és a szabadság felhasználási grafikon

Éves szabadságnapok automatikus frissítése A rendszer automatikusan kezeli az év váltását és nullázza a felhasznált szabadságnapok számát. Ez biztosítja, hogy az Admin és Manager jogosultsággal rendelkező felhasználók adminisztratív munkája könnyítésre kerüljön. Ez az automatizált folyamat egy kényelmi funkció a rendszerben. please do the same thing with this section as well

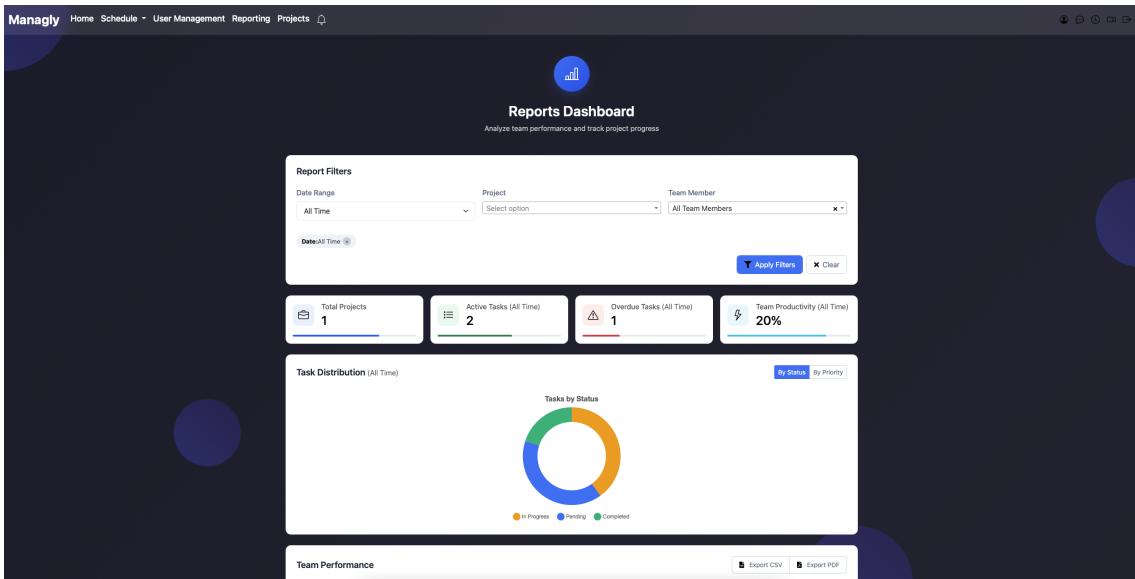
2.5.4. Kimutatások

A Managly rendszer jelentések kezelése funkciója egy átfogó elemzési eszközt kínál a projekt teljesítmény és csapat produktivitás nyomon követéséhez. Az irányítópult modern, intuitív felülete lehetővé teszi a különböző szempontok szerinti adatszűrést és elemzést, így a felhasználók könnyedén megérthetik és követhetik a projekt állapotát és a csapat teljesítményét.

A jelentések modul a következő kulcsfontosságú funkciókat kínálja:

- Projekt teljesítmény nyomon követése
- Csapat produktivitás elemzése
- Feladat eloszlás vizualizációja

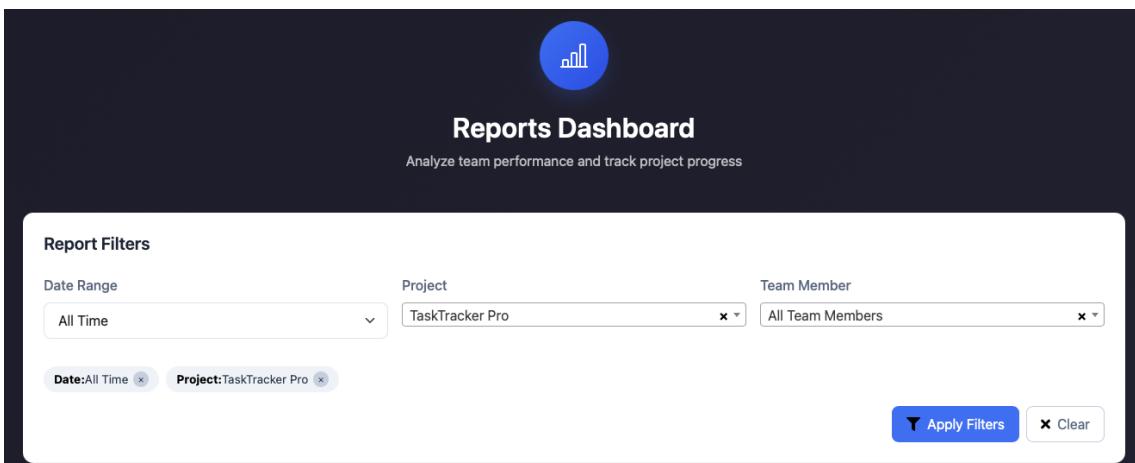
2. Felhasználói dokumentáció



2.20. ábra. Kimutatások oldal áttekintése

- Testreszabható szűrési lehetőségek
- Adatok exportálása CSV és PDF formátumban

A jelentések oldal tetején található szűrők segítségével a felhasználók pontosan meghatározhatják, hogy mely adatokat szeretnék megtekinteni. A szűrők kombinálhatók, így a felhasználók a saját igényeiknek megfelelően szűkíthetik a megjelenített adatokat.



2.21. ábra. Szűrési lehetőségek a kimutatások oldalon

Dátum Szűrés A dátum szűrés lehetővé teszi a különböző időszakok adatainak megtekintését. Az alapértelmezett beállítás az utolsó 30 nap adatait mutatja, de a felhasználók választhatnak más időszakot is:

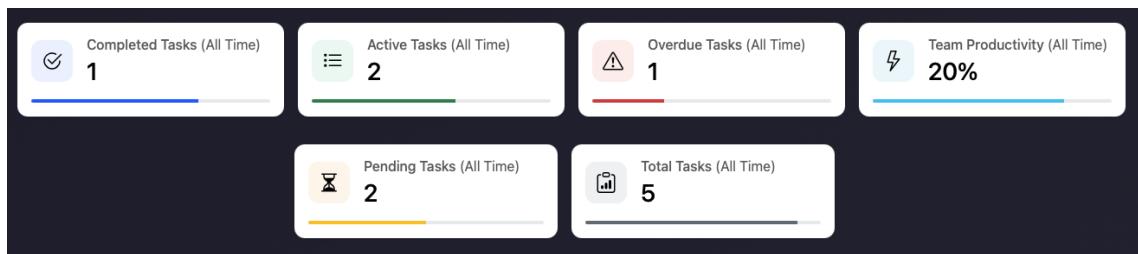
- Összes időszak megtekintése
- Utolsó 7 nap adatai
- Utolsó 30 nap adatai (alapértelmezett)
- Utolsó 90 nap adatai
- Egyéni dátumtartomány megadása

Projekt Szűrés A projekt szűrés segítségével a felhasználók kiválaszthatják, hogy mely projekt adatait szeretnék megtekinteni. Egy adott projekt kiválasztásakor a rendszer automatikusan módosítja a megjelenített statisztikákat, hogy csak a kiválasztott projekthez tartozó adatokat mutassa.

Csapat Tag Szűrés A csapat tag szűrés lehetővé teszi az egyéni teljesítmény nyomon követését. A felhasználók kiválaszthatnak egy adott csapattagot, és részletesen elemezhetik annak teljesítményét, beleérvtve a befejezett feladatok számát, a ledolgozott órákat és a produktivitás százalékos arányát.

A rendszer vizuálisan megjeleníti az aktívan használt szűrőket, így a felhasználók minden tisztában vannak azzal, hogy milyen szűrési feltételek alapján jelennek meg az adatok. A szűrők gyorsan eltávolíthatók, és a szűrők kombinációja áttekinthető, így a felhasználók könnyen módosíthatják a megjelenített adatokat.

A szűrők alatt található statisztikai kártyák gyors áttekintést nyújtanak a legfontosabb mutatókról. Ezek a kártyák vizuálisan jelenítik meg a kulcsfontosságú metrikákat, így a felhasználók egy pillantással megérthetik a projekt és a csapat aktuális állapotát.



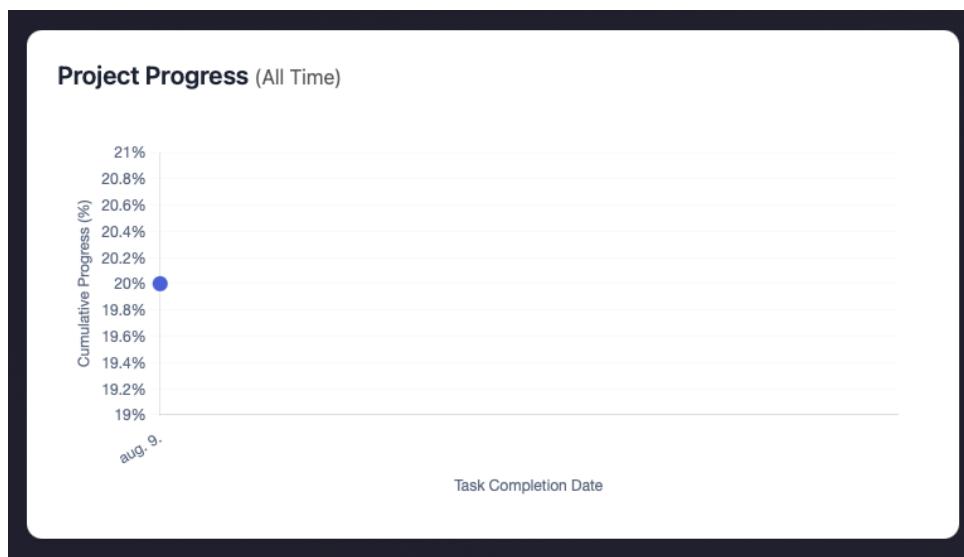
2.22. ábra. Statisztikai kártyák a kimutatások oldalon

Projektek és Feladatok A statisztikai kártyák között megtalálható az összes projekt száma (vagy a kiválasztott projekt befejezett feladatainak száma), az aktív feladatok száma, a késedelmes feladatok száma és a csapat produktivitás százalékos aránya. Ezek a mutatók segítenek a projekt haladásának és a csapat teljesítményének gyors értékelésében.

Projekt-specifikus Statisztikák Amikor egy adott projekt van kiválasztva, a rendszer további projekt-specifikus statisztikákat jelenít meg. Ezek közé tartozik a függően lévő feladatok száma és az összes feladat száma, amelyek segítenek a projekt aktuális állapotának pontosabb megértésében.

Integrálásra kerültek különböző vizuális elemek is, amik fejlesztik a felhasználói élményt és áttekinthetőséget.

Projekt Haladás Diagram A projekt haladás diagram egy vonaldiagram formájában mutatja be a projekt haladását az idő függvényében. A diagram interaktív, így a felhasználók az egérrel a diagram felett való mozgatással részletes információkat kaphatnak az egyes pontokról. A diagramon látható a befejezett feladatok százalékos aránya, amely segít a projekt haladásának vizuális követésében.

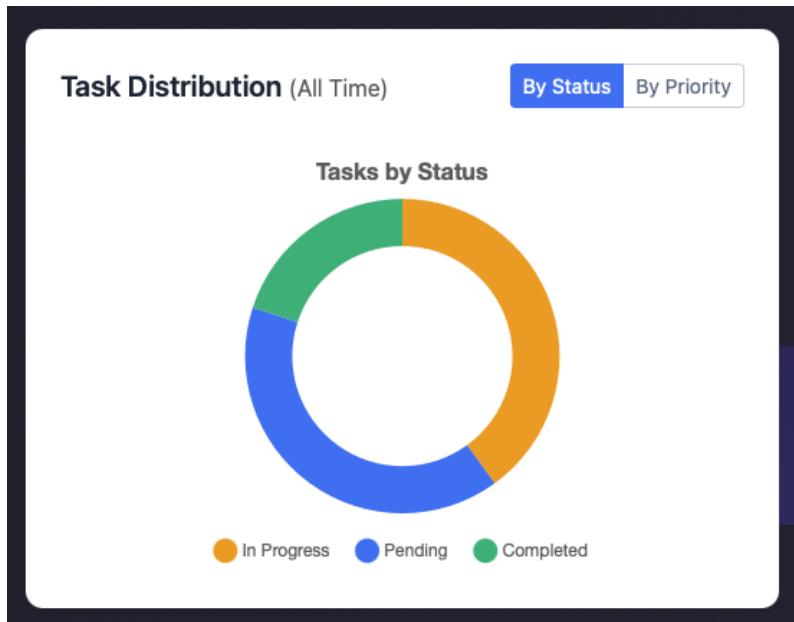


2.23. ábra. Projekt haladás diagram

Feladat Eloszlás Diagram A feladat eloszlás diagram egy kördiagram formájában mutatja be a feladatok eloszlását. A diagram két különböző nézet között válthat:

- Állapot szerinti eloszlás (Befejezett, Folyamatban, Függőben, Nem kezdett, Késedelmes)
- Prioritás szerinti eloszlás (Alacsony, Közepes, Magas, Sürgős)

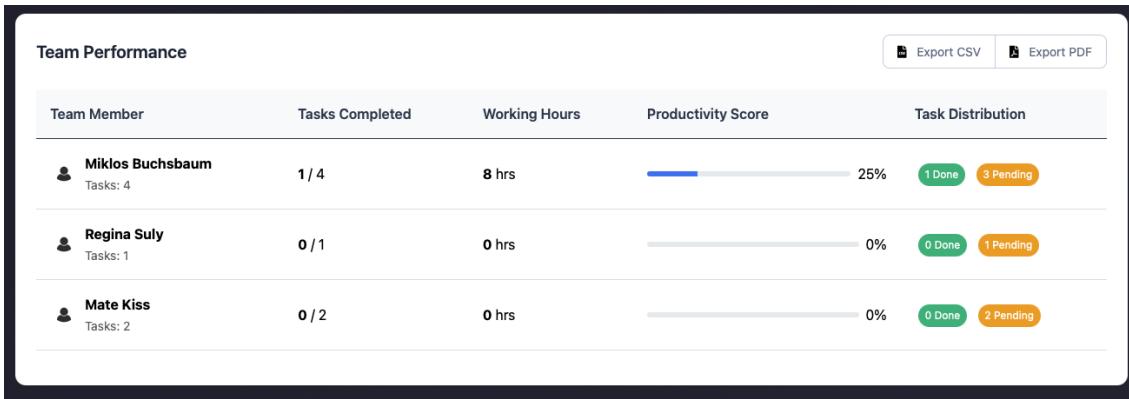
A diagram színkódolással segíti az egyes kategóriák megkülönböztetését, így a felhasználók könnyen megérthetik a feladatok eloszlását.



2.24. ábra. Feladat eloszlás diagram

További megtalálható az oldalon egy csapat teljesítmény táblázat. A csapat teljesítmény táblázat részletes áttekintést nyújt minden csapattag teljesítményéről. A táblázat oszlopai a következő információkat tartalmazzák:

- Csapattag neve és profilképe
- Befejezett feladatok száma az összes feladathoz képest
- Összes ledolgozott óra
- Produktivitás százalékos aránya vizuális sávdiagrammal
- Feladat eloszlás gyors áttekintése



2.25. ábra. Csapat teljesítmény táblázat

Exportálási Lehetőségek A csapat teljesítmény adatai exportálhatók CSV és PDF formátumban is. A PDF vagy CSV exportálás során a rendszer egy professzionálisan formázott dokumentumot generál, amely tartalmazza a dátumtartomány információt, a generálás időpontját és a teljes csapat teljesítmény adatait. A PDF és CSV dokumentum később nyomtatható vagy megosztható más felhasználókkal.

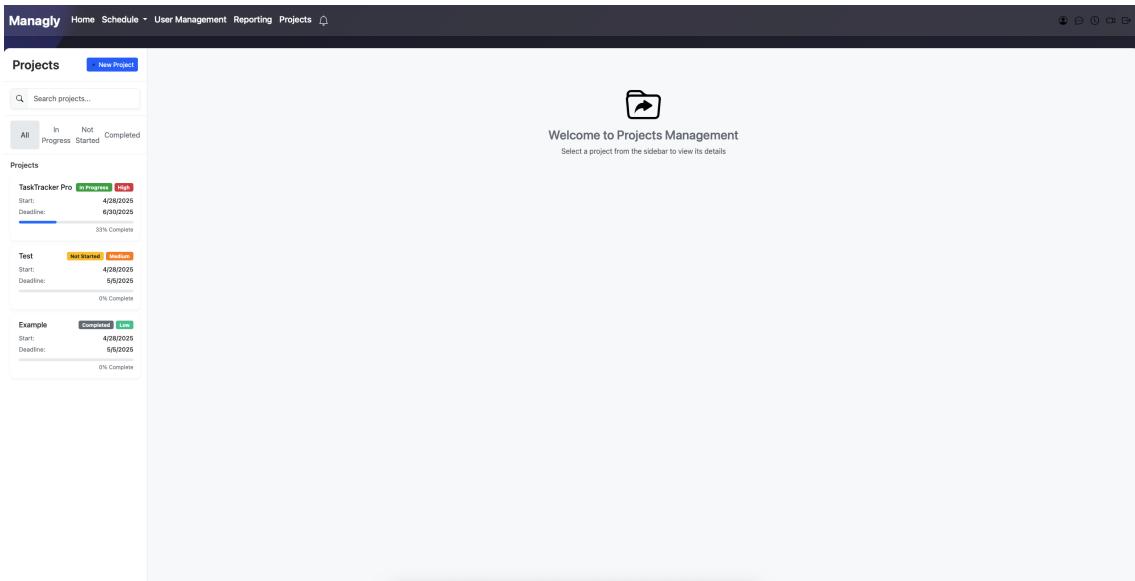
2.5.5. Projektek

A Managly rendszer Projektek modulja egy átfogó megoldást kínál a projektmenedzsment feladatokra, lehetővé téve a felhasználók számára a projektek létrehozását, követését, és a csapatmunka hatékony koordinálását. Az oldal kialakítása a modern projektmenedzsment igényeihez igazodik, felhasználóbarát felülettel, amely segíti a gyors és hatékony munkavégzést.

A Projektek oldal két fő részre tagolódik: bal oldalon található a projektek lista, míg a jobb oldalon az aktuálisan kiválasztott projekt részletei jelennek meg. Ez a kétpaneles elrendezés lehetővé teszi a felhasználók számára, hogy gyorsan navigáljanak a különböző projektek között, miközben részletes információkat látnak a kiválasztott projektről.

Projektek Oldalsáv Az oldalsáv a projektek gyors elérését és szűrését teszi lehetővé. Itt találhatók a szűrési lehetőségek, a keresési funkció és a projektek lista. Az oldalsáv tetején elhelyezkedő "Új projekt létrehozása" gomb csak adminisztrátori jogosultsággal rendelkező felhasználók számára érhető el.

A projektek listája alatt az egyes projekteket kártyák jelenítik meg, amelyek tartalmazzák a projekt nevét, állapotát és egyéb alapvető információkat. A projektek



2.26. ábra. Projektek oldal áttekintése a kétpaneles elrendezéssel

színkódolással is el vannak látva az állapotuk alapján, ami segíti a vizuális azonosítást.

Projektek Szűrése és Keresése Az oldalsáv felső részén található szűrési lehetőségek segítségével a felhasználók különböző szempontok szerint szűkíthetik le a megjelenített projekteket. A szűrési lehetőségek között szerepelnek:

- minden projekt megjelenítése
- folyamatban lévő projektek
- nem kezdett projektek
- befejezett (archivált) projektek

A szűrők alatt található keresőmező lehetővé teszi a projektek név szerinti keresését. A felhasználó által beírt kulcsszavak alapján a rendszer azonnal frissíti a projektlistát, csak azokat a projekteket jelenítve meg, amelyek nevében szerepel a keresett kifejezés.

Az adminisztrátori jogosultsággal rendelkező felhasználók új projekteket hozhatnak létre a rendszerben. Az "Új projekt létrehozása" gombra kattintva egy úrlap jelenik meg, ahol megadhatók a projekt részletei.

Projekt Létrehozási Űrlap Az új projekt létrehozásakor a következő adatokat kell megadni:

- Projekt neve (kötelező): A projekt egyedi azonosítója a felhasználók számára.
- Leírás: A projekt céljának és hatókörének részletes leírása.
- Kezdési dátum (kötelező): A projekt hivatalos kezdési dátuma.
- Határidő (kötelező): A projekt tervezett befejezési dátuma.
- Prioritás: A projekt fontosságának jelzése (Alacsony, Közepes, Magas).
- Állapot: A projekt kezdeti állapota (Nem kezdett, Folyamatban).
- Csapattagok: A projekthez rendelt felhasználók kiválasztása, akik értesítést kapnak a hozzárendelésről.

2.27. ábra. Új projekt létrehozása űrlap

Amikor a felhasználó kiválaszt egy projektet a listából, a jobb oldali panelen megjelennek a projekt részletei. Ez az oldal átfogó képet nyújt a projektről, beleérte a projekt állapotát, a csapattagokat, a feladatokat és a projekt tevékenységi naplóját.

Projekt Fejléc A projekt fejléce tartalmazza a legfontosabb információkat a projektről:

- Projekt neve és leírása

2. Felhasználói dokumentáció

The screenshot displays the Managly web application's user interface for managing projects. The main area shows a project named 'TaskTracker Pro' with the following details:

- Project Details:** Started: Apr 28, 2025; Deadline: Jun 30, 2025; Status: In Progress; Priority: High.
- Progress:** 33% completed, 1 of 3 tasks completed.
- Team Members:** Buchsbaum Miklos (Project Lead), Mate Kiss, Szabo Marton, Tamas Schnitzer.
- Tasks:**
 - Design UI Wireframes:** Create high-fidelity wireframes for all core pages (dashboard, task list, task detail, settings) to establish the visual structure and user flow. Assigned to: Buchsbaum Miklos, Mate Kiss.
 - Set Up Project Backend:** Initialize the backend using Node.js and Express. Set up RESTful APIs and connect to a MongoDB database to store users, tasks, and project data. Assigned to: Buchsbaum Miklos, Tamas Schnitzer.
- Recent Activity:** Shows three entries from 'Buchsbaum Miklos' indicating task completion and status changes.

2.28. ábra. Kiválasztott projekt áttekintése

- Kezdési dátum és határidő
- Projekt állapota és prioritása színkódokkal ellátva
- Projekt haladási százalék, amely a befejezett feladatok arányát mutatja

This screenshot shows the 'TaskTracker Pro' project page with the following visible elements:

- Project Details:** Started: Apr 28, 2025; Deadline: Jun 30, 2025; Status: In Progress; Priority: High.
- Progress:** 33% completed, 1 of 3 tasks completed.
- Recent Activity:** Shows three entries from 'Buchsbaum Miklos' indicating task completion and status changes.
- In the top right corner of the main content area, there are 'Edit' and 'Delete' buttons.

2.29. ábra. Projekt fejléc a legfontosabb projekt információkkal

A fejléc jobb oldalán található a projekt műveleti menüje, amely lehetőséget biztosít a projekt szerkesztésére és törlésére.

Tevékenységi Napló A tevékenységi napló a projekt összes tevékenységét rögzíti, beleérte a feladatok létrehozását, módosítását, a csapattagok változását és egyéb fontos eseményeket. A napló kronológiai sorrendben jeleníti meg az eseményeket, a legújabbakkal kezdve. minden bejegyzés tartalmazza a tevékenység típusát, a végrehajtó felhasználót, az időpontot és a tevékenység részleteit.

2. Felhasználói dokumentáció

The screenshot displays the Managly software interface. On the left, there's a sidebar with 'Projects' and 'Team Members' sections. The 'Projects' section shows a summary for 'TaskTracker Pro' with a deadline of June 30, 2025, and a progress bar at 33%. Below it are sections for 'Test' and 'Example'. The 'Team Members' section lists 'Buchsbaum Miklos' as the 'Project Lead' and 'Mate Kiss' as a 'Member'. On the right, a large window titled 'Project Activity History' is open, showing a timeline of events for the 'TaskTracker Pro' project. Events include task creation, updates, and assignments. A specific task, 'Set Up Project Backend', is highlighted with a yellow border. At the bottom right of the activity history window, there are buttons for 'Add Member' and 'Manage'.

2.30. ábra. Projekt tevékenységi napló

A tevékenységi napló különösen hasznos a projekt vezetői számára, akik nyomon követhetik a projekt előrehaladását és a csapattagok aktivitását. A felhasználók a "Összes tevékenység megtekintése" gombra kattintva láthatják a projekt teljes tevékenységi történetét.

Csapattagok Szekció A csapattagok szekció mutatja a projekthez rendelt felhasználókat és szerepköreiket. minden csapattag mellett megjelenik a profilképe, neve és projektben betöltött szerepe (Projekt Vezető vagy Tag). A projekt vezetője és az adminisztrátorok számára itt jelenik meg a "Csapattagok kezelése" gomb is, amely lehetővé teszi új tagok hozzáadását, meglévő tagok szerepkörének módosítását vagy tagok eltávolítását.

This screenshot shows the 'Team Members' section of the Managly interface. It lists three members: 'Buchsbaum Miklos' (Project Lead), 'Mate Kiss' (Member), and 'Szabo Marton' (Member). Each member is represented by a small profile icon and their name. To the right of the list, there are buttons for 'Add Member' and 'Manage'.

2.31. ábra. Csapattagok szekció a projektben résztvevő tagokkal

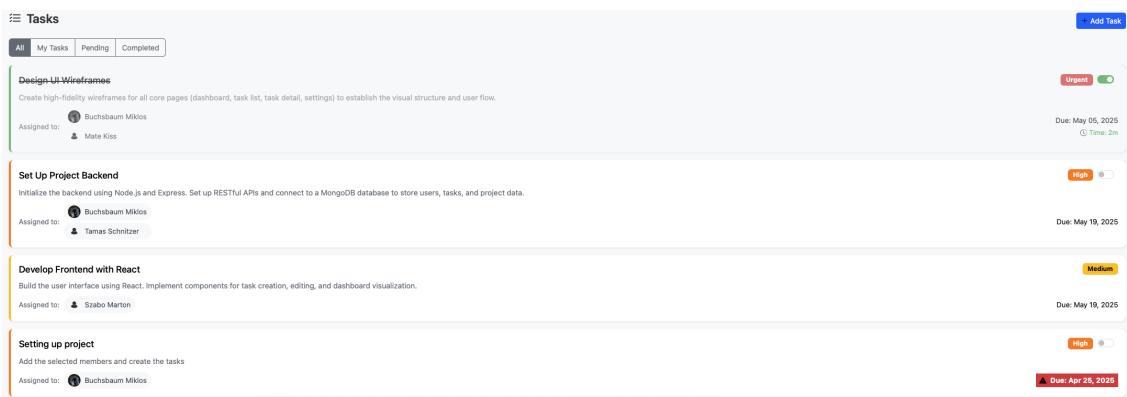
Csapattagok Hozzáadása A "Csapattagok kezelése" gombra kattintva, majd a megjelenő modálablakban az "Új tag hozzáadása" opciót választva a projekt vezetője vagy adminisztrátor új tagokat adhat a projekthez. A felhasználók kereshetők név szerint, és a kiválasztott felhasználók hozzáadhatók a projekthez. Az új csapattagok automatikusan értesítést kapnak a hozzárendelésről.

Szerepkörök Módosítása A csapattagok listájában minden tag mellett található egy szerepkör legördülő menü, amely lehetővé teszi a tagok szerepkörének módosítását. A lehetséges szerepkörök közé tartozik a "Projekt Vezető" és a "Tag". A szerepkör módosítását csak a projekt vezetője vagy adminisztrátor végezheti el.

Csapattagok Eltávolítása A csapattagok listájában minden tag mellett található egy "Eltávolítás" gomb, amellyel a tag eltávolítható a projektből. Az eltávolítás előtt a rendszer megerősítést kér, hogy elkerülje a véletlen eltávolításokat. A csapattag eltávolítását csak a projekt vezetője vagy adminisztrátor végezheti el, és a Projekt Vezető nem távolítható el, amíg van más csapattag a projektben.

Feladatok Szekció A feladatok szekció a projekthez tartozó összes feladatot megjeleníti, lehetőséget biztosítva azok szűrésére és rendezésére. A feladatok szűrhetők:

- Összes feladat: Az összes projekthez tartozó feladat megjelenítése.
- Saját feladatok: Csak a bejelentkezett felhasználóhoz rendelt feladatok megjelenítése.
- Függőben lévő feladatok: A nem befejezett feladatok megjelenítése.
- Befejezett feladatok: A befejezett feladatok megjelenítése.



2.32. ábra. Feladatok szekció a különböző szűrési lehetőségekkel

Minden feladat egy kártya formájában jelenik meg, amely tartalmazza a feladat címét, leírását, határidejét, prioritását, állapotát és a hozzárendelt felhasználókat. A feladatok állapota és prioritása színkódolással is el van látva a gyors vizuális azonosítás érdekében. A feladatkártyákra kattintva megjelenik a feladat részletes

nézete, ahol a felhasználók módosíthatják a feladat tulajdonságait és törölni is tudja az adott feladatot.

Mivel minden feladat határidőhöz van szabva, ezért, ha egy feladat határidőre nem készül el, azt a program külön vizuálisan jelzi neki. Ha valaki a feladat határideje után fejezi azt be, az is jelezve lesz. Adott feladat befejezése után a felhasználó láthatja azt is, hogy mennyi időt töltött az adott feladattal.



2.33. ábra. Lejárt határidejű feladat vizuális jelzése

Új Feladat Létrehozása Új feladat létrehozásához a felhasználónak a "Feladat létrehozása" gombra kell kattintania a feladatok szekciójában. Megjelenik egy űrlap, ahol megadhatók a feladat részletei:

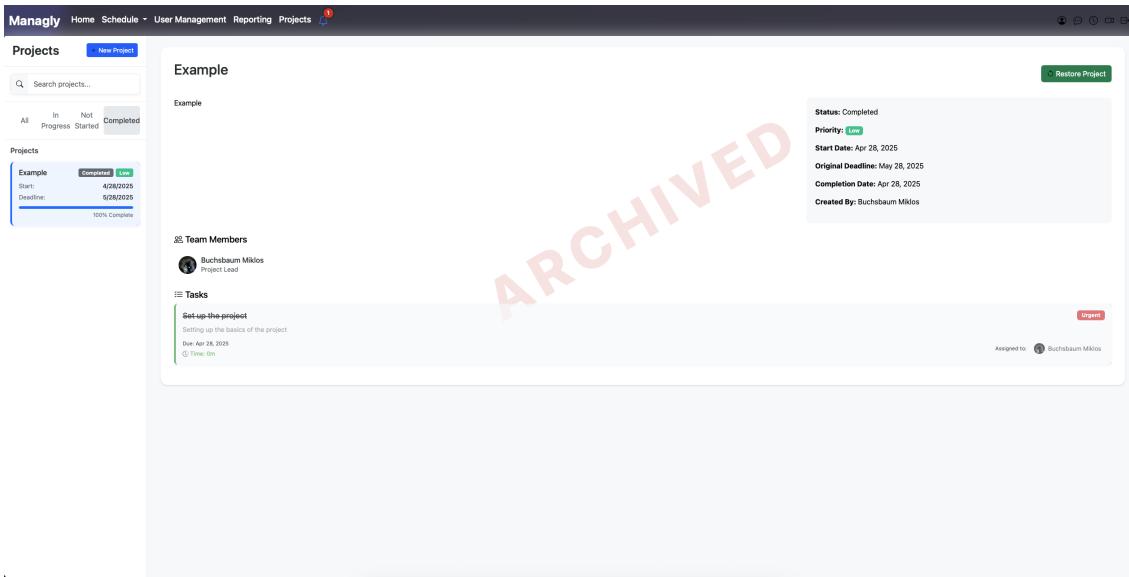
- Feladat címe (kötelező): A feladat rövid, leíró neve.
- Leírás: A feladat részletes leírása és követelményei.
- Határidő (kötelező): A feladat befejezési határideje.
- Prioritás: A feladat fontossága (Alacsony, Közepes, Magas).
- Hozzárendelés (kötelező): A feladatért felelős csapattag(ok) kiválasztása.

A feladat létrehozása után az automatikusan megjelenik a feladatok listájában, és értesítést kapnak a hozzárendelt felhasználók.

A befejezett projektek archiválhatók, ami lehetővé teszi a projektek tisztán tartását, miközben megőrzi az összes projekttörténetet.

Projekt Archiválása Amikor egy projekt eléri a "Befejezett" állapotot, automatikusan archiválódik. Az archivált projektek továbbra is megtekinthetők, de nem módosíthatók. Az archivált projektek a szűrők között a "Befejezett" kategóriában találhatók.

Archivált Projekt Részletei Az archivált projektek részletei hasonlóak az aktív projektekrehez, de bizonyos funkciók, mint például a feladatok létrehozása vagy módosítása, nem érhetők el. Az archivált projektek részletei között szerepel a befejezés dátuma, a projektben résztvevő csapattagok listája és az összes feladat állapota.



2.34. ábra. Archivált projekt részletei

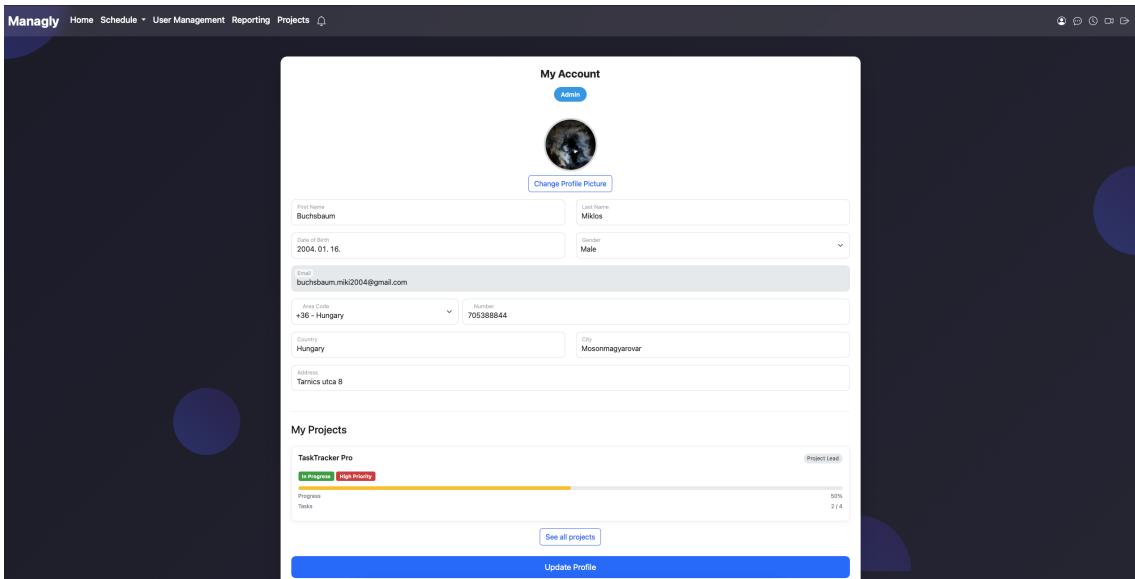
Projekt Visszaállítása Az adminisztrátorok visszaállíthatják az archivált projekteket aktív állapotba, ha szükséges. A "Projekt visszaállítása" gombra kattintva megjelenik egy űrlap, ahol megadható az új határidő, állapot és prioritás. A visszaállítás után a projekt ismét aktívvá válik, és minden funkcionális elérhetővé válik.

2.5.6. Felhasználói profil

A Managly rendszer felhasználói profil funkciója lehetővé teszi a felhasználóknak személyes adataik, elérhetőségeik és projektjeik kezelését egy intuitív, felhasználóbarát felületen keresztül. A profiloldal nemcsak a személyes információk szerkesztését teszi lehetővé, hanem áttekintést is nyújt a felhasználó aktuális projektjeiről, ezzel segítve a hatékonyabb munkavégzést és a jobb szervezettséget.

A kártya tetején látható a "My Account" fejléc, amely alatt a felhasználó rangja jelenik meg, amely lehet Admin, Manager vagy Employee. Ezt a rangot egy színkódolt jelvény jelzi, amely segít vizuálisan megkülönböztetni a különböző szerepkörökkel rendelkező felhasználókat.

A felhasználók személyre szabhatják profiljukat egy egyedi profilkép hozzáadásával. A "Change Profile Picture" gombra kattintva lehetőség nyílik új profilkép feltöltésére. A rendszer támogatja a JPEG, PNG és GIF formátumú képfájlokat, amelyek mérete nem haladhatja meg az 5 MB-ot. A feltöltés során a rendszer automatikusan ellenőrzi a fájl méretét és formátumát, és hiba esetén értesítést küld a



2.35. ábra. A profil oldal áttekintése

felhasználónak. A sikeres feltöltés után az új profilkép azonnal megjelenik az oldalon, így a felhasználók rögtön láthatják a változtatást.

A profil űrlap több szakaszra oszlik, amelyek lehetővé teszik a különböző személyes adatok kezelését, mint például a név, telefonszám es lakkcím.

Az űrlap alján található az "Update Profile" gomb, amelyre kattintva a felhasználó elküldheti a módosított adatokat. A rendszer ellenőrzi az összes mezőt, és csak akkor frissíti a profilt, ha az összes kötelező mező ki van töltve és az adatok megfelelnek a követelményeknek. Sikeres frissítés esetén a rendszer egy zöld üzenetben értesíti a felhasználót a sikeres műveletről. Ha hiba lép fel a frissítés során, a rendszer piros üzenetben jelzi a hibát.

A személyes adatok alatt található a "My Projects" szekció, amely áttekintést nyújt a felhasználó aktuális projektjeiről. Ez a funkció különösen hasznos azoknak, akik több projekten is dolgoznak, és gyorsan szeretnének információt kapni a projektek állapotáról anélkül, hogy a projektek oldalra kellene navigálniuk.

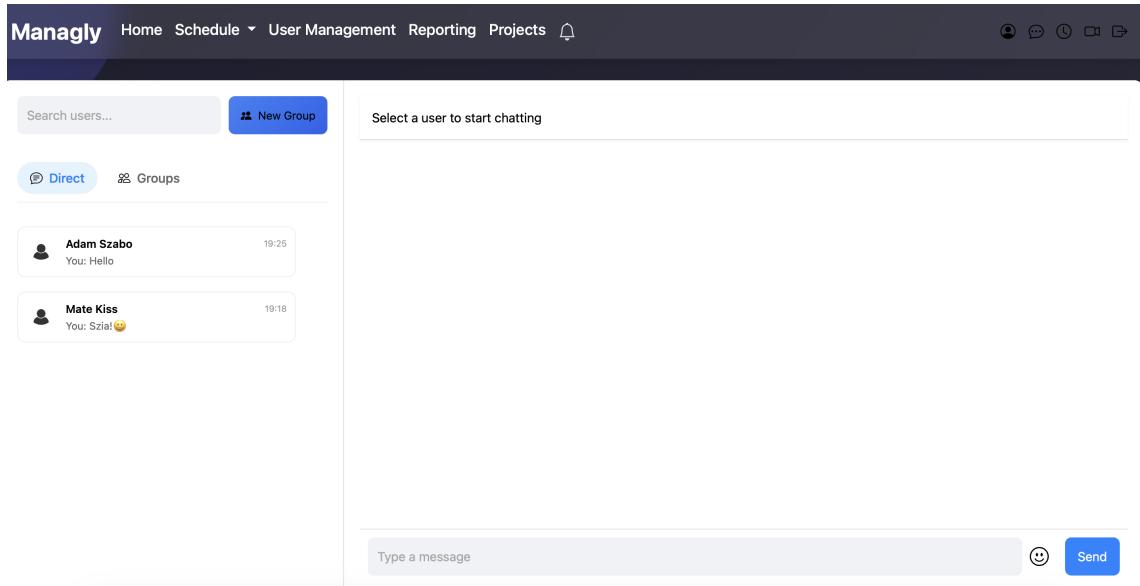
Minden projekt egy külön kártyán jelenik meg, amely tartalmazza a projekt nevét, a felhasználó szerepkörét a projektben (például Project Manager), a projekt állapotát és prioritását. Az állapot és a prioritás színkódolással is ellátott, amely segíti a gyors vizuális azonosítást. minden projekt kártyáján megtalálható egy haladásjelző sáv, amely vizuálisan ábrázolja a projekt befejezettségének százalékos arányát. Ez az információ lehetővé teszi a felhasználók számára, hogy gyorsan felnérjék, mennyire haladnak a projektjeikkkel. Ha a felhasználó még nem tagja egyetlen projektnek sem,

akkor a rendszer egy üzenetet jelenít meg, amely tájékoztatja a felhasználót, hogy még nincs besorolva projektekre.

2.5.7. Csevegés Funkció

2.5.7.1. Áttekintés

A Managly rendszer csevegés (chat) funkciója lehetővé teszi a vállalat munkatársai közötti közvetlen és csoportos kommunikációt egy egyszerű, mégis hatékony felületen keresztül. Ez a modul valós idejű üzenetváltást biztosít, amely jelentősen növeli a csapatmunka hatékonyságát és a kollégák közötti együttműködést. A modern, felhasználóbarát kialakítás egyszerűbbé teszi a minden nap kommunikációt, miközben számos fejlett funkciót is tartalmaz.



2.36. ábra. Csevegés felülete

2.5.7.2. Csevegés Felület Felépítése

A csevegés oldal kettő fő részre tagolódik: a bal oldali beszélgetési lista és a középső üzenetváltási terület. Ez az elrendezés biztosítja, hogy a felhasználók könnyen áttekinthessék aktív beszélgetéseiket, miközben kényelmesen folytathatnak új beszélgetéseket.

Beszélgetési Lista A bal oldali panel tartalmazza az összes aktív beszélgetést, beleértve mind az egyéni, mind a csoportos csevegéseket. A beszélgetések időrendi

sorrendben jelennek meg, a legfrissebb üzenettel rendelkezők a lista tetején. minden beszélgetésnél látható:

- A beszélgetőpartner vagy csoport neve
- A beszélgetőpartner profilképe vagy a csoport ikonja
- Az utolsó üzenet előnézete

A beszélgetési lista tetején található a keresőmező, amely lehetővé teszi a felhasználók számára, hogy gyorsan megtalálják a keresett kollégákat vagy csoportokat. A keresés a felhasználók neve alapján történik, és már néhány karakter beírása után megjelennek a találatok.

Üzenetváltási Terület A középső panel az aktív beszélgetés üzeneteit jeleníti meg. Az üzenetek időrendi sorrendben jelennek meg, a régebbi üzenetek felül, az újabbak alul. minden üzenetnél látható:

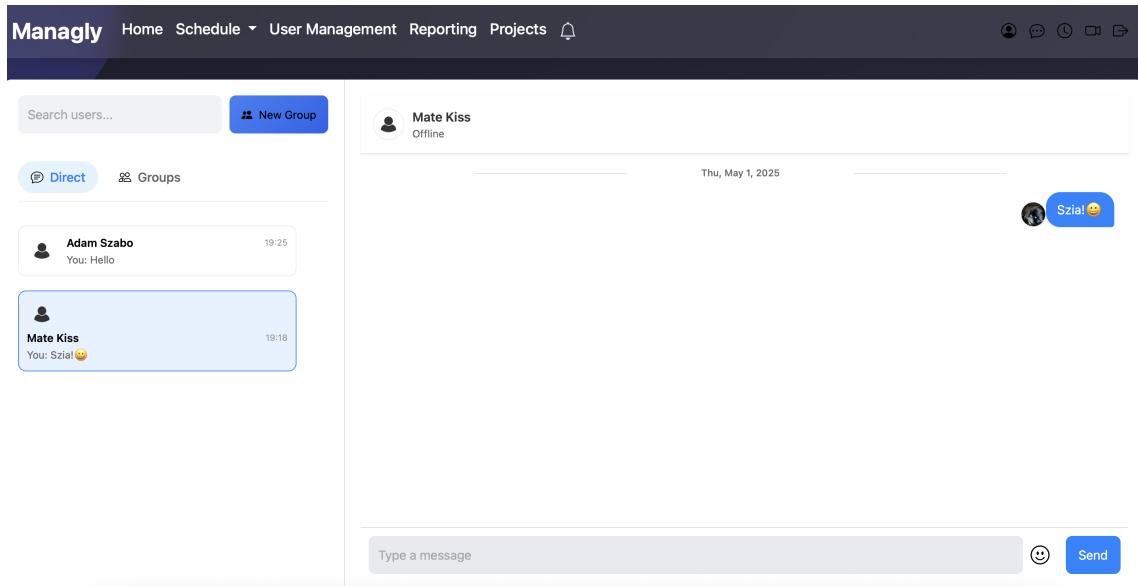
- A küldő neve és profilképe (csoportos beszélgetésekben)
- Az üzenet tartalma

Az üzenetváltási terület alján található a szövegbeviteli mező, ahol a felhasználók megírhatják és elküldhetik üzeneteiket. A szövegbeviteli mező mellett található a küldés gomb, valamint opcionális funkciógomb, például az emoji beszúrása.

2.5.7.3. Egyéni Beszélgetések

Az egyéni beszélgetések lehetővé teszik a felhasználók közötti közvetlen kommunikációt. Ezek a beszélgetések privátak, csak a résztvevők láthatják az üzeneteket. Új beszélgetés megkezdéséhez a felhasználónak a keresőmezőben név szerint rá kell keresnie a másik felhasználóra és a találatok közül kiválasztva a megfelelő felhasználót azonnal megjelenik a jobb oldalon a beszélgetés ablak.

Üzenetek Küldése és Fogadása Az üzenetek küldése rendkívül egyszerű: a felhasználó beírja az üzenetét a szövegbeviteli mezőbe, majd a küldés gombra kattint vagy megnyomja az Enter billentyűt. Az elküldött üzenet azonnal megjelenik a beszélgetésben, és a rendszer valós időben továbbítja azt a címzettnek.



2.37. ábra. Csevegés felülete egy kiválasztott felhasználóval

Amikor a felhasználó új üzenetet kap, az azonnal megjelenik a megfelelő beszélgetésben, ha az éppen nyitva van. Ha a beszélgetés nincs megnyitva, a rendszer értesítést küld, és az olvasatlan üzenetek száma megjelenik a beszélgetési listában az adott beszélgetés mellett.

Üzenetek Törlése és Visszavonása A felhasználók törölhetik saját üzeneteiket vagy kérhetik azok visszavonását. Az üzenetek törlése kétféleképpen történhet:

- Törlés csak saját magam számára: Az üzenet eltűnik a felhasználó beszélgetéséből, de a partner továbbra is látja.
- Törlés mindenki számára: Az üzenet eltűnik minden fél beszélgetéséből, helyette egy "Az üzenet törölve" jelzés jelenik meg.

Az üzenet törlése vagy visszavonása az üzenetre jobb gombbal kattintva kezdeményezhető.

2.5.7.4. Csoportos Beszélgetések

A csoportos beszélgetések lehetővé teszik három vagy több felhasználó közötti kommunikációt. Ezek a beszélgetések ideálisak csapatmunka, projektek vagy részletek közötti együttműködés esetén.

Csoport Létrehozása Új csoport létrehozásához a felhasználónak a "Új csoport" vagy hasonló gombra kell kattintania a beszélgetési lista tetején. Ezután meg kell adnia a csoport nevét, majd kiválasztania a csoport tagjait a vállalat dolgozóinak listájából. A csoport létrehozása után a rendszer automatikusan létrehozza a csoportos beszélgetést, és értesíti az összes tagot a csoportba való felvételükről.

A csoport létrehozója automatikusan a csoport adminisztrátorává válik, ami bizonyos jogosultságokat biztosít számára a csoport kezelésében. Ezek közé tartozik a tagok hozzáadása vagy eltávolítása, a csoport nevének módosítása és adminisztrátori jogok adása más csoporttagoknak.

Üzenetek Kezelése Csoportokban A csoportos beszélgetésekben az üzenetek küldése és fogadása hasonlóan működik, mint az egyéni beszélgetésekben. A különbség az, hogy minden üzenet mellett megjelenik a küldő neve és profilképe, hogy egyértelműen azonosítható legyen, ki küldte az adott üzenetet. A csoportos beszélgetésekben küldött üzeneteket minden csoporttag megkapja.

Csoporthoz köthető funkciók A csoport adminisztrátorai több speciális jogosultsággal rendelkeznek a csoport kezelésére:

- Tagok hozzáadása: Új felhasználók meghívása a csoportba
- Tagok eltávolítása: Meglévő tagok eltávolítása a csoportból
- Csoport nevének módosítása: A csoport megjelenített nevének megváltoztatása
- Adminisztrátori jogok kezelése: Más tagok kinevezése adminisztrátorrá vagy adminisztrátori jogok visszavonása

A csoportbeállítások a csoport nevére való kattintáskor érhető el, amikor egy csoportos beszélgetés van kiválasztva. Itt jelenik meg a csoport tagjainak listája is, ahol az adminisztrátorok speciális jelöléssel vannak ellátva.

2.5.7.5. Online Állapot Jelzése

A Managly csevegés funkciója valós időben jelzi a felhasználók online állapotát, ami segíti a hatékony kommunikációt. A felhasználók online állapota vizuálisan jeleznik meg a profilképek mellett. Zöld jelzés esetén a felhasználó elérhető, szürke jelezés

esetén a felhasználó nem elérhető. Az online állapot automatikusan frissül, amikor a felhasználó bejelentkezik vagy kijelentkezik a rendszerből. Az állapotjelzők minden felhasználóhoz megjelennek a beszélgetési listában.

2.5.8. Munkaidő Nyilvántartás

A Managly rendszer Munkaidő Nyilvántartás modulja átfogó megoldást kínál a munkaidő követésére egyéni és csapatszinten. A felhasználók rögzíthetik munkába érkezésüket és távozásukat, követhetik munkaóráikat, az adminisztrátorok pedig átláthatják a csapattagok adatait, támogatva a hatékony időgazdálkodást és pontos elszámolást.

2.5.8.1. Felület Felépítése

Az oldal kétpártos szerkezetű rendelkezik, amelyen a felhasználó két fő között válthat: Saját idő (minden felhasználó számára elérhető) és Csapat áttekintés (csak Admin és Manager jogosultsággal). A rendszer automatikusan ellenőrzi a jogosultságokat és megjeleníti vagy elrejti a Csapat áttekintés fület.

2.5.8.2. Személyes Munkaidő Követés

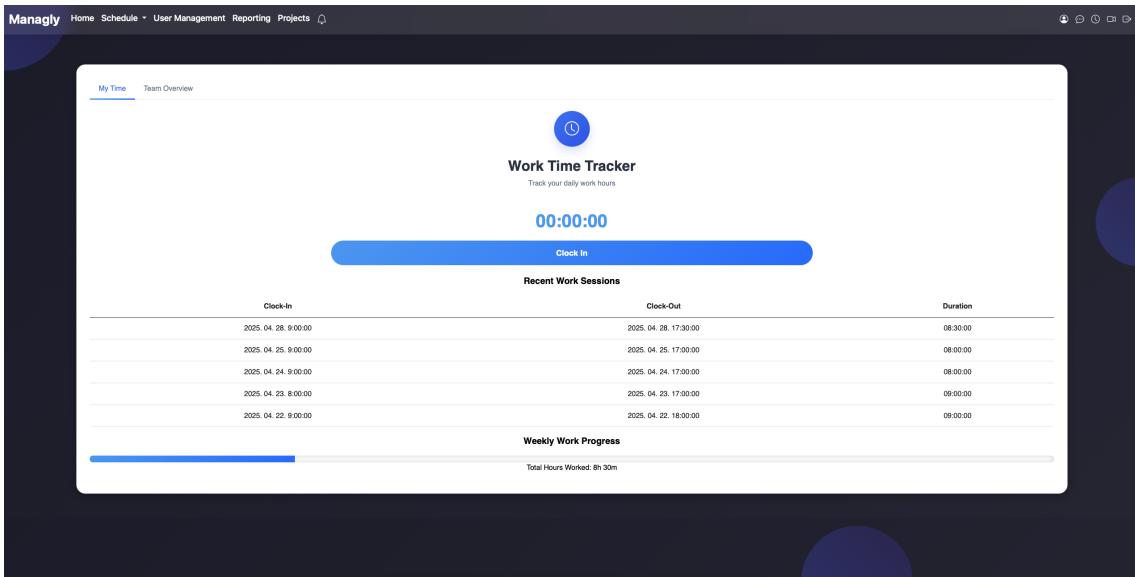
A Saját idő fülön található a munkaidő rögzítéshez szükséges időmérő (digitális óra 00:00:00 formátumban), valamint kontextustól függően a Bejelentkezés vagy Kijelentkezés gomb. Bejelentkezéskor a rendszer rögzíti a kezdés időpontját, elindítja az időmérőt (zöldre váltva), míg kijelentkezéskor rögzíti a befejezés idejét és visszaállítja az időmérőt alaphelyzetbe (kék színűre).

Ha a felhasználó kijelentkezés nélkül lép ki, majd később visszatér, a rendszer automatikusan ellenőrzi az aktív munkamenet állapotát és folytatja az időmérést. A fül második részén a korábbi munkamenetek listája található (utolsó 5, időrendi sorrendben) a kezdés és befejezés időpontjával, valamint az időtartammal. Az oldal alján haladásjelző sáv vizualizálja a heti ledolgozott órák arányát a teljes munkahét-hez (40 óra) viszonyítva.

2.5.8.3. Csapat Munkaidő Áttekintés

A Csapat áttekintés fülön a vezetők fejlett szűrési lehetőségekkel (keresőmező név szerint, állapotszűrő) tekinthetik át a csapattagok adatait. A szűrők alatt három

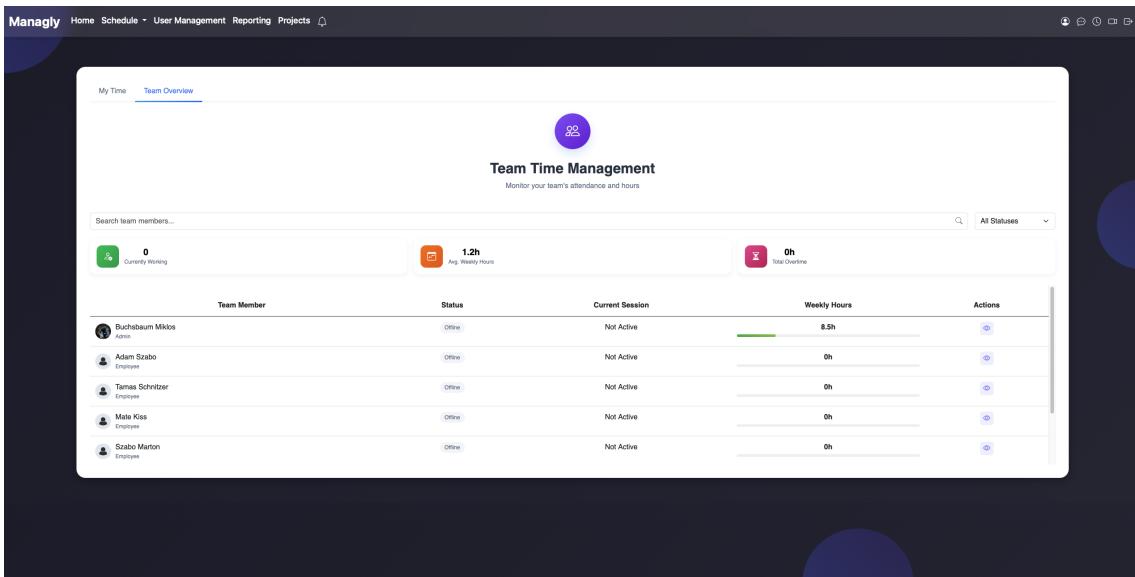
2. Felhasználói dokumentáció



2.38. ábra. Személyes munkaidő követés felülete

informatív kártya jeleníti meg a csapat munkaidő statisztikáit: jelenleg dolgozók száma, átlagos heti óraszám, és összes túlóra.

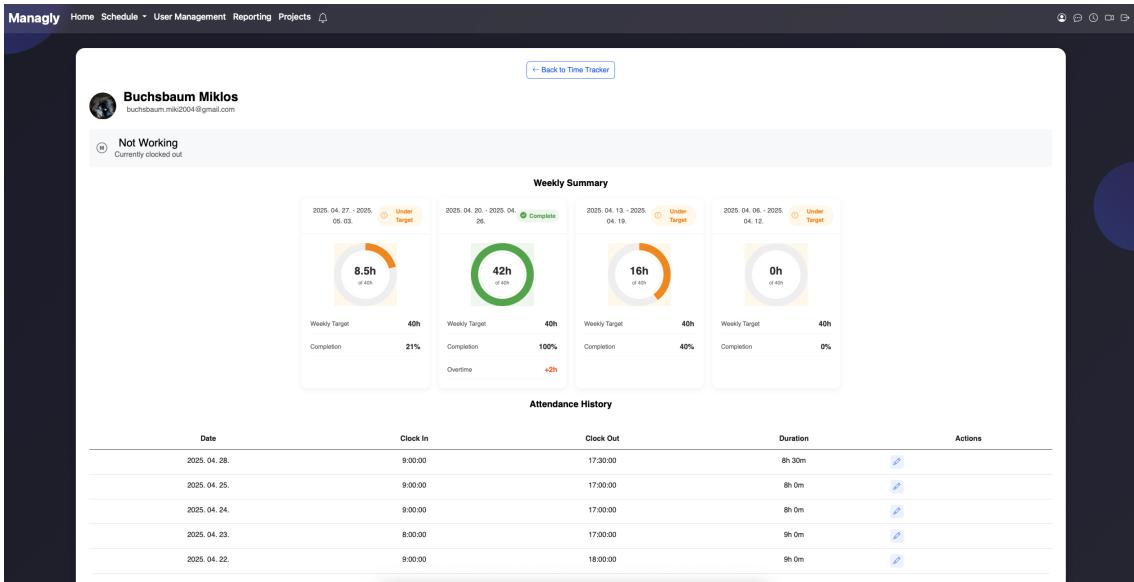
A fül fő részét a csapattagok táblázata adja, amely tartalmazza a munkatárs nevét és pozíóját, jelenlegi munkavégzési állapotát, az aktív munkamenet időtartamát valós időben frissülő időmérővel, a heti órák számát vizuális jelzéssel a normál és túlórákról, valamint a részletes adatok megtekintéséhez szükséges műveleti gombot.



2.39. ábra. Csapat munkaidő áttekintés felülete

2.5.8.4. Részletes Felhasználói Nézet

Az alkalmazott részletes munkaidő oldalán az alapvető információk mellett átfontosított összesítések találhatók különböző időtávokban: napi bontásban az elmúlt napok ledolgozott órái, és heti összesítésben az elmúlt hetek munkaórái az esetleges túlórák jelzésével. Az oldal részletes történetet mutat az alkalmazott korábbi munkameneleiről, beleértve a kezdés és befejezés időpontját, valamint az időtartamot.



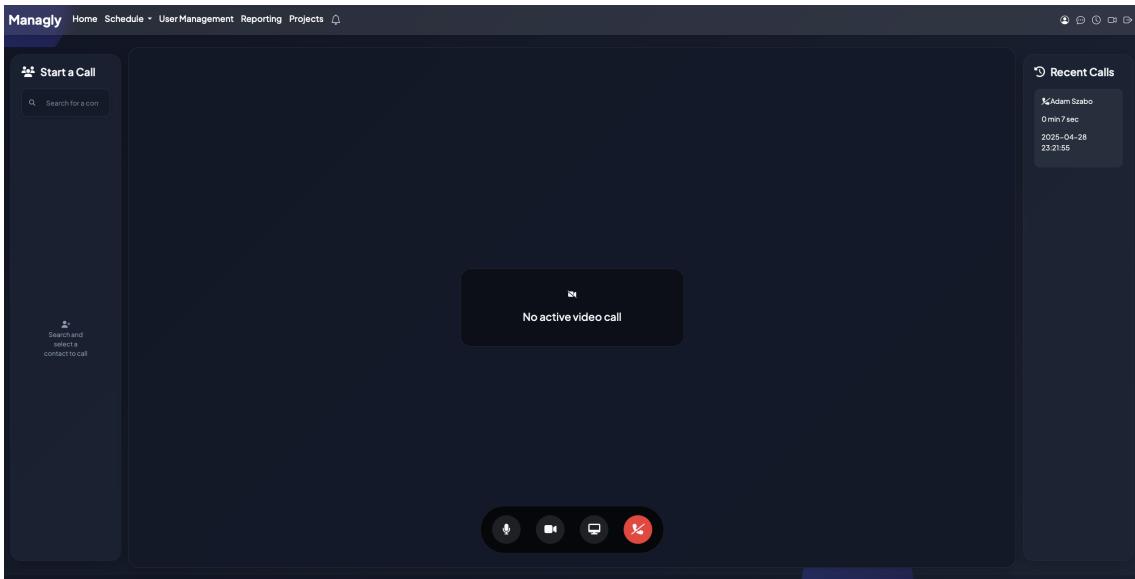
2.40. ábra. Részletes felhasználói nézet

2.5.8.5. Munkaidő Szerkesztés

Admin és Manager jogosultsággal rendelkező felhasználók szerkeszthetik a munkaidő-bejegyzéseket olyan esetekben, amikor manuális korrekció szükséges (például ha egy alkalmazott elfelejtett bejelentkezni vagy kijelentkezni). minden módonról a rendszer automatikus naplóbejegyzést készít, amely tartalmazza a változatás részleteit és a módosítást végző adminisztrátor azonosítóját.

2.5.9. Videóhívás Rendszer

A Managly rendszer Videóhívás modulja egy WebRTC technológián alapuló komplett megoldás a cégen belüli valós idejű videókommunikációra, amely integrált képernyőmegosztást és hangtovábbítást kínál ergonomikus, három paneles elrendezésben.



2.41. ábra. A Videóhívás felület átfogó nézete a három paneles elrendezéssel

2.5.9.1. Felület Felépítése

A videóhívás felület három fő panelból áll:

A bal oldali panel a hívások kezdeményezéséért felel, tartalmazza a keresőmezőt valós idejű szűréssel, a kapcsolat kiválasztást, a hívásindító gombot és a várakozásjelzőt. A felhasználó a keresőmezőbe gépelve azonnal láthatja a megfelelő cégen belüli felhasználókat, kiválaszthatja a hívni kívánt személyt, és elindíthatja a hívást.

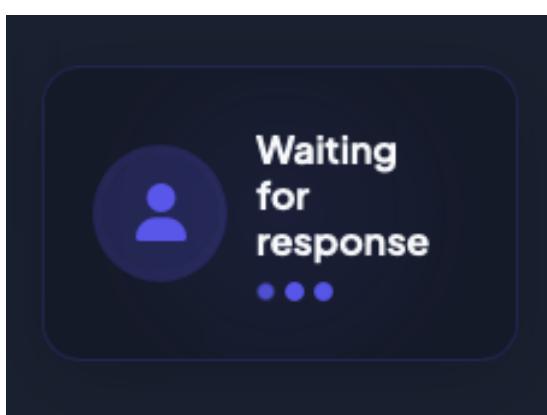
A középső panelen helyezkedik el a videóhívás terület a fő videóképpel (beszélgetőpartner vagy képernyőmegosztás), a saját kamera kisebb, áthelyezhető képével, és a hívásvezérlő gombokkal (mikrofon, kamera kapcsolók, képernyőmegosztás, hívás befejezése). Itt jelennek meg a bejövő hívás értesítések és a visszacsatlakozási lehetőség is az esetlegesen megszakadt aktív hívásokhoz.

A jobb oldali panel megjeleníti a korábbi hívások időrendi listáját a nevekkel, időpontokkal és eredményekkel (befejezett, nem fogadott), valamint a jelenlegi hívási kísérletek állapotinformációit. A rendszer minden hívás után automatikusan frissíti ezt a panelt.

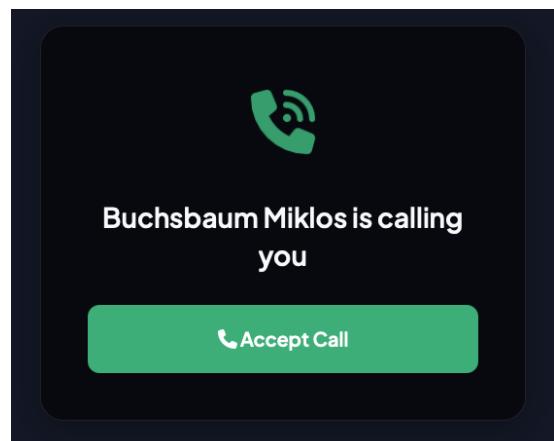
2.5.9.2. Fő Funkcionalitás

A videóhívás kezdeményezésekor a felhasználó megkeresi és kiválasztja a hívni kívánt személyt, majd a hívásindító gombra kattint. A rendszer értesíti a hívott felet, aktiválja a hívó kameráját és mikrofonját (engedélykérés után), és várakozási állapo-

tot jelez. Ha a hívás egy percen belül nem kerül fogadásra, a rendszer automatikusan megszakítja a kísérletet.



2.42. ábra. Kimenő hívás kezdeményezése és várakozási képernyő



2.43. ábra. Bejövő hívás értesítés a fogadó felhasználó oldalán

Bejövő hívásnál a rendszer értesítést jelenít meg a hívó nevével. Az elfogadás után automatikusan létrejön a videókapcsolat, aktiválódik a fogadó fél kamerája és mikrofonja, és megjelennek a megfelelő videoképek. A rendszer rögzíti a hívás kezdeti időpontját a későbbi időtartam-számításhoz.

Képernyőmegosztáskor a felhasználó kiválaszthatja a megosztani kívánt tartalmat (teljes képernyő vagy alkalmazás ablak), amely a másik fél számára fő videótartalom válik, míg a kameraképek kisebb méretben jelennek meg. A megosztás a gomb újbóli megnyomásával szüntethető meg.

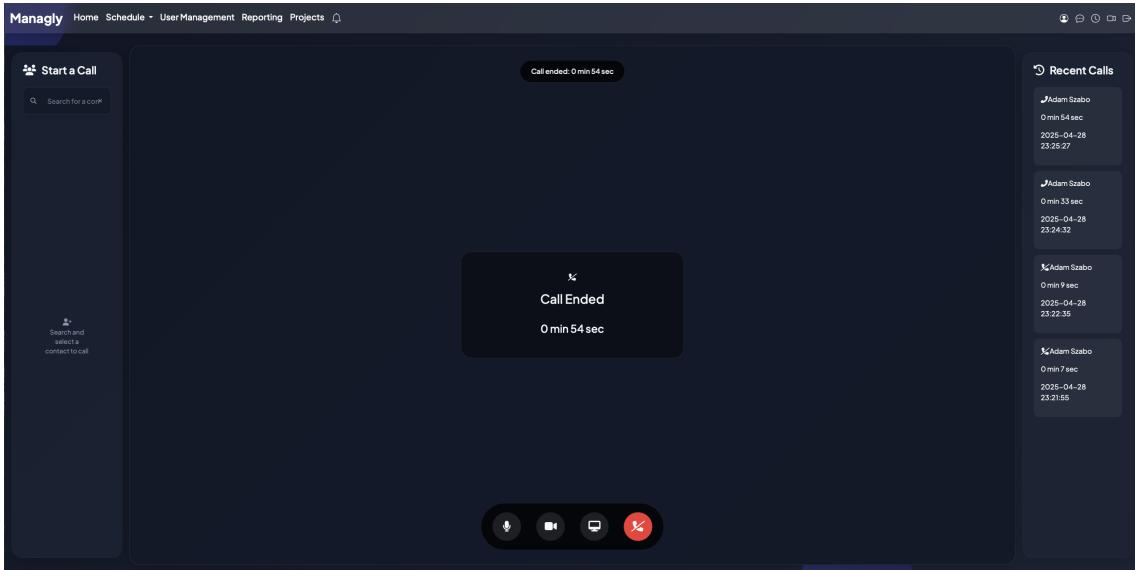
A hívás befejezése történhet a befejezés gombra kattintással, a böngészőablak bezárásával, vagy hosszabb hálózati kapcsolatvesztés esetén automatikusan. A rendszer rögzíti a befejezés időpontját, kiszámítja és megjeleníti a hívás időtartamát, majd kikapcsolja a kamerát és mikrofont.

2.5.9.3. Értesítések és Kapcsolatkezelés

A rendszer értesítéseket küld a bejövő hívásokról, nem fogadott hívásokról és hívás állapotváltozásokról, melyek a felhasználói felületen és a rendszer általános értesítési területén is megjelennek.

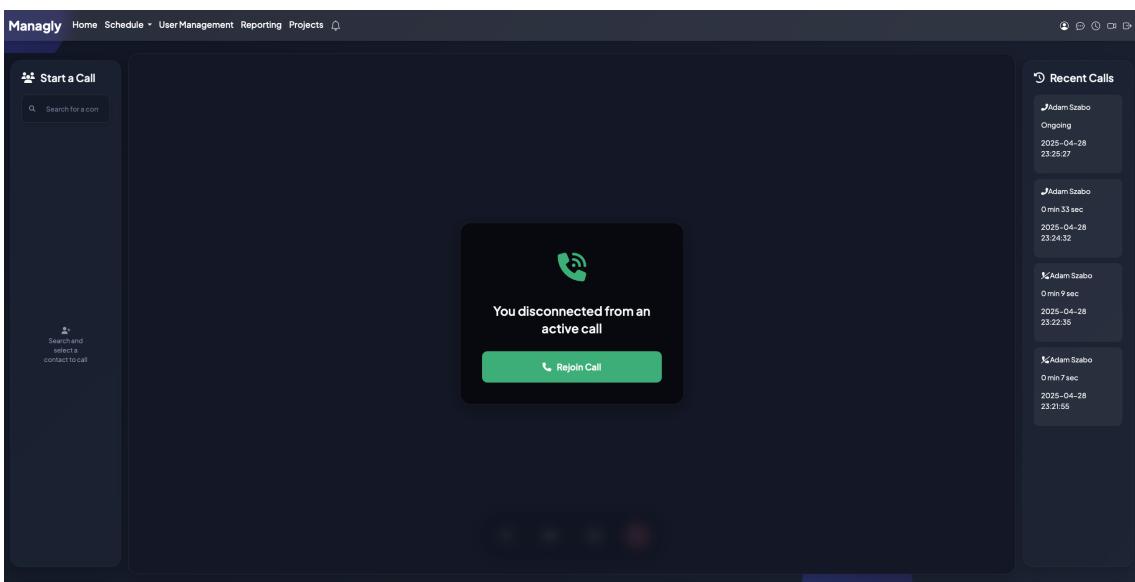
Hálózati kapcsolat megszakadásakor a rendszer jelez a felhasználónak, és a kapcsolat helyreállása után automatikusan vagy felhasználói kezdeményezésre újraépíti a hívást. Ha a felhasználó kilép a videóhívás oldalról aktív hívás közben, majd vissza-

2. Felhasználói dokumentáció



2.44. ábra. Hívás befejezése utáni összegző képernyő

tér, a rendszer felajánlja az újracsatlakozás lehetőségét, biztosítva a hívás folytatását véletlenszerű oldalbezárás után is.

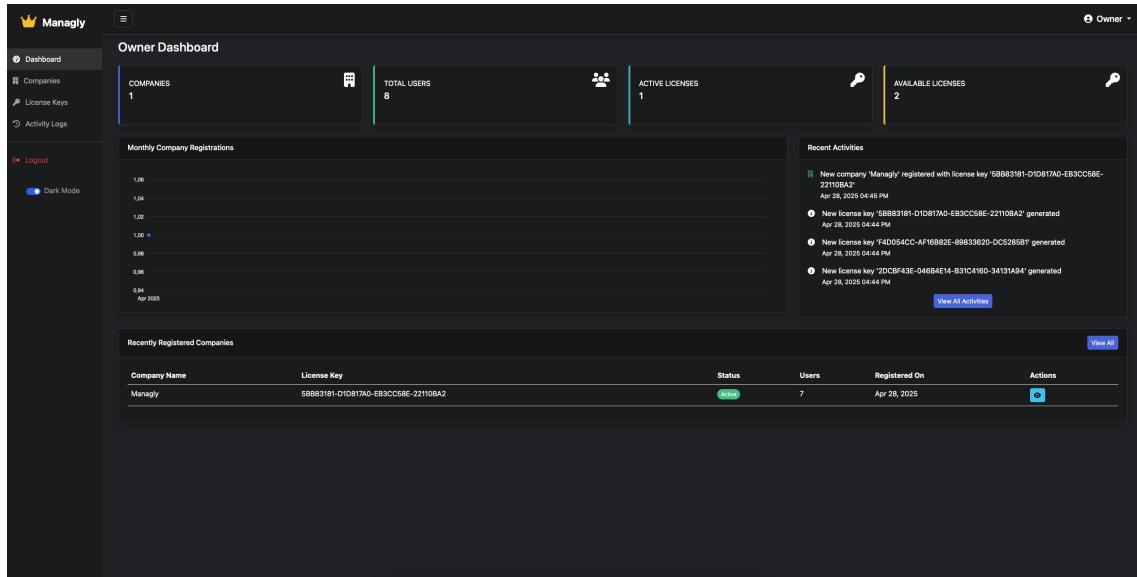


2.45. ábra. Újracsatlakozási lehetőség felajánlása

2.6. Owner felület

A Managly rendszer Tulajdonosi Kezelőfelülete egy átfogó irányítópultot biztosít a szoftver tulajdonosai számára a regisztrált vállalatok, licenckulcsok és kapcsolódó aktivitások nyomon követésére és kezelésére. Ez a felület kizárálag a "Owner"

jogosultságú felhasználók számára elérhető, és lehetővé teszi a teljes rendszer felügyeletét.



2.46. ábra. A Tulajdonosi Kezelőfelület áttekintése

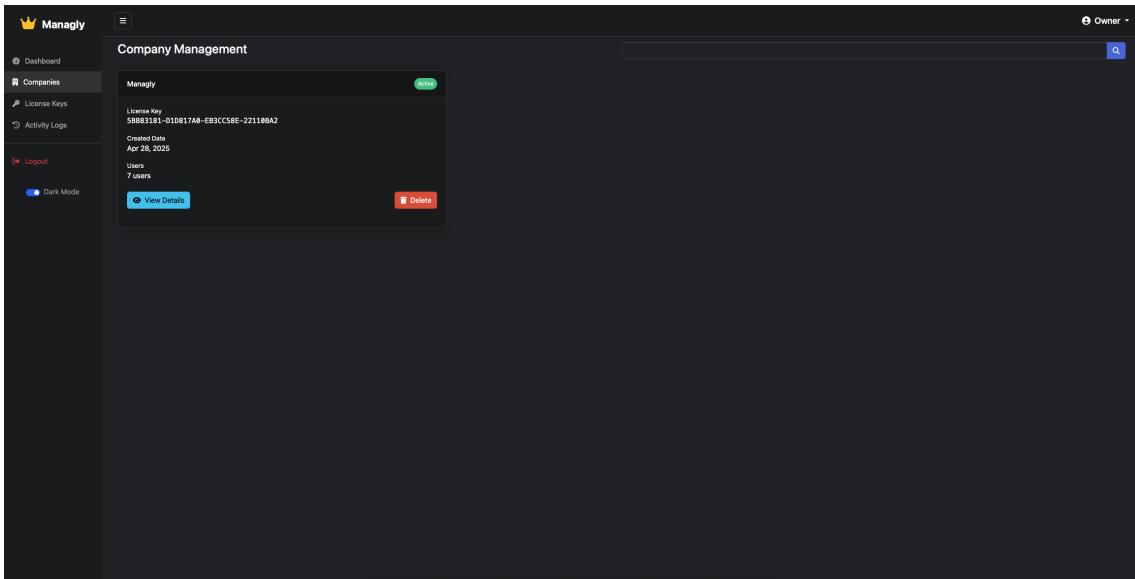
2.6.1. Irányítópult (Dashboard)

Az irányítópult központi oldala négy összesítő kártyával kezdődik, melyek a vállalatok és felhasználók számát, valamint az aktív és elérhető licencek mennyiségét jelenítik meg. Ez alatt egy vonaldiagram mutatja az elmúlt hat hónap vállalati regisztrációinak trendjét, segítve a növekedés és licencigény előrejelzését.

Az oldal jobb oldalán a legfrissebb tulajdonosi tevékenységek listája található, amely a rendszerben végzett műveleteket (vállalat regisztrációk/törlések, licenckulcs műveletek) jeleníti meg ikonokkal, leírással és időponttal ellátva. Az oldal alsó részén található táblázat a legutóbb regisztrált vállalatok adatait (név, licenckulcs, állapot, felhasználószám, regisztráció dátuma) és a kapcsolódó műveleteket teszi elérhetővé.

2.6.2. Vállalatok Kezelése

Ez az oldal lehetővé teszi az összes regisztrált vállalat kezelését egy keresőmezővel ellátott felületen. A vállalatok kártyás elrendezésben jelennek meg, feltüntetve a legfontosabb információkat: vállalat neve, licenc állapota, licenckulcs, létrehozás dátuma, felhasználók száma, lejáratú dátum és műveleti gombok.



2.47. ábra. Vállalatok kezelése kártyás elrendezéssel és keresőmezővel

A vállalat törlésekor a rendszer megerősítést kér, figyelmeztetve a felhasználót a következményekre (felhasználók törlése, licenckulcs felszabadítása, adatok végeleges törlése). Sikeres törlés után a rendszer értesítést jelenít meg és a licenckulcs ismét elérhetővé válik.

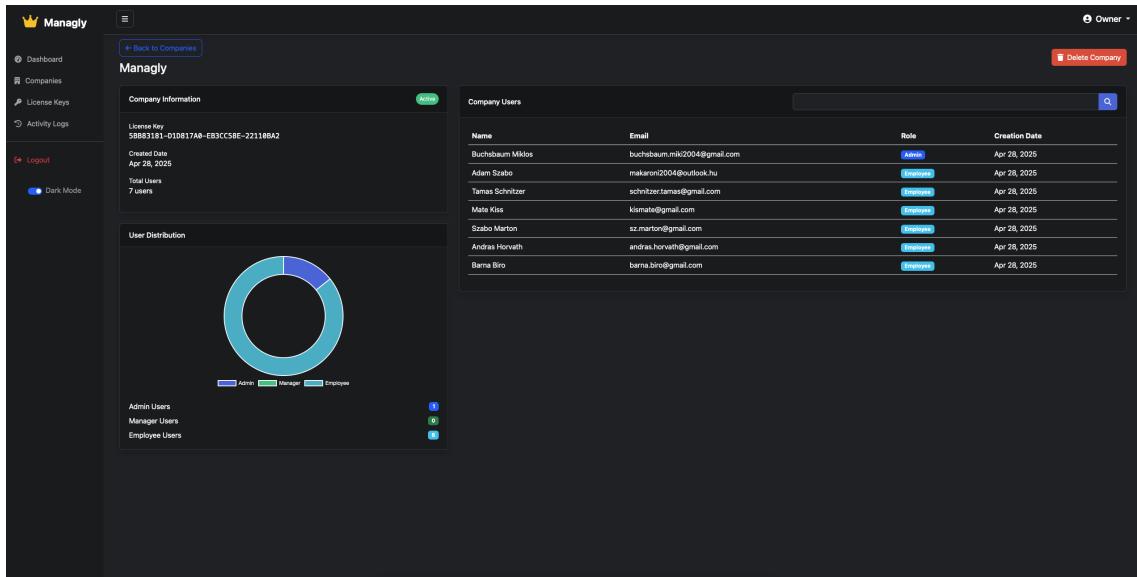
2.6.3. Vállalati Részletek

A vállalati részletek oldal bal oldalán panel jeleníti meg a vállalat alapvető információit (név, licenc állapota, kulcs, létrehozás és lejárat dátuma, felhasználók száma). Ez alatt egy fánk diagram és számadatok mutatják a felhasználók szerepkör szerinti eloszlását (adminisztrátorok, vezetők, alkalmazottak).

Az oldal jobb oldalán egy kereshető táblázat listázza a vállalathoz tartozó felhasználókat nevükkel, e-mail címükkel, szerepkörükkel és létrehozási dátumukkal. Innen is elérhető a vállalat törlésének lehetősége, amely folyamata megegyezik a Vállalatok kezelése oldalon leírtakkal.

2.6.4. Licenckulcsok Kezelése

Ez az oldal négy összesítő kártyával kezdődik, melyek az aktív, elérhető, lejárt és összes licenckulcs számát jelenítik meg. Alatta keresőmező található a kulcsok szűréséhez kulcsérték, állapot vagy vállalatnév alapján.



2.48. ábra. Vállalati alapinformációk és felhasználói szerepkör eloszlás

A licenckulcsok táblázatban jelennek meg, feltüntetve a kulcs értékét, állapotát, kiosztását, létrehozási és lejáratú dátumát, valamint a kapcsolódó műveleteket (szerkesztés, törlés). Új licenckulcs a "Licenckulcsok generálása" gombra kattintva hozható létre, opcionálisan megadott lejáratú dátummal.

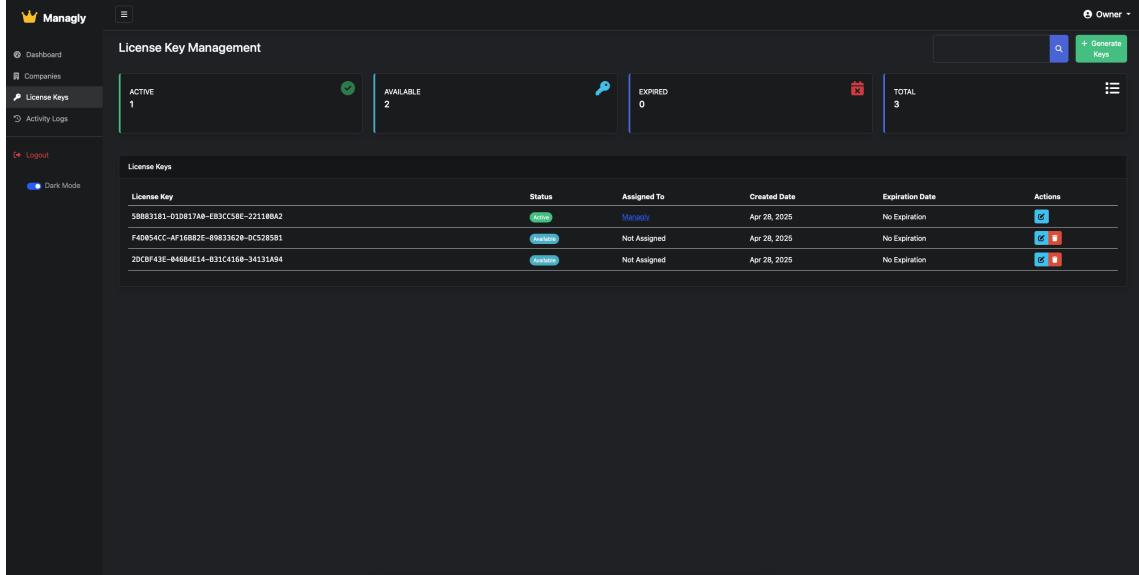
A licenckulcsok szerkesztésekor módosítható azok állapota és lejáratú dátuma, figyelembe véve az állapotváltoztatás következményeit (pl. aktívról elérhetőre változtatás a kulcs visszavonásával jár). Csak elérhető vagy lejárt állapotú kulcsok törölhetők, aktív kulcsok nem.

2.6.5. Tevékenységnapló

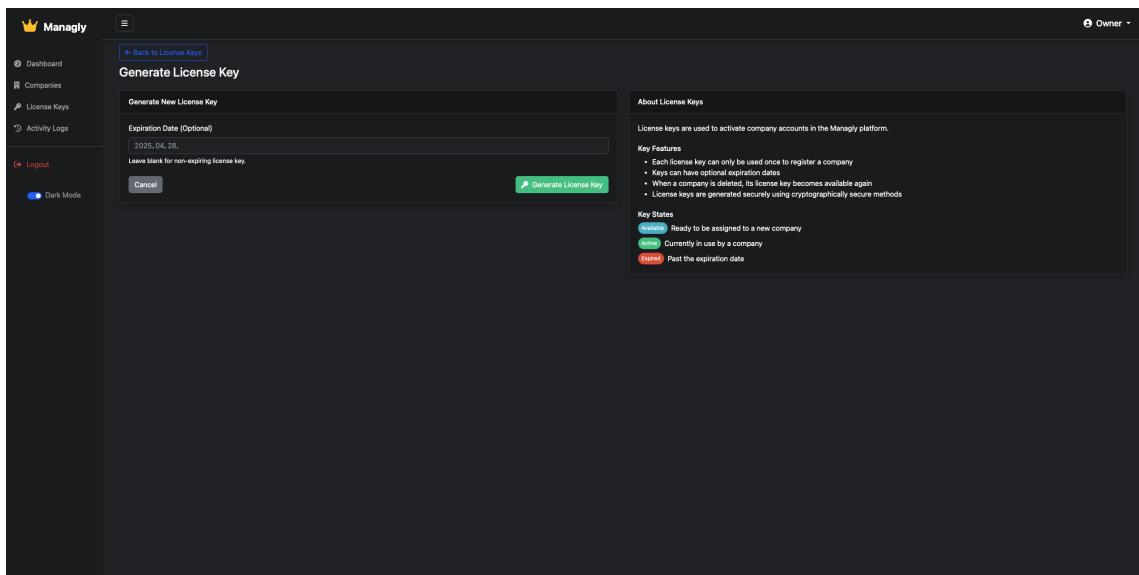
A tevékenységnapló oldal részletes áttekintést nyújt a tulajdonosi műveletekről. Az oldal tetején szűrőpanel található a tevékenységek típus, dátumtartomány és rendezés szerinti szűréséhez, valamint egy keresőmező a tevékenységleírás, típus vagy vállalatnév alapú kereséshez.

A szűrők alatt egy táblázat jeleníti meg a tevékenységnaplót ikonokkal jelölt tevékenységtípusokkal, leírással, időponttal és az érintett vállalattal, valamint a szűrési feltételeknek megfelelő tevékenységek számával.

2. Felhasználói dokumentáció

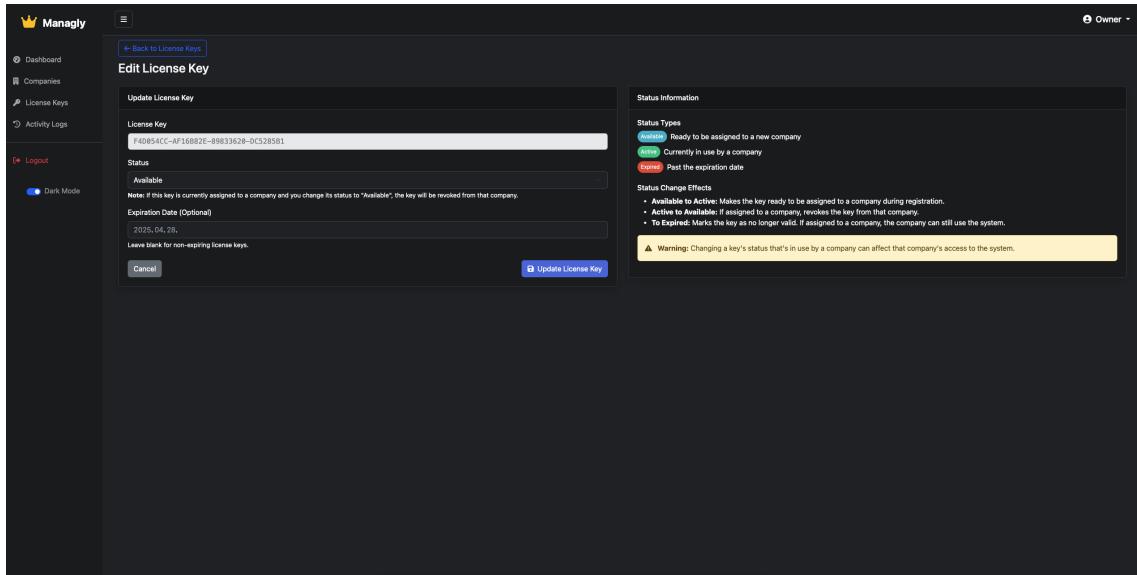


2.49. ábra. Licenckulcsok összesítő kártyái és keresési lehetőségek



2.50. ábra. Licenckulcsok létrehozása

2. Felhasználói dokumentáció



2.51. ábra. Licenckulcsok szerkesztése

Activity Logs			
Filter Activities		Date Range	Sort By
Activity Type	Description	Date & Time	Company
Activity Log			
Activity	Description	Date & Time	Company
Company Registered	New company 'Managly' registered with license key '6B883181-D1D817A0-E83CC58E-2210BA2'	Apr 28, 2025 04:45 PM	Managly
LicenseKeyGenerated	New license key '98883181-D1D817A0-E83CC58E-2210BA2' generated	Apr 28, 2025 04:44 PM	
LicenseKeyGenerated	New license key 'F4D554CC-AF16882E-89833620-DC528381' generated	Apr 28, 2025 04:44 PM	
LicenseKeyGenerated	New license key '2DCBF43E-04884E14-B31C4160-34131A64' generated	Apr 28, 2025 04:44 PM	

2.52. ábra. Tevékenységnapló áttekintése

3. fejezet

Fejlesztői dokumentáció

3.1. A megoldandó feladat

A Managly rendszer egy komplex vállalati menedzsment alkalmazás, amely lehetővé teszi a vállalatok számára a belső munkafolyamatok digitalizálását és hatékony kezelését. A rendszer fő célja a projekt-, idő-, és humánerőforrás-menedzsment integrálása egy egységes platformon, miközben valós idejű kommunikációs lehetőségeket is biztosít a felhasználók számára.

3.1.1. A felhasználói felületre vonatkozó elvárások

A felhasználói felület tervezésénél a következő követelményeket kellett figyelembe venni:

- **Reszponzivitás:** Az alkalmazásnak különböző képernyőméretekben és eszközökön is megfelelően kell működnie.
- **Esztétika:** Legyen esztétikus. A felületen alkotó komponensek készüljenek egységes séma alapján. A térközök, margók mérete szintén legyen konzisztens.
- **Intuitív használat:** A felhasználói felületnek könnyen tanulhatónak és használhatónak kell lennie, minimális betanítási idővel.
- **Valós idejű frissítések:** A felhasználói felületnek képesnek kell lennie a valós idejű adatfrissítések megjelenítésére, különösen a kommunikációs és értesítési funkciók esetében.

- **Jogosultsági szinteknek megfelelő nézetek:** A különböző jogosultsági szinttel rendelkező felhasználóknak eltérő funkcionálitást és nézeteket kell biztosítani.

3.1.2. A szerveroldali komponensre vonatkozó elvárások

A szerveroldali komponens fejlesztésénél a következő követelményeket kellett figyelembe venni:

- **Skálázhatóság:** A rendszernek képesnek kell lennie növekvő felhasználószám és adatmennyiség kezelésére.
- **Biztonság:** Többszintű biztonsági modell implementálása, beleértve a szerepkör-alapú hozzáférést és az adatvédelmi előírások betartását.
- **Hatókony adatkezelés:** Az adatbázis-műveletek optimalizálása a gyors válaszidők biztosítása érdekében.
- **Valós idejű kommunikáció:** A szervernek támogatnia kell a valós idejű adatcserét a kliensekkel (chat, videóhívás, értesítések).
- **Többfelhasználós környezet:** Konkurens hozzáférés kezelése és az adatintegritás biztosítása.
- **Licenckezelés:** A vállalati licencek és hozzáférések menedzselése.

3.2. Fejlesztői környezet

A Managly alkalmazás fejlesztéséhez a következő technológiákat és keretrendszegeket használtuk:

- **ASP.NET Core MVC:** A szerveroldali alkalmazás alapja, amely biztosítja a Model-View-Controller architektúrát.
- **Entity Framework Core:** ORM (Object-Relational Mapping) keretrendszer az adatbáziskezeléshez.
- **Identity Framework:** Felhasználókezelési és hitelesítési keretrendszer.
- **SignalR:** Valós idejű kommunikációs könyvtár a szerver és a kliensek között.

- **JavaScript/jQuery/Bootstrap:** Frontend fejlesztéshez használt technológiák.
- **WebRTC:** Böngésző-alapú valós idejű kommunikáció a videóhívások megvalósításához.
- **SQL Server/MySQL:** Relációs adatbáziskezelő rendszer.

A fejlesztés során a következő eszközöket használtuk:

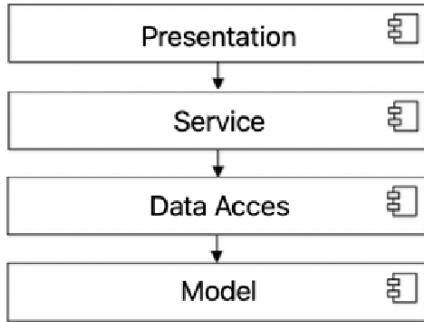
- **Visual Studio:** Elsődleges IDE a backend fejlesztéshez.
- **Visual Studio Code:** Frontend fejlesztéshez és gyors módosításokhoz.
- **Git:** Verziókezelő rendszer.
- **NuGet:** .NET csomagkezelő.

3.3. Az alkalmazás rétegei és felelősségi köreik

Az alkalmazás tervezésénél törekedtem az objektumorientált paradigmák követésére, tervezési minták használatára, illetve a SOLID elvek betartására. Ez utóbbi esetében különösen figyeltem arra, hogy ahol csak lehetőség van, konkrét implementációk helyett absztrakciók között állítsak fel függőségeket, és kihasználjam a függőség befecskendezés nyújtotta előnyökét.

1. Definíció. Függőség befecskendezés esetén a kliens biztosít egy átadási pontot a szolgáltatás interfésze számára, ahol a végrehajtás során a szolgáltatás megvalósítását adhatjuk át. Az átadási pontot lehet: konstruktor, metódus, interfész; a befecskendezést végző komponens lehet egy felsőbb réteg, a rétegződéstől független környezeti elem.

Összetettebb alkalmazások esetén érdemes több rétegű architektúrát alkalmazni, hiszen ezzel számos előnyhöz juthatunk. A felelősségi körök szétválasztásával kódunk sokkal átláthatóbbá válik, amely elősegíti annak bővíthetőségét és újrafelhasználhatóságát, de a tesztelhetőséget is jelentős mértékben növeli. A Managly alkalmazás 4 rétegből épül fel, melyek hierarchikus kapcsolatban állnak egymással. A 3.1 ábra az alkalmazás rétegeit és azok egymásra épülését szemlélteti.



3.1. ábra. Az alkalmazás rétegei és azok egymásra épülésének szemléltetése

3.3.1. A modell réteg (Model Layer)

A modell réteg képezi az alkalmazás alapját: minden más réteg függ a modell rétegtől, viszont a modell réteg nem rendelkezik függőséggel más rétegek felé. A Managly alkalmazásban a `Models` könyvtár tartalmazza ezt a réteget, amelynek felelőssége, hogy leírja az alkalmazás entitásait, azok tulajdonságait, illetve meghatározza az egyes entitások között fennálló kapcsolatokat. A modell réteg elemei a következők:

- Entitás modellek: `User`, `Company`, `Project`, `Tasks`, `Schedule`, `Attendance`, `Message`, `GroupChat`, `Leave`, stb.
- Segéd modellek: `Login`, `Register`, `ChangePassword`, `CreateProfile`, stb.
- Nézet modellek: `UserManagement`, `ProjectViewModels`, `DashboardLayout`, `ProjectMemberManagementViewModel`, stb.
- DTO-k (Data Transfer Objects): `LeaveUpdateDTO`, `CreateGroupDto`, `RoleUpdateViewModel`, stb.
- Felsorolás típusok: Az `Enums` mappában található típusok, mint például `NotificationType`, `LicensekeyStatus`, stb.

Az entitások elsősorban adatátviteli objektumként szolgálnak, meghatározva azt, hogy az adatbázisban milyen mezők szükségesek az eltárolásukra. Emellett kiszámolt tulajdonságok definiálására is lehetőséget nyújtanak, amelyek az adatbázisban nem kerülnek tárolásra. A modell entitásokhoz tartozó validációs attribútumok biztosítják az adatok integritását már a bevitel szintjén.

3.3.2. Az adatkezelési réteg (Data Access Layer)

Az adatkezelési réteg felelős az adatok perzisztens tárolásáért és az adatbázissal való kommunikációért. A Managly projektben a `Data` könyvtár tartalmazza ezt a réteget. Központi eleme az `ApplicationContext` osztály, amely az Entity Framework Core (EF Core) ORM-et használja az adatbázisműveletek elvégzésére. A függőség befecskendezés segítségével az adatbázis-kontextust a felsőbb rétegeknek átadjuk, így azok könnyedén végezhetnek adatbázis-műveleteket anélkül, hogy közvetlenül kellene foglalkozniuk az adatbázis-kezelés részleteivel.

Az adatkezelési réteg további elemei:

- Adatbázis inicializálás: `SeedData` osztály, amely a kezdeti teszt- és alapadatokat tölti be a rendszerbe
- Adatbázis migrációk: a `Migrations` könyvtárban található osztályok, amelyek az adatbázis séma változásainak kezelését biztosítják
- Entity konfigurációk: a modell entitások közötti kapcsolatok konfigurálása az `ApplicationContext` osztályban

Az EF Core objektum-relációs leképzés segítségével a C# objektumokat automatikusan relációs adatbázis táblákká alakítja, így a fejlesztés során elsősorban az objektumorientált paradigmára koncentrálhatsunk, az adatbáziskezelés technikai részleteit pedig az ORM kezeli.

2. Definíció. Az objektumrelációs leképzés (object-relational mapping - ORM) magas szintű transzformációját adja az adatbázisnak, amely a programban könnyen használható, ugyanakkor szabályozza az adatok kezelésének módját. A létrejött entitás osztályok adatokat tárolnak és azok közötti kapcsolatokat írnak le.

3.3.3. A szolgáltatási réteg (Service Layer)

A szolgáltatási réteg közvetítő szerepet tölt be az adatkezelési réteg és a prezentációs réteg között. A Managly alkalmazásban ez a réteg több komponensből áll össze, amelyek együtt biztosítják az üzleti logika működését:

- Valós idejű kommunikációs hubok: A `Hubs` könyvtárban található `ChatHub` és `VideoCallHub` osztályok a SignalR keretrendszer használják a felhasználók közötti valós idejű kommunikáció biztosítására.

- Segédosztályok: A `Helpers` könyvtárban találhatók különböző segédosztályok, mint például a `CountryCallingCodes`, amelyek az alkalmazás funkcionálitását támogatják.
- Szolgáltatások: Az alkalmazás különböző szolgáltatásai, amelyek a kontrollerek és a adatkezelési réteg között közvetítenek, bár ezek a Managly projektben közvetlenül nem különülnek el önálló könyvtárba, hanem többnyire a kontrollerekben valósulnak meg.

Az alkalmazásban kiemelt szerepet kap a függőség befecskendezés elve. A `Program.cs` fájlban regisztrált szolgáltatások interfészeken keresztül kommunikálnak egymással, ami növeli a kód modularitását és könnyebbé teszi a tesztelést. A SignalR hubok például központi szerepet játszanak a valós idejű funkciók megvalósításában, így a chat és videóhívás funkciók hatékonyan működhetnek.

3.3.4. A prezentációs réteg (Presentation Layer)

A prezentációs réteg felelős a felhasználói felület megjelenítéséért és a felhasználói interakciók kezeléséért. A Managly alkalmazásban ez a réteg két fő részből áll:

- Kontrollerek: A `Controllers` könyvtárban találhatók, mint például `HomeController`, `ProjectsController`, `ChatController`, `VideoConferenceController`, `ClockInController`, stb. Ezek kezelik a HTTP kéréseket, feldolgozzák az adatokat és átadják azokat a nézeteknek.
- Nézetek: A `Views` könyvtárban találhatók, a különböző kontrollerekhez tartozó alkönyvtárakban (pl. `Home`, `Projects`, `Chat`, `VideoConference`, `ClockIn`, stb.). Ezek a Razor nézetek határozzák meg a felhasználói felület megjelenését.
- Kliens oldali scriptek és stílusok: A `wwwroot` könyvtárban található `js` és `css` alkönyvtárak tartalmazzák a kliens oldali funkcionálitást biztosító JavaScript kódot és a CSS stíluslapokat.

A kontrollerek az MVC (Model-View-Controller) tervezési minta szerint működnek, ahol a kontroller fogadja a kéréseket, interakcióba lép a modellel, majd kiválasztja a megfelelő nézetet a válasz megjelenítéséhez. A Managly alkalmazásban a kontrollerek felelősek továbbá különböző API végpontok biztosításáért is, amelyeket a kliens oldali JavaScript kód használ az AJAX kérések során.

A prezentációs réteg részét képezi továbbá az RESTful API végpontokat biztosító API alkönyvtár a **Controllers** könyvtáron belül, amely lehetővé teszi a kliens oldali JavaScript számára az adatok aszinkron módon történő lekérését és módosítását.

Az alkalmazás rétegei tehát szigorú hierarchiában állnak egymással, ahol minden réteg csak az alatta lévő rétegtől függ, ami növeli a kód újrafelhasználhatóságát, karbantarthatóságát és tesztelhetőségét. A Dependency Injection (DI) konténer, amelyet az ASP.NET Core keretrendszer biztosít, lehetővé teszi a függőségek kezelését a különböző rétegek között, így a rendszer könnyen bővíthető és módosítható.

3.4. Perzisztencia

A prezentációs réteg a legfelső szint az alkalmazás architektúrájában, amely közvetlenül kommunikál a felhasználóval. Ez a réteg felelős a felhasználói interakciók kezeléséért és a megfelelő adatok megjelenítéséért. Komponensei:

- Kontrollerek: `HomeController`, `ProjectsController`, `ChatController`, stb.
- Nézetek: Razor nézetek a felhasználói felület megjelenítéséhez
- JavaScript fájlok: kliens oldali logika (`Chat.js`, `Projects.js`, stb.)
- CSS fájlok: stílusok és elrendezés

A prezentációs rétegen a kontrollerek kezelik a felhasználói kéréseket és továbbítják azokat a megfelelő szolgáltatásoknak. A kontrollerekben nincs üzleti logika, ezek csak a kérések fogadását és a válaszok visszaküldését végzik. Az MVC (Model-View-Controller) tervezési mintát követve a kontroller megkapja a kérést, a modellen keresztül elvégzi a szükséges műveleteket, majd a megfelelő nézetet adja vissza.

Az alkalmazás emellett támogatja az AJAX kéréseket és a valós idejű funkcionálitást a SignalR technológia segítségével, ami lehetővé teszi például a csevegés, videóhívások és valós idejű értesítések megvalósítását.

3.4.1. Adatkezelési réteg

Az adatkezelési réteg felel az állandó adattárolásért, a rendszer legalsó szintje. Ez a réteg kommunikál közvetlenül az adatbázissal, és különféle adatbázis-műveleteket hajt végre, például az entitások létrehozását, lekérdezését, frissítését és törlését

(CRUD műveletek). A Managly alkalmazás esetében az adatkezelési réteg Entity Framework Core (EF Core) segítségével valósul meg.

3.4.2. Entity Framework Core használata

A Managly alkalmazás a Microsoft Entity Framework Core (EF Core) objektum-relációs leképező (ORM) keretrendszerét használja az adatbázis-műveletek kezelésére. Az EF Core egy hatékony eszközkészlet, amely lehetővé teszi a fejlesztők számára, hogy adatbázis-entitásokkal objektumként dolgozzanak, ezáltal kiküszöböltve a nyers SQL-lekérdezések nagy részét.

A rendszer az `ApplicationDbContext` osztályt használja, amely az EF Core `IdentityDbContext<User>` osztályából származik. Ez a kontextus osztály szolgál hídként az alkalmazás és az adatbázis között.

Az `ApplicationDbContext` osztály a következő főbb funkciókat biztosítja:

- **Entitások definiálása:** A `DbSet<T>` tulajdonságok határozzák meg, hogy mely entitások képeződnek le adatbázistáblákként.
- **Kapcsolatok és konfigurációk:** Az `OnModelCreating` metódusban definiáljuk az entitások közötti kapcsolatokat, elsődleges kulcsokat, és egyéb konfigurációs beállításokat.
- **Adatbázis műveletek:** Az adatok lekérdezése, mentése, frissítése és törlése a kontextuson keresztül történik.

3.4.3. Adatbázis konfiguráció

Az alkalmazás a `Program.cs` fájlban konfigurálja az adatbázis-kapcsolatot, ahol a MySQL adatbázist használ. A kapcsolat a következőképpen van beállítva:

```
1 builder.Services.AddDbContext<ApplicationDbContext>(options =>
2     options.UseMySql(
3         builder.Configuration.GetConnectionString(
4             "DefaultConnection"),
5             new MySqlServerVersion(new Version(8, 0, 0))
6     ) );
```

3.1. forráskód. Adatbázis konfiguráció a `Program.cs` fájlban

A konfiguráció a következő elemeket tartalmazza:

- Az adatbázis kapcsolati string a konfigurációs fájlból kerül beolvasásra
- A MySQL adatbázisszerver verziója explicit módon meg van határozva
- A DbContext szolgáltatásként kerül regisztrálásra a Dependency Injection konténerbe

3.4.4. Migrációk

Az Entity Framework Core migrációs rendszere segítségével az alkalmazás adatbázissémája szinkronban tartható a modellek változásaival. A migrációk lehetővé teszik az adatbázis struktúrájának verziókövetését és a változások automatikus alkalmazását.

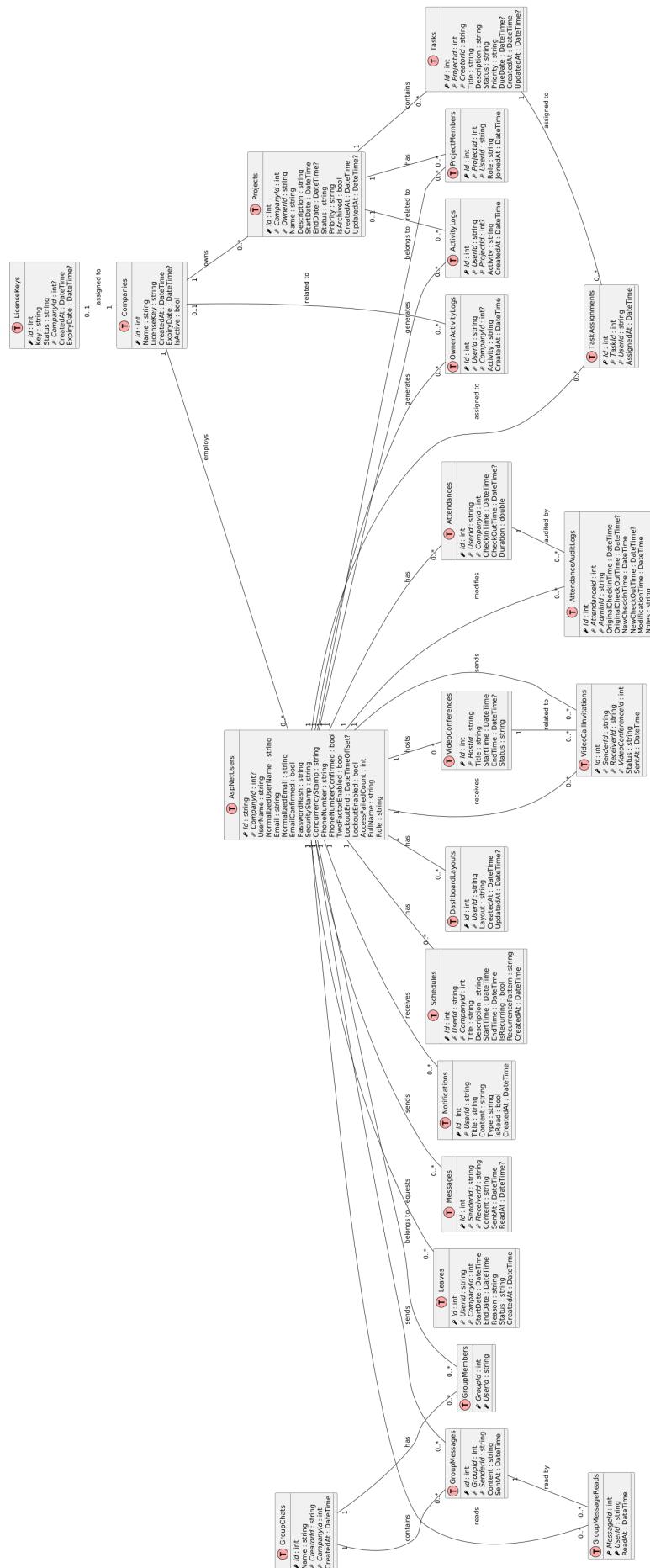
A Managly számos migrációt tartalmaz, amelyek az adatbázis fejlődését dokumentálják, kezdve az "InitialIdentitySetup" migrációtól a legutóbbi változtatásokig. Ezek a migrációk meghatározzák az adatbázis struktúráját, beleértve a táblákat, mezőket, indexeket, és kapcsolatokat.

3.4.5. Kapcsolatok és kezelésük

Az entitások közötti kapcsolatok az OnModelCreating metódusban és a modellek tulajdonságain keresztül vannak definiálva. Például:

- A User és Company közötti egy-a-sokhoz kapcsolat, ahol egy vállalathoz több felhasználó tartozhat
- A Project és Tasks közötti egy-a-sokhoz kapcsolat, ahol egy projekthez több feladat tartozhat
- Összetett kapcsolatok, mint például a TaskAssignment, amely kapcsolótáblájának szolgál a User és Tasks entitások között

Az Entity Framework Core automatikusan kezeli ezeket a kapcsolatokat, és lehetővé teszi a kapcsolódó entitások egyszerű lekérdezését, például Include() metódussal.



3.2. ábra. Az alkalmazás egyedkapcsolat diagramja SQL adatípusokkal

3.4.6. Lekérdezési optimalizáció

Az adatkezelési réteg felelős a hatékony lekérdezések biztosításáért. A Managly olyan technikákat alkalmaz, mint:

- A szükséges kapcsolatok szelektív betöltése az Include() metódussal
- Aszinkron műveletek használata a nagyobb adatmennyiségek kezelésére
- Szűrők és vetítések alkalmazása a lekérdezésekben a teljesítmény optimalizálása érdekében

Összefoglalva, az adatkezelési réteg az Entity Framework Core segítségével absztraktiokat biztosít az adatbázis-műveletek felett, ami leegyszerűsíti a fejlesztést és karbantartást, miközben gondoskodik az adatok hatékony és biztonságos kezeléséről.

3.5. Autentikáció

A Managly alkalmazás az ASP.NET Core Identity keretrendszer implementálja a felhasználói fiókok létrehozásának, kezelésének és hitelesítésének komplex folyamataihoz.

3.5.1. Felhasználói fiókok és szerepkörök

A rendszerarchitektúra négy hierarchikus szerepkört definiál, amelyek különböző hozzáférési és irányítási jogosultságokat biztosítanak: az Owner a teljes rendszer felügyelője, aki a vállalatok és licencek menedzselsére jogosult; az Admin vállalati szintű adminisztrációs feladatokat lát el a szervezeti felhasználók és beállítások kezelésével; a Manager projektvezetői kapacitásban működik, jogosultsága kiterjed projektek létrehozására és feladatok hozzárendelésére; míg az Employee alapszintű hozzáférésével elsősorban a rá osztott feladatokat hajtja végre. Ezek a szerepkörök egy jól strukturált jogosultsági keretet biztosítanak, amely lehetővé teszi a differenciált hozzáférés-szabályozást az alkalmazás funkcióihoz.

3.5.2. Jogosultságkezelés

A Managly komplex jogosultságkezelése két egymást kiegészítő mechanizmusra épül. A Role-based Authorization paradigmá szerint a felhasználók egy vagy több szerepkörhöz rendelhetők, amely meghatározza alapvető jogosultsági szintjüket; míg a Resource-based Authorization rendszer az erőforrások tulajdonságaira koncentrál, így biztosítva, hogy a felhasználók csak a számukra releváns entitásokhoz férjenek hozzá. Ez a kettős megközelítés különösen hatékony a több vállalatot kiszolgáló környezetben, ahol az adatok izolációja kulcsfontosságú.

A jogosultságok érvényesítése három architekturális szinten valósul meg: Controller szinten az AuthorizeAttribute attribútumok deklaratív módon korlátozzák a hozzáférést; Action szinten részletesebb, kontextus-függő jogosultság-ellenőrzés történik, amely figyelembe veszi a kérés paramétereit és a felhasználói állapotot; Nézet szinten pedig a felhasználói felület elemei dinamikusan, a felhasználó jogosultságainak megfelelően jelennek meg vagy rejtődnek el, így biztosítva a konzisztens felhasználói élményt.

3.5.3. Hitelesítési folyamat

A Managly hitelesítési protokollja több rétegű biztonsági modellt implementál. A felhasználók azonosítása elsősorban felhasználónév és jelszó kombinációján alapul, amelyet a rendszer szigorúan ellenőriz. A sikeres autentikáció után a felhasználói munkamenetet cookie-alapú technológia őrzi, amely időkorlátos hozzáférést biztosít. A jelszavak tárolása kiemelten biztonságos: a rendszer PBKDF2 hash algoritmust alkalmaz, sóval kiegészítve, ami hatékonyan véd a jelszó-visszafejtési kísérletek ellen. Az architektúra emellett előkészített a többféle hitelesítési módszer integrálására, beleértve a kétfaktoros autentikációt és a külső szolgáltatókon keresztüli bejelentkezést, amelyek a jövőbeli bővítések során implementálhatók.

3.6. Parancsok és lekérdezések

Az alkalmazás arkitektúrájának fontos részét képezi a parancsok és lekérdezések megfelelő strukturálása, amely biztosítja a rendszer skálázhatóságát és karbantarthatóságát. A Managly rendszerben az MVC architektúra alapján építettük fel a parancsok és lekérdezések kezelését, kiegészítve néhány speciális tervezési mintával.

3.6.1. Vezérlő-alapú kéréskezelés

A Managly alkalmazás az ASP.NET Core MVC keretrendszerre épül, ahol a kérések kezelése vezérlőkön (Controller) keresztül történik. minden funkcionális területhez külön vezérlő tartozik, amely a megfelelő műveletek végrehajtásáért felelős. A vezérlők a kéréseket akciók formájában fogadják, amelyek lehetnek parancsok (utasítások) vagy lekérdezések.

3.6.2. Hibakezelés és eredmények kezelése

A Managly rendszerben a parancsok és lekérdezések végrehajtása során különös figyelmet fordítottunk a hibakezelésre. A hibák kezelése több rétegben is megvalósul, biztosítva a rendszer robusztusságát.

A hibakezelés főbb komponensei:

- **Validáció:** A kérések adatainak ellenőrzése a ModelState és adatannotációk segítségével.
- **Try-catch blokkok:** A végrehajtás során fellépő kivételek kezelése.
- **Struktúrált hibaüzenetek:** A felhasználó számára érthető hibaüzenetek generálása.
- **Naplázás:** A hibák naplózása a rendszerben későbbi elemzés céljából.

A parancsok és lekérdezések visszatérési értékei strukturált formában kerülnek továbbításra, általában JSON formátumban, amely a dinamikus felhasználói felület frissítését teszi lehetővé AJAX kérések esetén.

3.6.3. DTO-modellek használata

A Managly rendszerben adatátviteli objektumokat (DTO - Data Transfer Object) használunk a vezérlők és a nézetek közötti kommunikáció során. Ez lehetővé teszi a domain modellek és a megjelenítési logika elkülönítését, valamint a szükséges adatok hatékony továbbítását.

Főbb DTO kategóriák a rendszerben:

- **Kérés DTO-k:** A felhasználói kérések adatainak továbbítására szolgálnak.

- **Válasz DTO-k:** A műveletek eredményeinek visszaadására használjuk őket.
- **Nézet modellek:** A megjelenítési réteg számára szükséges adatok strukturálását biztosítják.

A DTO-k és domain modellek közötti átalakítást a vezérlőkben végezzük el, biztosítva a domain logika elkülönítését a megjelenítési rétegtől.

3.6.4. Az MVC architektúra alkalmazása

A Managly rendszerben az MVC (Model-View-Controller) minta szolgál a parancsok és lekérdezések kezelésének alapjául:

- **Model:** A domain entitások és üzleti szabályok rétege.
- **View:** A felhasználói interfész rétege, amely a modellek adatait jeleníti meg.
- **Controller:** A kérések fogadásáért és feldolgozásáért felelős réteg.

A Managly rendszerben a vezérlők szerepe a kérések fogadása, validálása, a megfelelő üzleti logika meghívása, valamint a válaszok előkészítése és visszaadása. A vezérlők közvetlenül az adattárolási réteggel (Entity Framework Core) kommunikálnak.

```

1 // Controller
2 [HttpGet]
3 public async Task<IActionResult> GetProjects()
4 {
5     try
6     {
7         var currentUser = await _userManager.GetUserAsync(User);
8         if (currentUser == null)
9         {
10             return StatusCode(500, new { error = "Current user not
11                                         found" });
12         }
13
14         var projects = await _context.Projects
15             .Where(p => p.CompanyId == currentUser.CompanyId)
16             .Select(p => new { p.Id, p.Name })
17             .ToListAsync();
18     }
19 }
```

```

17         return Ok(projects);
18     }
19     catch (Exception ex)
20     {
21         return StatusCode(500, new { error = ex.Message });
22     }
23 }
24 }
```

3.2. forráskód. MVC minta alkalmazása a Managly rendszerben

3.6.5. Aszinkron műveletek

A Managly rendszerben az összes adatbázis-művelet és külső erőforrás elérés aszinkron módon történik, biztosítva a rendszer skálázhatósgát és hatékony erőforrás-kezelését. Ez a megközelítés lehetővé teszi a szerver számára, hogy a váratkozási idő alatt más kéréseket tudjon kiszolgálni.

```

1 [HttpGet]
2 public async Task<IActionResult> Profile()
3 {
4     var userId = User.FindFirstValue(ClaimTypes.NameIdentifier);
5     var user = await _userManager.FindByIdAsync(userId);
6
7     if (user == null)
8     {
9         return RedirectToAction("Login", "Home");
10    }
11
12    var model = new UserProfile
13    {
14        Id = user.Id,
15        Email = user.Email,
16        Name = user.Name,
17        LastName = user.LastName,
18        ProfilePicturePath = user.ProfilePicturePath ?? "/images/
19                      default/default-profile.png",
20        PhoneNumber = user.PhoneNumber,
21        DateOfBirth = user.DateOfBirth,
22        Gender = user.Gender,
23        Country = user.Country,
```

```

23         City = user.City,
24         Address = user.Address
25     };
26
27     var projects = await _context.ProjectMembers
28         .Include(pm => pm.Project)
29         .Where(pm => pm.UserId == userId && pm.Project.Status != "
30             Completed")
31         .Select(pm => new UserProjectViewModel
32         {
33             ProjectId = pm.ProjectId,
34             ProjectName = pm.Project.Name,
35             Role = pm.Role,
36             Status = pm.Project.Status
37         })
38         .ToListAsync();
39
40     model.Projects = projects;
41
42     return View(model);

```

3.3. forráskód. Aszinkron műveletek a Managly rendszerben

A Managly rendszer parancsok és lekérdezések kezelése az MVC architektúrán alapul, kiegészítve hatékony hibakezeléssel, DTO-k használatával és aszinkron műveletekkel. Ez a megközelítés biztosítja a rendszer skálázhatóságát, karbantarthatóságát és felhasználói élményének optimalizálását.

3.6.6. HTTP kérések és válaszok

A Managly alkalmazás RESTful API-t biztosít az adatok lekérdezéséhez és módosításához. A kontrollerek feldolgozzák a bejövő HTTP kéréseket, és megfelelő válaszokat küldenek vissza.

3.6.7. Kérések formátuma

Az API a következő HTTP metódusokat támogatja:

- **GET**: Adatok lekérdezése.

- **POST:** Új erőforrások létrehozása.
- **PUT:** Meglévő erőforrások frissítése.
- **DELETE:** Erőforrások törlése.

3.6.8. Válaszok formátuma

Az API válaszai JSON formátumban érkeznek, és a következő struktúrát követik:

- Sikeres válaszok: HTTP 200 OK, 201 Created, 204 No Content.
- Hibaválaszok: HTTP 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error.

3.6.9. Validáció és hibaüzenetek

Az API kérések validációja a következő szinteken történik:

- **Model validáció:** Data Annotations attribútumokkal.
- **Üzleti logikai validáció:** A szolgáltatás rétegen.
- **Hibakezelés:** Exception filterekkel és middleware-rel.

A hibaüzenetek egységes formátumban kerülnek visszaküldésre:

- Hibaüzenet részletes leírása.
- Hibakód a könnyebb azonosítás érdekében.
- Esetleges validációs hibák listája.

3.6.10. Valós idejű kommunikáció

A Managly rendszerben implementált SignalR hub-ok kulcsszerepet játszanak a valós idejű kommunikáció megvalósításában, biztosítva, hogy a felhasználók azonnal értesüljenek a releváns eseményekről és interakciókról anélkül, hogy manuálisan frissíteniük kellene az alkalmazást.

A Managly rendszerben implementált SignalR hub-ok kulcsszerepet játszanak a valós idejű kommunikáció megvalósításában, biztosítva, hogy a felhasználók azonNAL értesüljenek a releváns eseményekről és interakciókról anélkül, hogy manuálisan frissíteniük kellene az alkalmazást.

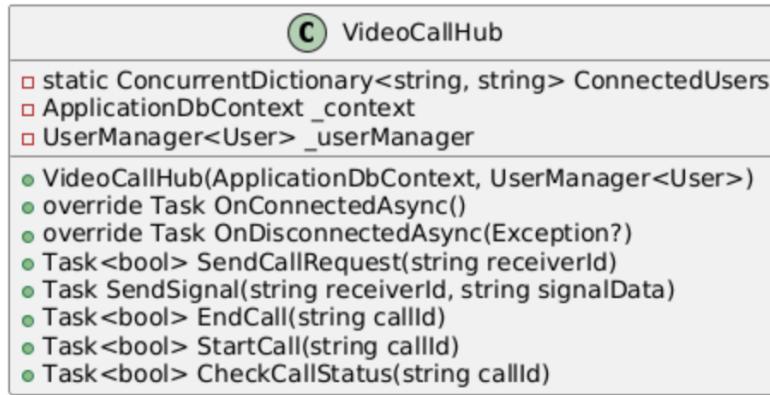
A ChatHub elsősorban a szöveges kommunikáció zavartalan működését szolgálja. Lehetővé teszi az azonnali üzenetküldést egyéni és csoportos beszélgetések formájában, ahol a küldött üzenetek azonnal megjelennek a címzett felhasználók felületén. Ezen túlmenően képes jelezni a felhasználók online állapotát és az olvasottsági státuszt, így a kommunikáció résztvevői minden tisztában lehetnek azzal, hogy üzenetük kézbesítve és olvasva lett-e. A hub kezelési logikája biztosítja azt is, hogy a felhasználók több eszközön vagy böngészőablakban is bejelentkezhetnek egyidejűleg, fenntartva a folytonos kapcsolatot a szerverrel.



3.3. ábra. A ChatHub UML diagramja

A VideoCallHub a valós idejű videohívások lebonyolításáért felelős. A WebRTC technológiával integrálva lehetővé teszi a hang- és videójelek továbbítását a felhasználók között minimális késleltetéssel. A hub közvetítő szerepet tölt be a hívás felépítésének fázisában, továbbítva a szükséges szignálokat, amelyek segítségével a peer-to-peer kapcsolat felépül a kommunikáló felek között. Képes hívási értesítéseket küldeni, jelezni a hívás állapotváltozásait, valamint kezelní a képernyőmegosztás funkcionálitást is, amely különösen hasznos munkahelyi környezetben és együttműködés során.

Mindkét hub integrálva van az értesítési rendszerrel, amely lehetővé teszi, hogy a felhasználók azonnal tájékoztatást kapjanak a beérkező hívásokról, üzenetekről vagy rendszerszintű eseményekről, függetlenül attól, hogy az alkalmazás melyik részét használják éppen. Ez az infrastruktúra jelentősen javítja a felhasználói élményt, mivel a valós idejű frissítések révén csökken a reakcióidő és nő az interakciók hatékonyisége.



3.4. ábra. A VideocallHub UML diagramja

A hub-ok implementációja során külön figyelmet kapott a skálázhatóság és a hibatűrő működés. A felhasználói kapcsolatok nyilvántartása thread-safe kollekciókban történik, amely lehetővé teszi a rendszer biztonságos működését magas terhelés mellett is, miközben a kapcsolatok megszakadása esetén automatikus újracsatlakozási mechanizmusok lépnek életbe.

3.7. A szerveroldal belépési pontja

A Managly alkalmazás szerveroldali komponense az ASP.NET Core keretrendszerre épül, amely modern, moduláris és nagy teljesítményű platform webalkalmazások fejlesztéséhez. A rendszer belépési pontja a `Program.cs` fájl, amely az alkalmazás inicializálásáért, a szolgáltatások konfigurálásáért és a HTTP kérések feldolgozási folyamatának beállításáért felelős.

Az alkalmazás indításakor a következő főbb lépések történnek:

1. A webszerver létrehozása és konfigurálása
2. A szolgáltatások regisztrálása a Dependency Injection konténerbe
3. Az adatbáziskapcsolat és az Entity Framework Core konfigurálása
4. A hitelesítési és jogosultságkezelési rendszer beállítása
5. Az útvonalak és a kéréskezelési csővezeték (pipeline) konfigurálása
6. Az alkalmazás indítása és a kérések fogadása

A Managly rendszer modularitását és konfigurálhatóságát a szolgáltatások regisztrációja biztosítja, ahol minden komponens függőségeit és beállításait deklaratív

módon definiáljuk. Ez a megközelítés lehetővé teszi a komponensek könnyű cseréjét és tesztelhetőségét, valamint a rendszer rugalmas skálázását.

3.7.1. Vezérlők

A Managly alkalmazásban a vezérlők (controllers) alkotják a fő belépési pontot a HTTP kérések feldolgozásához. minden funkcionális területhez dedikált vezérlő tartozik, amely az adott terület összes műveleteinek kezeléséért felelős. A rendszer moduláris szerkezetének köszönhetően a vezérlők logikusan szervezett egységeket alkotnak, amelyek jól definiált felelősségi körökkel rendelkeznek.

A főbb vezérlők és funkcióik a Managly rendszerben:

AccountController: Felhasználói fiók kezelése, beleértve a profil szerkesztését és a jelszóváltoztatást (3.5. ábra).



3.5. ábra. Az AccountController UML diagrammja

AdminController: Adminisztrátori műveletek végrehajtása, például felhasználók létrehozása, szerepkörök kezelése és vállalati beállítások módosítása (3.6. ábra).



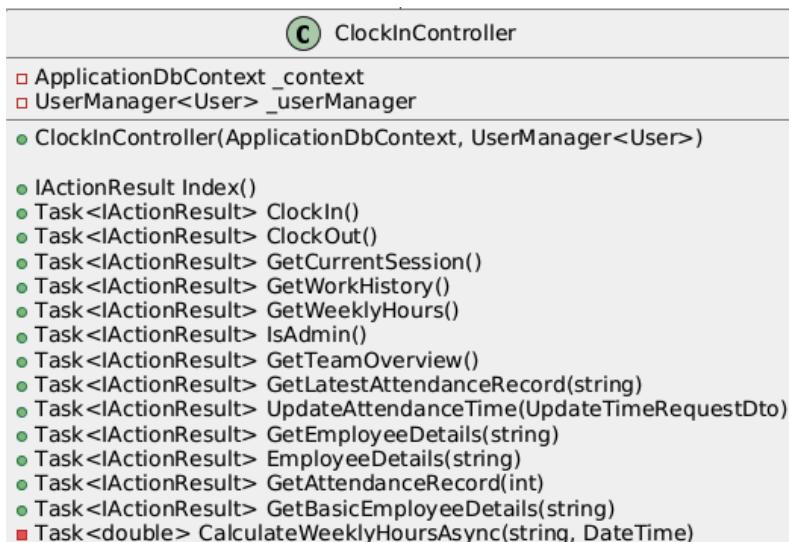
3.6. ábra. Az AdminController UML diagrammja

ChatController: A valós idejű üzenetküldési funkciók kezelése, beleértve az egyéni és csoporthoz beszélgetéseket (3.7. ábra).



3.7. ábra. A ChatController UML diagrammja

ClockInController: A munkaidő nyilvántartás műveletei, beleértve a bejelentkezést, kijelentkezést és a jelenlét statisztikák lekérdezését (3.8. ábra).



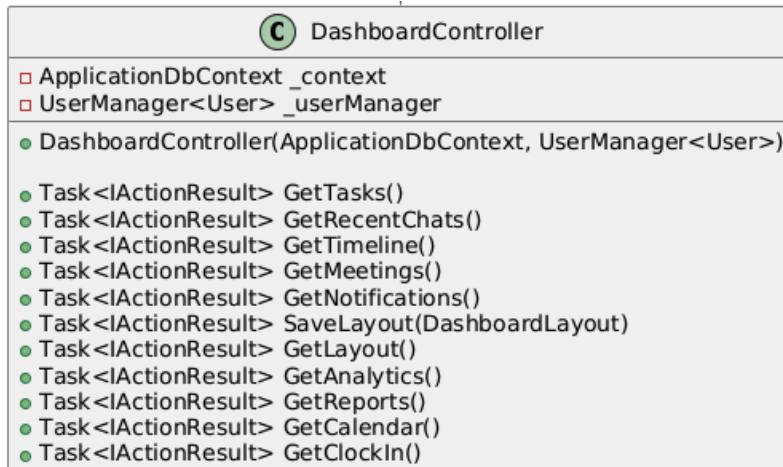
3.8. ábra. A ClockInController UML diagrammja

DashboardController: A személyre szabható irányítópult adatainak kezelése és megjelenítése (3.9. ábra).

HomeController: Az alkalmazás alapvető navigációs funkciói, beleértve a bejelentkezési és regisztrációs folyamatokat (3.10. ábra).

NotificationController: Az értesítési rendszer kezelése, beleértve az értesítések létrehozását, lekérdezését és olvasottként jelölését (3.11. ábra).

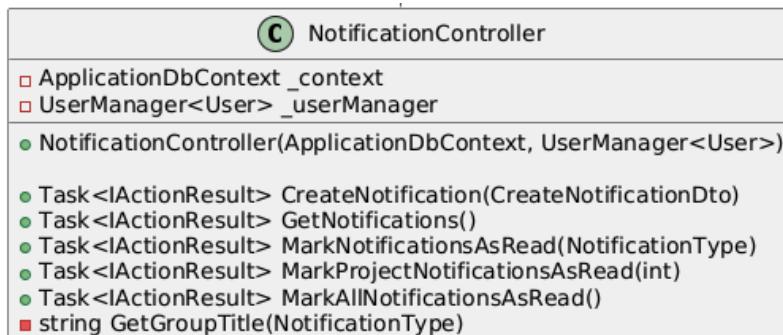
OwnerController: A rendszer tulajdonosának speciális műveletei, például licenckulcsok kezelése és tulajdonosi statisztikák lekérdezése (3.12. ábra).



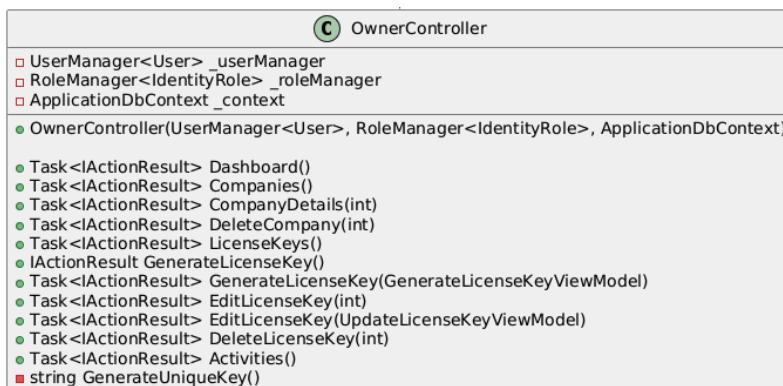
3.9. ábra. A DashboardController UML diagrammja



3.10. ábra. A HomeController UML diagrammja

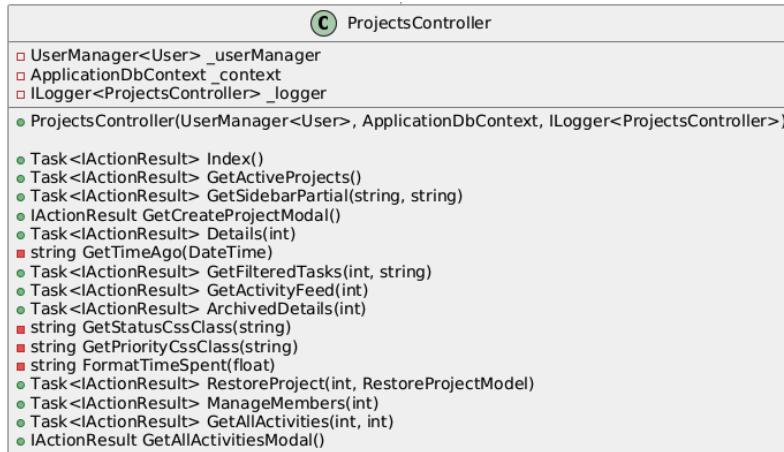


3.11. ábra. A NotificationController UML diagrammja



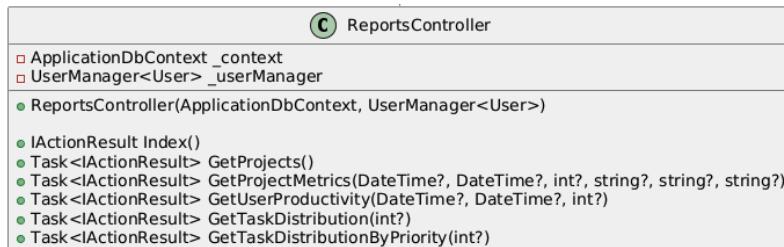
3.12. ábra. Az OwnerController UML diagrammja

ProjectsController: Projektek létrehozása, szerkesztése, törlése és a projekttagok kezelése (3.13. ábra).



3.13. ábra. A ProjectsController UML diagrammja

ReportsController: Különböző jelentések és statisztikák generálása a rendszer használatáról és a teljesítményről (3.14. ábra).



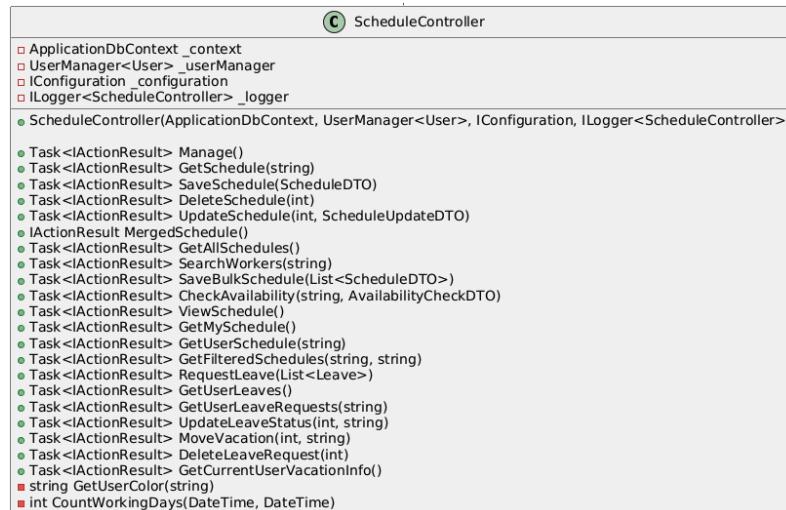
3.14. ábra. A ReportsController UML diagrammja

ScheduleController: Munkaidő-beosztás és szabadság kezelése, beleértve a szabadságkérelmek jóváhagyását és elutasítását (3.15. ábra).

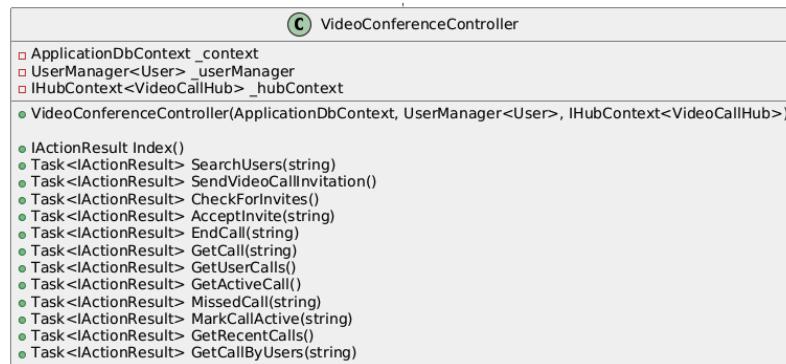
VideoConferenceController: Videóhívások és konferenciák kezelése, beleértve a hívások kezdeményezését és fogadását (3.16. ábra).

A vezérlők implementálják a RESTful architektúra elveit, ahol minden erőforrás egyedi URI-val rendelkezik, és a HTTP metódusok (GET, POST, PUT, DELETE) az erőforrásokon végrehajtandó műveletek típusát határozzák meg. Ez a megközelítés konzisztens interfész biztosít a különböző kliens alkalmazások számára, függetlenül attól, hogy azok böngészőben vagy mobilalkalmazásban futnak.

A vezérlők további fontos jellemzői a Managly rendszerben:



3.15. ábra. A ScheduleController UML diagramma



3.16. ábra. A VideoConferenceController UML diagramma

- Attribútum-alapú routing:** A kérési útvonalak közvetlenül a vezérlő osztályokon és metódusokon deklaráltak, egyszerűsítve az útvonalak kezelését és megértését.
- Szerepkör-alapú hozzáférés-szabályozás:** A vezérlők és műveletek védetek a jogosulatlan hozzáféréstől az **[Authorize]** attribútum használatával.
- Modellek validációja:** A kérések adatai automatikusan validálódnak a modell annotációk alapján, biztosítva az adatok integritását.
- Függőséginjektálás:** A vezérlők függőségei, például az adatbázis-kontextus vagy az azonosítási szolgáltatások, konstruktor injektálással kerülnek be, növelve a tesztelhetőséget és a kódminőséget.

3.7.2. HTTP kérések és válaszok felépítése

A Managly rendszerben a HTTP kérések és válaszok gondosan strukturált és szabványosított formátumot követnek, amely biztosítja a kliens és szerver közötti hatékony kommunikációt, valamint egyszerűsíti a hibaelhárítást és a fejlesztést.

A HTTP kérések fő komponensei:

- **URL:** A kért erőforrás azonosítója, amely az alkalmazás strukturáját tükrözi, például `/Projects/Details/5` a projektadatok lekéréséhez, vagy `/Reports/GetTeamStatistics` a csapatstatisztikák lekérdezéséhez.
- **HTTP metódus:** Az erőforráson végrehajtandó művelet típusa, amely lehet:
 - GET: Adatok lekérdezése (például projektek listája, felhasználói profil)
 - POST: Új erőforrás létrehozása (például új projekt, új felhasználó)
 - PUT/PATCH: Meglévő erőforrás módosítása (például profil frissítése)
 - DELETE: Erőforrás törlése (például projekt vagy feladat törlése)
- **Fejlécek:** Metaadatok a kérésről, beleértve a hitelesítési token-t, tartalomtípust és elfogadott válaszformátumokat.
- **Törzs:** A kérés adatai, általában JSON formátumban, amely tartalmazza az erőforrás létrehozásához vagy módosításához szükséges adatokat.

A HTTP válaszok fő komponensei:

- **Státuszkód:** Jelzi a kérés feldolgozásának eredményét:
 - 2xx: Sikeres kérés (pl. 200 OK, 201 Created, 204 No Content)
 - 3xx: Átirányítás (pl. 302 Found, 304 Not Modified)
 - 4xx: Kliensoldalú hiba (pl. 400 Bad Request, 401 Unauthorized, 404 Not Found)
 - 5xx: Szerveroldalú hiba (pl. 500 Internal Server Error)
- **Fejlécek:** Metaadatok a válaszról, beleértve a tartalomtípust, gyorsítótárazási irányelveket és CORS beállításokat.
- **Törzs:** A válasz adatai, amelyek lehetnek:

- HTML-tartalom a böngészőben való megjelenítéshez
- JSON-struktúrált adatok AJAX-kérések vagy API-hívások esetén
- Fájlok letöltések esetén
- Hibaüzenetek hiba esetén

A Managly rendszerben a válaszok konzisztens struktúrát követnek, különösen az API-kérések esetében, ahol általában a következő formátumot használjuk:

- Sikeres válasz: `{"success": true, "data": {...}, "message": "..."}`
- Hibás válasz: `{"success": false, "error": "...", "details": [...]}`

Ez a konzisztens formátum egyszerűsíti a kliensoldalon a válaszok feldolgozását és a hibakezelést, valamint javítja a fejlesztői élményt.

A Managly rendszerben különös figyelmet fordítunk a hibakezelésre, ahol a hibákat a megfelelő HTTP-státuszkódokkal és részletes hibaüzenetekkel jelezzük, segítve a fejlesztőket a hibák gyors azonosításában és kijavításában. Az alkalmazás globális hibakezelőt is tartalmaz, amely biztosítja, hogy a kezeletlen kivételek megfelelő hibaválaszokká alakuljanak, és ne vezessék az alkalmazás összeomlásához.

3.8. Kliens

A Managly alkalmazás kliensoldali komponense egy összetett, modern webalkalmazás, amely a felhasználói élményt és a hatékony működést helyezi előtérbe. A megvalósítás során fontos szempont volt a reszponzív kialakítás, amely biztosítja, hogy az alkalmazás különböző eszközökön (asztali számítógép, tablet, mobiltelefon) egyaránt jól használható legyen.

A kliens architektúra jól strukturált és moduláris, az ASP.NET Core MVC keretrendszer és a Razor nézetek kombinációjával, modern JavaScript és CSS technikákkal kiegészítve. Ez a megközelítés egyensúlyt teremt a szerveroldali renderelés előnyei és a kliens oldali dinamikus működés között, lehetővé téve a hatékony fejlesztést és karbantartást.

3.8.1. A felület tervezése

A Managly felhasználói felületének tervezése során a fő szempontok az egyszerűség, az intuitív használat és a hatékonyság voltak. A felület vizuális stílusában a

modern, letisztult megjelenést céloztuk, amely professzionális, de mégis barátságos benyomást kelt.

A Managly felhasználói felülete konzisztens vizuális nyelvezetre épül, amely egyesíti az ergonómiai alapelveket a modern design trendekkel. Az egységes színpaletta (kék árnyalatok professzionális megjelenéssel, semleges szürkék háttérként, figyelemfelhívó kiemelő színek), a következetes tipográfiai hierarchia (sans-serif betűtípusok optimalizált olvashatósággal) és az intuitív ikonrendszer biztosítja a felhasználói élémény folytonosságát. A vizuális hierarchia egyértelműen jelzi az elemek fontosságát, miközben a kártyaalapú elrendezés és az enyhe árnyékolás elősegíti a tartalom hatékony szegmentálását. minden interakció azonnali vizuális visszajelzést kap, a gyakran használt funkciók könnyen elérhetők, az intuitív elrendezés pedig minimalizálja a tanulási görbületet.

A reszponzív megjelenítés Bootstrap és egyedi CSS kombinációjával valósul meg, három fő nézetkategóriával: asztali (teljes funkcionalitás, összetett elrendezés), tablet (optimalizált középméretű képernyőkre) és mobil nézet (egyszöagos elrendezés, prioritizált tartalom, érintés-optimalizált vezérlőelemek). A design figyelembe veszi az akadálymentességi szempontokat, biztosítva a széles körű hozzáférhetőséget.

3.8.2. Állapotkezelés

A Managly többrétegű állapotkezelési stratégiája biztosítja az adatok konziszenciáját és a felhasználói élémény folytonosságát. A szerveroldali állapot (adatbázisban és munkamenetekben tárolva) képezi az alapot, amelyet kiegészít a kliens oldali állapotkezelés JavaScript segítségével. A rendszer kihasználja a böngésző localStorage és sessionStorage funkcióit a felhasználói preferenciák megőrzésére, míg az URL-alapú állapottárolás lehetővé teszi specifikus nézetek megosztását és mentését. Az AJAX technológia aszinkron felületfrissítést biztosít teljes újratöltés nélkül, a SignalR pedig valós idejű frissítéseket tesz lehetővé a kollaboratív funkciókban. Az intelligens gyorsítótárazási stratégia csökkenti a szerverigénybevételt és optimalizálja a teljesítményt, csak szükség esetén frissítve a tárolt adatokat.

3.8.3. Oldalak

A Managly funkcionalitása logikusan elkülönített oldalakra tagolódik. A bejelentkezési és regisztrációs oldalak biztonságos belépési pontot biztosítanak; a személyre

szabható irányítópult (dashboard) átfogó áttekintést nyújt a felhasználó számára releváns információkról; a projekt oldalak részletes nézeteket kínálnak a projektek állapotáról, feladatokról és csapattagokról. A feladatkezelő támogatja a feladatok létrehozását, kategorizálását és delegálását; a csevegés és videokonferencia modulok valós idejű kommunikációt tesznek lehetővé; a naptár és időbeosztás funkciók segítik az események és határidők követését. A munkaidő-nyilvántartás és szabadságkezelés modulok az alkalmazottak munkaidejének adminisztrációját szolgálják; a jelentés-generáló oldal átfogó betekintést nyújt a projektek állapotába és teljesítménybe; az adminisztrációs felületek pedig a rendszer konfigurációját teszik lehetővé. A felső navigációs sáv és az oldalsó menü kontextus-függően biztosítja a navigációt a szerepköröknek megfelelően.

3.8.4. Közös komponensek

A Managly újrafelhasználható komponenskönyvtára egységes felhasználói élémentet biztosít. A navigációs sáv és az oldalsáv a fő navigációs mechanizmusok; a kártyák és táblázatok strukturált adatmegjelenítést tesznek lehetővé; a szabványosított űrlapok és beviteli mezők validációval segítik az adatbevitelt. A modális ablakok másodlagos interakciókra szolgálnak; az értesítési rendszer (toast üzenetek, értesítési panel) tájékoztatja a felhasználót a rendszerállapotról; a betöltő animációk jelzik az aszinkron folyamatokat. Az adatvizualizációs komponensek (diagramok, grafikonok) támogatják a jelentéseket; a fájlfeltöltők, dátumválasztók, keresőmezők és kapcsolódó elem választók pedig specializált funkciókat látnak el. E komponensek Razor parciális nézetek, JavaScript és CSS modulok kombinációjaként valósulnak meg.

3.8.5. HTTP kérések kezelése

A Managly kliens-szerver kommunikációja többféle módszert alkalmaz. A hagyományos űrlap beküldések egyszerűbb műveleteknél szolgálnak; az AJAX kérések (jQuery és vanilla JavaScript) biztosítják az aszinkron adatcserét teljes oldal újratöltése nélkül; a SignalR kapcsolatok valós idejű, kétirányú kommunikációt tesznek lehetővé; a Fetch API pedig modern megközelítést kínál az újabb moduloknál. A kéréskezelés komplex hibakezeléssel rendelkezik, vizuális visszajelzést ad a hosszabb műveleteknél, és kliens oldali gyorsítótárazást alkalmaz a teljesítmény optimalizá-

lására. A batch kérések csökkentik a hálózati kommunikációt; a progressive enhancement biztosítja az alapfunkciók JavaScript nélküli működését; a CSRF védelem biztonságot nyújt; az automatikus újrapróbálkozás pedig növeli a megbízhatóságot. A konzisztens JSON formátumú válaszok egyszerűsítik a fejlesztést és hibakeresést.

3.9. Tesztelés

3.9.1. Automatikus egységtesztek

A Managly rendszer szerveroldali komponensei automatikus, fehér doboz egységtesztek segítségével kerültek validálásra. A tesztelhetőséget nagyban elősegítette a függőség befecskendezés (dependency injection) széles körű alkalmazása, amely lehetővé tette a komponensek izolált tesztelését. A tesztek implementációja során a Moq keretrendszer alkalmaztuk a külső komponensek viselkedésének szimulálására.

Az egységtesztek az NUnit keretrendszer segítségével kerültek implementálásra, amely standard választás .NET környezetben. A tesztosztályok a tesztelés alá vont kontroller alapján strukturálódtak, a tesztmetódusok elnevezése a [tesztelt_metódus]_[tesztelt_eset]_[elvárt_viselkedés] konvenciót követi, felépítésük pedig az arrange/act/assert paradigmát implementálja.

A komplexebb kontrollerek tesztelésénél, ahol számos befecskendezett függőség jelenléte szükséges, az Autofac.Extras.Moq könyvtárat alkalmaztuk. Ez jelentősen redukálta a szükséges kódmezőt, mivel dinamikusan és automatikusan generálja a mock objektumokat és injektálja azokat a tesztelt osztályba, kiküszöbölte a nagyszámú osztályszintű változó deklarálásának szükségességét.

A adatbázis-interakciók mockolására specializált osztályhierarchiát dolgoztunk ki, amelyből származtathatók a konkrét tesztosztályok. Ez a megközelítés centralizálja a közös konfigurációs és inicializációs logikát.

A tesztek elsődleges fókusza a komplex üzleti logikával rendelkező kontrollerek alapos validációja. A külső könyvtárak funkcionálitását természetesen nem teszteltük, csak saját implementációinkat. A ManaglyTest projektben összesen 127 teszteset került implementálásra.

Controller tesztelési stratégiák A kontroller tesztelés során standardizált tesztelési specifikációkat dolgoztunk ki. A kontroller tesztosztály SetUp metódusa min-

den teszeset előtt inicializálja a szükséges mock objektumokat és az adatkörnyezetet, amely a sikeres végrehajtás (happy path) alapesetet reprezentálja. Az első teszeset mindenek validálását célozza, míg a továbbiak a különböző elágazások és hibaesetek lefedésére fókusznak.

A GET metódusok tesztelése során vizsgáltuk, hogy:

- A kért entitás megtalálása esetén a kontroller helyesen visszaadja-e azt
- Nem létező entitás esetén megfelelő NotFound válasz érkezik-e
- Jogosultság hiányában megfelelő Forbidden válasz érkezik-e

A paraméterezett lekérdezések esetében teszteltük, hogy a szűrési és rendezési paraméterek helyesen kerülnek-e alkalmazásra az eredményhalmaz formálásában.

A POST, PUT és DELETE metódusok tesztelése során különös figyelmet fordítottunk az adatintegritást és biztonságot garantáló validációk ellenőrzésére, valamint a tranzakciók helyes végrehajtására.

Kontroller	Tesztek száma
AccountController	11
AdminController	16
ChatController	18
ClockInController	24
DashboardController	9
ProjectsController	23
ReportsController	5
ScheduleController	21
Összesen	127

3.1. táblázat. Tesztelt kontrollerek és a tesztek száma

3.9.2. Alkalmazás manuális tesztelése

Az alkalmazás fekete doboz tesztelése manuális módszerrel történt, amely során kitértem minden lehetséges szélsőséges esetre.

Tesztelés nagy mennyiségi adat esetén Vizsgáltuk az alkalmazás teljesítményét nagy adatmennyiség esetén is. 100+ egyidejű projekttel, feladattal és szabadságkerelemmel tesztelve nem volt érzékelhető teljesítménycsökkenés. Ezt a következő optimalizációs stratégiák segítették:

- Az asszociációk szerveroldalon, az adatbázis-lekérdezés során kerülnek feloldásra
- Lapozás implementálása minden listázó nézetnél (maximum 10 elem oldalanként)
- Külön optimalizált lekérdezési sémák az egy, illetve több entitás visszaadásának eseteire
- Indexelés a gyakran keresett oszlopokon

Többfelhasználós tesztelésnél (pl. chat, videóhívás esetén) a böngésző inkognitó módját alkalmaztuk a második felhasználó számára, elkerülve a local storage megosztásból eredő problémákat, különös tekintettel a JWT token potenciális konfliktusaira.

4. fejezet

Összegzés

A szakdolgozatban egy komplex vállalati menedzsment webalkalmazás tervezése, fejlesztése és validációja került bemutatásra, amely az ASP.NET Core keretrendszer modern technológiáira épül. A Managly rendszer átfogó funkcionálitást biztosít a céges folyamatok teljes spektrumának digitalizálására, a felhasználókezeléstől a projektmenedzsmentig, a kommunikációtól a munkaidő-nyilvántartásig.

A felhasználói dokumentációban a rendszer működését a tipikus munkafolyamatok és felhasználói esetek mentén tekintettük át, minden funkcionálitást szemléletes képernyőképekkel illusztrálva. A fejlesztői dokumentációban részletesen elemeztük az alkalmazás architektúráját, kitérve a vezérlők, modellek és adatréteg strukturális felépítésére és interakciójára. A dokumentáció külön hangsúlyt fektet az egyes komponensek implementációs részleteire, bemutatva a fejlesztés során felmerült ki-hívásokat és az alkalmazott megoldásokat.

A valós idejű kommunikáció megvalósítása során például fontos tervezési döntéseket kellett hozni a SignalR technológia alkalmazásánál, mérlegelve a skálázhatóság és a felhasználói élmény szempontjait. A videóhívás funkcionál kompromisszumot jelentett a WebRTC implementációja, ahol a böngészőkompatibilitás és a médiafolyamkezelés optimalizációja között kellett egyensúlyt találni. A munkaidő-nyilvántartás rendszerében pedig az időzónák kezelése és a szabadságnapok számításának automatizálása jelentett összetett feladatot.

A tesztelési fejezet bemutatta mind az automatizált egységesítések, mind a manuális tesztelés során alkalmazott módszereket. A kontroller-tesztek kiterjedt lefedettséget biztosítottak a kritikus üzleti logika szempontjából, míg a manuális tesztelés az integrált rendszerszintű működést és a felhasználói élményt vizsgálta.

Személyes tapasztalatom a Managly fejlesztése során az volt, hogy a választott moduláris architektúra jelentősen megkönnyítette az új funkciók implementálását és a meglévők módosítását. A függőség befecskendezés elvének következetes alkalmazása nem csak a tesztelhetőséget javította, hanem a kódbázis hosszú távú karbantarthatóságát is. A kontroller-alapú strukturálás lehetővé tette a funkcionálisok logikus szeparációját, miközben a központi adatmodell biztosította a rendszer integritását.

A fejlesztés során különös figyelmet fordítottam a felhasználói élmény optimalizálására, amit a modern frontend technológiák integrációja és a reaktív felhasználói felület kialakítása támogatott. A SignalR és WebRTC technológiák bevezetése jelentősen gazdagította az alkalmazás kommunikációs képességeit, míg a komplex adatmodellek és relációk kezelése során az Entity Framework Core biztosította a hatékony adatelérést és -manipulációt.

A Managly platform sikeres implementációja demonstrálja, hogy a gondosan megtervezett architektúra, a modern technológiák integrációja és a szisztematikus tesztelési megközelítés együttesen biztosíthatja egy komplex vállalati alkalmazás robosztus és fenntartható működését, miközben kielégíti a felhasználói igények széles spektrumát.

Köszönetnyilvánítás

Szeretnék köszöntet mondani témavezetőmnek, Varga Dánielnek, hogy hozzájárult segítségével szakdolgozatom elkészítéséhez. Köszöntettel tartozom a tanulmányaim során megszerzett tudásért a kar oktatónak, akik előadását vagy gyakorlatát hallgattam.

Ábrák jegyzéke

2.1.	A Managly bejelentkező felülete	15
2.2.	A Dashboard felülete a widgetekkel Epmployee felhasználók számára . .	16
2.3.	Az Admin felhasználói felület többletfunkciói	16
2.4.	Az Owner speciális felhasználói felülete	17
2.5.	Példa felugró értesítésre	17
2.6.	Validációs hibaüzenetek	18
2.7.	Példa megerősítésére szolgáló párbeszédablakra	19
2.8.	Fiók törölve oldal	20
2.9.	Felhasználói beosztás megtekintése	22
2.10.	Esemény részletei	23
2.11.	Szabadságkeret követése	23
2.12.	Szabadságigénylés folyamata	24
2.13.	Szabadságkérélemekek állapota	24
2.14.	Beosztás kezelése adminisztrátori nézet	25
2.15.	Ismétlődő műszakok kezelése	27
2.16.	Összevont beosztás nézet	28
2.17.	Felhasználókezelési felület áttekintése	32
2.18.	Felhasználó profil részletei	32
2.19.	Új felhasználó létrehozása	33
2.20.	Kimutatások oldal áttekintése	36
2.21.	Szűrési lehetőségek a kimutatások oldalon	36
2.22.	Statisztikai kártyák a kimutatások oldalon	37
2.23.	Projekt haladás diagram	38
2.24.	Feladat eloszlás diagram	39
2.25.	Csapat teljesítmény táblázat	40
2.26.	Projektek oldal áttekintése a kétpaneles elrendezéssel	41
2.27.	Új projekt létrehozása űrlap	42

2.28. Kiválasztott projekt áttekintése	43
2.29. Projekt fejléc a legfontosabb projekt információkkal	43
2.30. Projekt tevékenységi napló	44
2.31. Csapattagok szekció a projektben résztvevő tagokkal	44
2.32. Feladatok szekció a különböző szűrési lehetőségekkel	45
2.33. Lejárt határidejű feladat vizuális jelzése	46
2.34. Archivált projekt részletei	47
2.35. A profil oldal áttekintése	48
2.36. Csevegés felülete	49
2.37. Csevegés felülete egy kiválasztott felhasználóval	51
2.38. Személyes munkaidő követés felülete	54
2.39. Csapat munkaidő áttekintés felülete	54
2.40. Részletes felhasználói nézet	55
2.41. A Videohívás felület átfogó nézete a három paneles elrendezéssel	56
2.42. Kimenő hívás kezdeményezése és várakozási képernyő	57
2.43. Bejövő hívás értesítés a fogadó felhasználó oldalán	57
2.44. Hívás befejezése utáni összegző képernyő	58
2.45. Újracsatlakozási lehetőség felajánlása	58
2.46. A Tulajdonosi Kezelőfelület áttekintése	59
2.47. Vállalatok kezelése kártyás elrendezéssel és keresőmezővel	60
2.48. Vállalati alapinformációk és felhasználói szerepkör eloszlás	61
2.49. Licenckulcsok összesítő kártyái és keresési lehetőségek	62
2.50. Licenckulcsok létrehozása	62
2.51. Licenckulcsok szerkesztése	63
2.52. Tevékenységnapló áttekintése	63
 3.1. Az alkalmazás rétegei és azok egymásra épülésének szemléltetése	67
3.2. Az alkalmazás egyedkapcsolat diagramja SQL adatípusokkal	73
3.3. A ChatHub UML diagramja	81
3.4. A VideocCallHub UML diagramja	82
3.5. Az AccountController UML diagrammja	83
3.6. Az AdminController UML diagrammja	83
3.7. A ChatController UML diagrammja	84
3.8. A ClockInController UML diagrammja	84

3.9. A DashboardController UML diagrammja	85
3.10. A HomeController UML diagrammja	85
3.11. A NotificationController UML diagrammja	85
3.12. Az OwnerController UML diagrammja	85
3.13. A ProjectsController UML diagrammja	86
3.14. A ReportsController UML diagrammja	86
3.15. A ScheduleController UML diagrammja	87
3.16. A VideoConferenceController UML diagrammja	87