

**Záródolgozat**

Készítették:  
Hilóczki Tamás – Monostori Róbert – Zelenák Zalán Noel  
Konzulens:  
Kasza László Róbert

Miskolc  
2025

**Miskolci SZC Kandó Kálmán Informatikai Technikum**

**Miskolci Szakképzési Centrum**

**SZOFTVERFEJLESZTŐ- ÉS TESZTELŐ SZAK**

**HMZ RT.**

Komplett szálloda rendszer

Hilóczki Tamás – Monostori Róbert – Zelenák Zalán Noel

2024-2025

Tartalom

[1. A Projekt Környezete és Céljai 4](#_Toc195128400)

[1.1. Fejlesztőkörnyezet és Eszközök 4](#_Toc195128401)

[1.2. Technológiai Választások Indoklása és Projektcél 5](#_Toc195128402)

[1.3. Projektcsapat és Szerepkörök 7](#_Toc195128403)

[2. A Projekt Bemutatása 7](#_Toc195128404)

[2.1. Frontend Webalkalmazás 8](#_Toc195128405)

[2.1.1. Főoldal és Navigáció 8](#_Toc195128406)

[2.1.2. Szobák Böngészése és Részletek 9](#_Toc195128407)

[2.1.3. Foglalási Folyamat 10](#_Toc195128408)

[2.1.4. Felhasználói Fiók (Bejelentkezés, Regisztráció, Profil) 10](#_Toc195128409)

[2.1.5. További Oldalak 11](#_Toc195128410)

[2.2. Backend API Bemutatása (Swagger) 11](#_Toc195128411)

[2.2.1. Főbb API Végpontok (Controller-ek) és Feladataik 12](#_Toc195128412)

[2.3. Oldallogika és Adatfolyam 13](#_Toc195128413)

[2.4 WPF Admin Kliens bemutatása 13](#_Toc195128414)

[2.4.1 Bejelentkezés és Hitelesítés 13](#_Toc195128415)

[2.4.2 Fő ablak és Navigáció (MainWindow) 14](#_Toc195128416)

[2.4.3 Főbb Adminisztrációs Funkciók 14](#_Toc195128417)

[2.4.3.1 Szobakezelés: 15](#_Toc195128418)

[2.4.3.2 Dolgozók Kezelése: 16](#_Toc195128419)

[2.4.3.3 Események és Eseményfoglalások Kezelése: 16](#_Toc195128420)

[2.4.3.4 Email Küldő Modul: 17](#_Toc195128421)

[2.5. Projektmenedzsment (Trello) 18](#_Toc195128422)

[2.6. Verziókezelés (GitHub) 18](#_Toc195128423)

[2.7. Adatbázis Terv 19](#_Toc195128424)

[2.7.1. Főbb Táblák és Kapcsolataik 19](#_Toc195128425)

[3. Telepítés és Konfiguráció 20](#_Toc195128426)

[3.1. Rendszerkövetelmények 20](#_Toc195128427)

[3.2. Adatbázis Beállítása 20](#_Toc195128428)

[3.3. Backend Konfiguráció és Futtatás 20](#_Toc195128429)

[3.4. Frontend Build és Kiszolgálás 20](#_Toc195128430)

[3.5. WPF Kliens Telepítése 20](#_Toc195128431)

[3.6. Cloudflare Tunnel Beállítása 21](#_Toc195128432)

[3.7 Összegzés, fejlesztési lehetőségek: 21](#_Toc195128433)

[Mellékletek 23](#_Toc195128434)

# 1. A Projekt Környezete és Céljai

Ez a fejezet bemutatja a HMZ szállodamenedzsment rendszer fejlesztésének hátterét, a használt technológiákat és eszközöket, valamint a projektcsapat felépítését és a technológiai döntések indoklását.

## 1.1. Fejlesztőkörnyezet és Eszközök

A HMZ rendszer fejlesztése során modern és iparágban elterjedt eszközöket és technológiákat alkalmaztunk a hatékonyság, megbízhatóság és karbantarthatóság érdekében.

**Backend Fejlesztés:**

Programozási nyelv: C#

Keretrendszer: ASP.NET Core 8 [1]

Fejlesztőkörnyezet (IDE): Microsoft Visual Studio 2022

Adatbázis-kezelés: Entity Framework Core 8 [2]

API Dokumentáció: Swashbuckle (Swagger UI) [3]

**Frontend Fejlesztés:**

Programozási nyelv: JavaScript (ES6+)

Könyvtár/Keretrendszer: React (v18.x) [4]

Stílusozás: Tailwind CSS [5], CSS3

Fejlesztőkörnyezet: Visual Studio Code

Csomagkezelő: npm

**Adminisztrációs Kliens Fejlesztés:**

Programozási nyelv: C#

Keretrendszer: .NET 8 WPF (Windows Presentation Foundation)

Felhasználói felület: Material Design in XAML Toolkit [6]

Fejlesztőkörnyezet (IDE): Microsoft Visual Studio 2022

**Adatbázis:**

Adatbázis-kezelő rendszer: XAMPP MySQL

**Adminisztrációs eszköz: MySQL Workbench**

**Verziókezelés:**

**Git**

Kódtárhely: GitHub (Lásd 2.5. szakasz)

**Projektmenedzsment:**

Trello (Lásd 2.4. szakasz)

**Egyéb Eszközök:**

API tesztelés: Postman, Swagger UI

Hálózati elérhetőség: Cloudflare Tunnel [11]

## 1.2. Technológiai Választások Indoklása és Projektcél

A projekt technológiai stackjének összeállításakor több szempontot vettünk figyelembe:

ASP.NET Core (Backend): Robusztus, nagy teljesítményű és platformfüggetlen keretrendszer, amely kiválóan alkalmas komplex üzleti logikával rendelkező, skálázható RESTful API-k fejlesztésére. A Microsoft által támogatott, széleskörű közösséggel és dokumentációval rendelkezik. A beépített dependency injection, konfigurációs rendszer és Entity Framework Core integráció jelentősen gyorsítja a fejlesztést.

React (Frontend): Az egyik legnépszerűbb JavaScript könyvtár modern, dinamikus és felhasználóbarát webes felületek készítéséhez. Komponens alapú architektúrája elősegíti az újrafelhasználhatóságot és a karbantarthatóságot. A nagy ökoszisztéma (state management, routing, UI könyvtárak) rugalmasságot biztosít. A Tailwind CSS használata pedig hatékony és konzisztens stílusozást tesz lehetővé.

WPF (.NET 8) (Admin Kliens): Olyan adminisztrációs felületre volt szükség, amely kihasználja a natív Windows környezet előnyeit, gazdag felhasználói élményt nyújt, és könnyen integrálható a meglévő .NET alapú backenddel. A WPF lehetővé teszi komplexebb UI elemek és munkafolyamatok implementálását, a Material Design Toolkit pedig modern megjelenést biztosít. A .NET 8 választása a legújabb funkciók és teljesítményjavítások kiaknázását célozza.

MySQL (Adatbázis): Népszerű, nyílt forráskódú, megbízható és jól skálázható relációs adatbázis-kezelő rendszer. Költséghatékony megoldás, amely széles körben támogatott és jól dokumentált. Az Entity Framework Core kiválóan támogatja a MySQL-t, zökkenőmentes adatbázis-interakciót biztosítva a backend számára.

Git/GitHub: Ipari standard verziókezelő rendszer, amely elengedhetetlen a csapatmunka során a kódváltozások követéséhez, ütközések kezeléséhez és a fejlesztési folyamat átláthatóságához. A GitHub platformként biztosítja a kódtárolást, a kódellenőrzést (code review) és a CI/CD folyamatok integrálását (opcionális).

Trello: Egyszerű, vizuális projektmenedzsment eszköz, amely Kanban tábláival segíti a feladatok nyomon követését, a sprintek tervezését és a csapaton belüli kommunikációt a feladatok állapotáról.

A Projekt Elsődleges Célja: A projekt fő célkitűzése a fejlesztőcsapat tagjainak szakmai tudásának és felkészültségének bemutatása volt egy komplex, valós problémára épülő szoftverrendszer megtervezésén és implementálásán keresztül, a vizsgakövetelményeknek megfelelően.

## 1.3. Projektcsapat és Szerepkörök

A projekt sikeres megvalósításában a következő csapattagok vettek részt, meghatározott felelősségi körökkel:

Hilóczki Tamás - Backend Fejlesztő / Tesztelő:

Felelősségek: Az ASP.NET Core API tervezése és fejlesztése, adatbázis séma kialakítása és implementálása (EF Core), üzleti logika implementálása, API végpontok létrehozása, hitelesítés és biztonság megvalósítása, API dokumentáció (Swagger), rendszer tesztelése.

Zelenák Zalán Noel - Frontend Fejlesztő:

Felelősségek: A React alapú vendégoldali webalkalmazás tervezése és fejlesztése, komponensek létrehozása, API integráció, felhasználói felület és élmény (UI/UX) kialakítása, reszponzivitás biztosítása, stílusozás (Tailwind CSS).

Monostori Róbert - WPF Admin Kliens Fejlesztő / Tesztelő:

Felelősségek: Az adminisztrációs asztali alkalmazás tervezése és fejlesztése WPF segítségével, UI tervezés (Material Design), API integráció, adminisztratív funkciók implementálása (szobakezelés, foglaláskezelés, stb.), rendszer tesztelése.

A csapat tagjai szorosan együttműködtek, a projektmenedzsment feladatokat közösen látták el. A tesztelési feladatok Hilóczki Tamás és Monostori Róbert között oszlottak meg, amelyet kiterjedt felhasználói tesztelés egészített ki a minőség biztosítása érdekében. Dedikált projektmenedzseri, QA vagy DevOps szerepkör nem volt, ezeket a feladatokat a csapattagok megosztva végezték. A verziókezelést a GitHub segítségével végezték.

# 2. A Projekt Bemutatása

Ez a fejezet részletesen bemutatja a HMZ rendszer fő komponenseit: a vendégoldali webes felületet (Frontend), a háttérrendszert (Backend API), valamint a projektmenedzsment és adatbázis-tervezés aspektusait. A bemutatás célja a rendszer működésének és használatának gyakorlatias megismertetése, képernyőképekkel és vizuális elemekkel illusztrálva.

## 2.1. Frontend Webalkalmazás

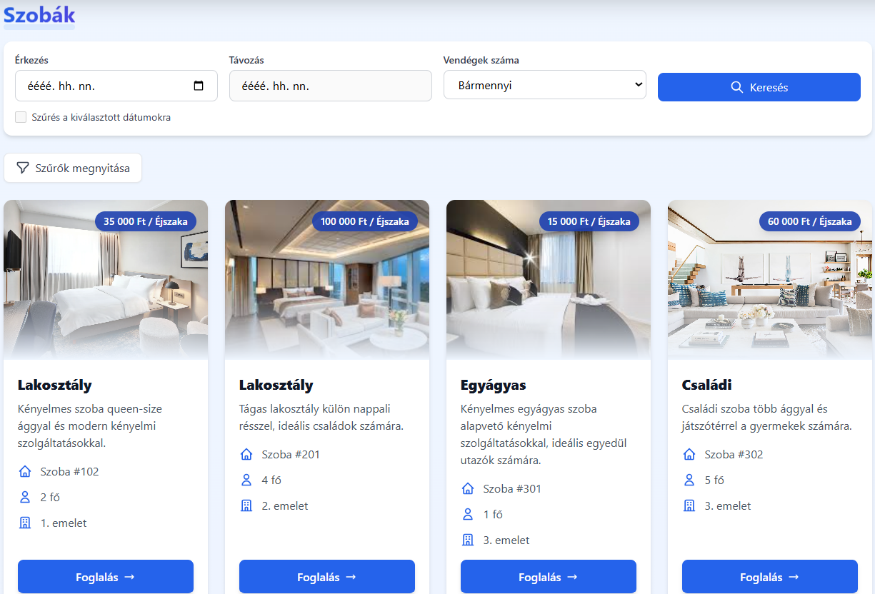
A Frontend alkalmazás a vendégek elsődleges kapcsolódási pontja a szállodával. Célja, hogy intuitív, gyors és esztétikus felületet biztosítson a szobakereséshez, foglaláshoz és a szállodával kapcsolatos információk eléréséhez. React technológiára épül, amely dinamikus és reszponzív felhasználói élményt tesz lehetővé. A webalkalmazás a hmzrt.eu címen érhető el.

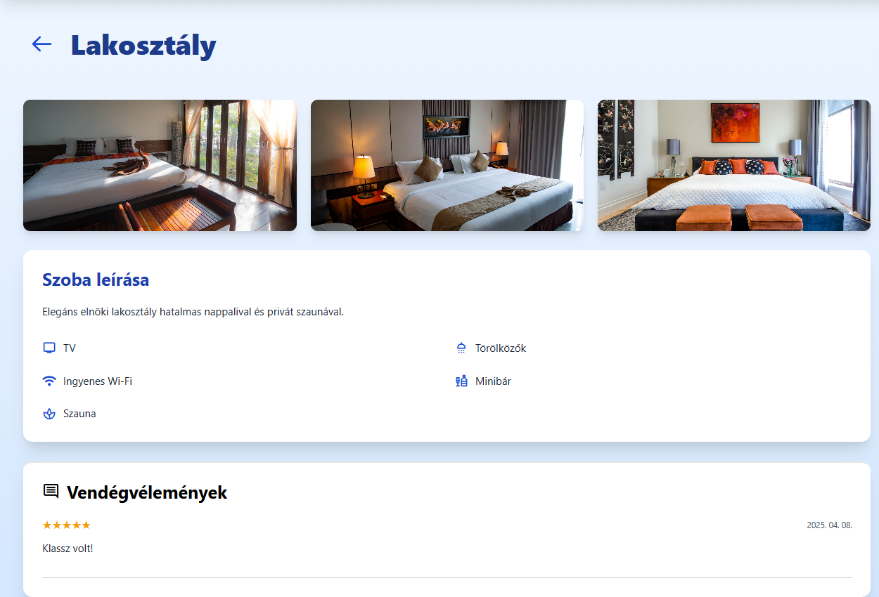
### 2.1.1. Főoldal és Navigáció

A Főoldal (Hero.js, Navbar.js) az első benyomás a vendég számára. Tartalmaz egy figyelemfelkeltő részt, rövid bemutatkozást, és a navigációs sávot (Navbar.js), amely biztosítja a könnyű hozzáférést az oldal főbb részeihez.



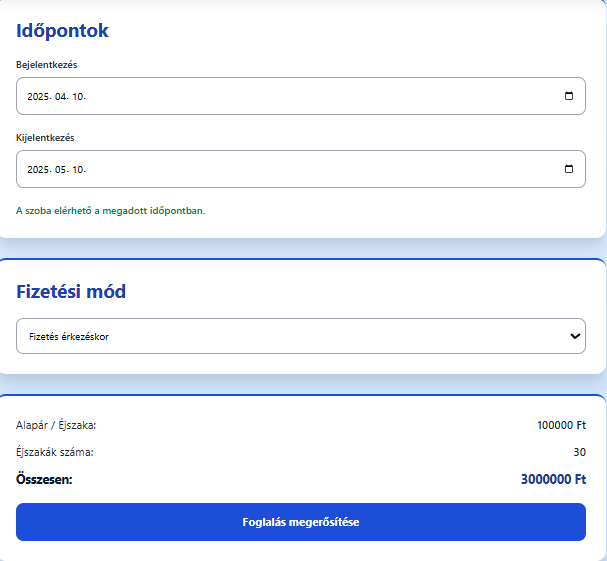
### 2.1.2. Szobák Böngészése és Részletek

A "Szobák" menüpont (Rooms.js) alatt a vendégek megtekinthetik a rendelkezésre álló szobatípusokat, szűrhetnek és áttekinthetik a listát. Egy adott szobára kattintva a részletes oldal (DetailItem.js) jelenik meg további információkkal és foglalási lehetőséggel.



### 2.1.3. Foglalási Folyamat

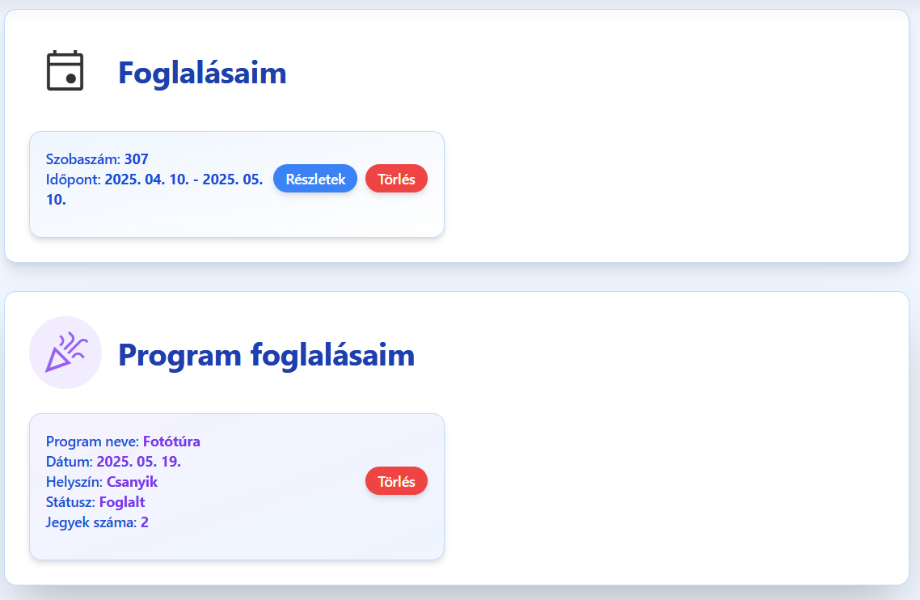
A foglalás (Foglalas.js) több lépésből áll: dátumválasztás és elérhetőség ellenőrzése, vendégadatok megadása, esetleges extrák hozzáadása, összegzés és fizetési mód választása, végül a visszaigazolás megjelenítése és e-mail küldése.





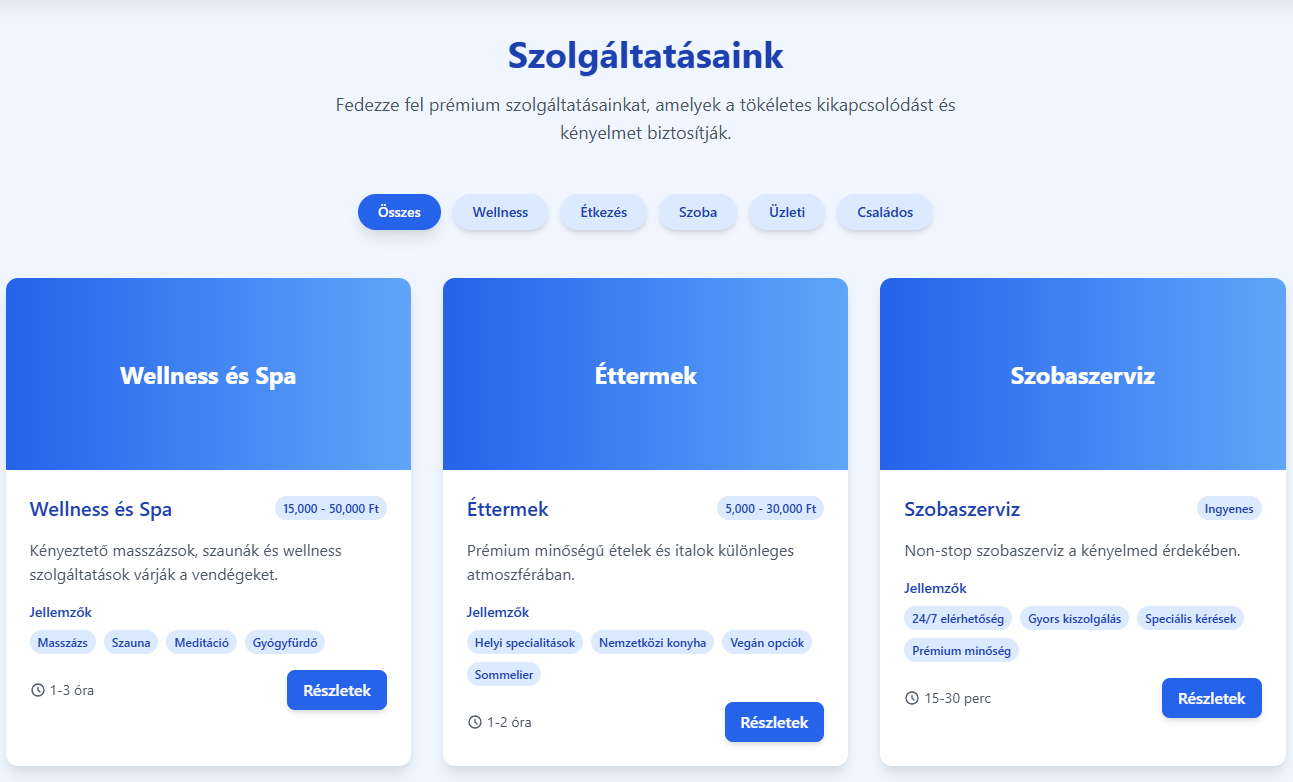
2.1.4. Felhasználói Fiók (Bejelentkezés, Regisztráció, Profil)

A vendégek regisztrálhatnak (RegisterModal.js) és bejelentkezhetnek (LoginModal.js), akár kétfaktoros hitelesítéssel (2FA - VertificationModal.js). Bejelentkezés után a profil oldalon (ProfileModal.js) kezelhetik adataikat és foglalásaikat.



### 2.1.5. További Oldalak

A weboldal további informatív oldalakat is tartalmaz (Programok, Szolgáltatások, Rólunk).



## 2.2. Backend API Bemutatása (Swagger)

A Backend API (ASP.NET Core) a rendszer központi logikai és adatkezelő egysége. RESTful végpontokat biztosít JSON formátumban a Frontend és a WPF Kliens számára. Az API dokumentációja és tesztelési felülete a Swagger UI segítségével érhető el az api.hmzrt.eu/swagger címen.

api.hmzrt.eu/swagger végpontjai a \*Mellékletek\* részben találhatóak.

### 2.2.1. Főbb API Végpontok (Controller-ek) és Feladataik

A Swagger felületen látható főbb végpont csoportok (Controller-ek) és azok felelősségi körei:

UserAccounts: Felhasználói fiókok kezelése (regisztráció, login, 2FA, profil).

Rooms: Szobainformációk kezelése (lekérdezés, módosítás).

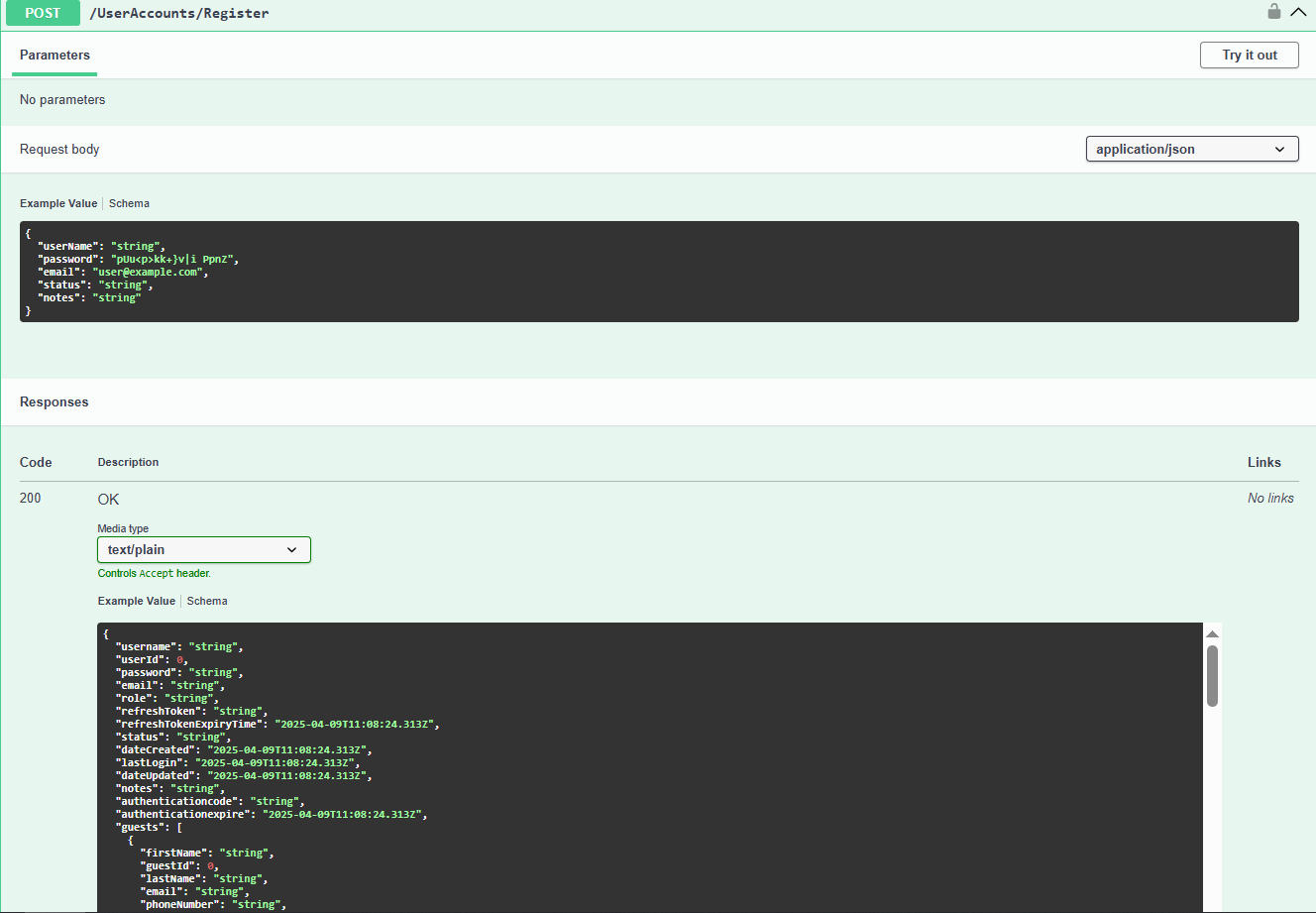
Bookings: Foglalások kezelése (létrehozás, lekérdezés, módosítás).

Guests: Vendégadatok kezelése.

Events / EventBookings: Események és kapcsolódó foglalások kezelése.

(További Controller-ek rövid említése a korábbi lista alapján: RoomMaintenance, Payments, Invoices, Services, Promotions, Staff stb.)

A végpontok JWT token alapú hitelesítést használnak a védett műveletekhez.



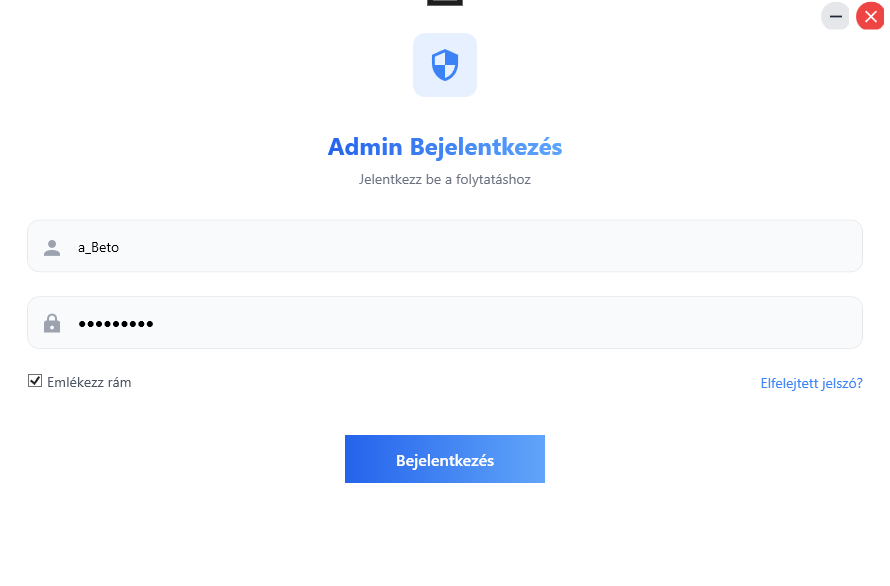
## 2.3. Oldallogika és Adatfolyam

A rendszer működése során a kliens (Frontend vagy WPF) HTTP kérést küld a Backend API-nak (api.hmzrt.eu). Az API feldolgozza a kérést (hitelesítés, üzleti logika, adatbázis műveletek), majd JSON választ küld vissza a kliensnek, amely frissíti a felhasználói felületet. Ez a REST alapú kommunikáció biztosítja a komponensek közötti kapcsolatot.

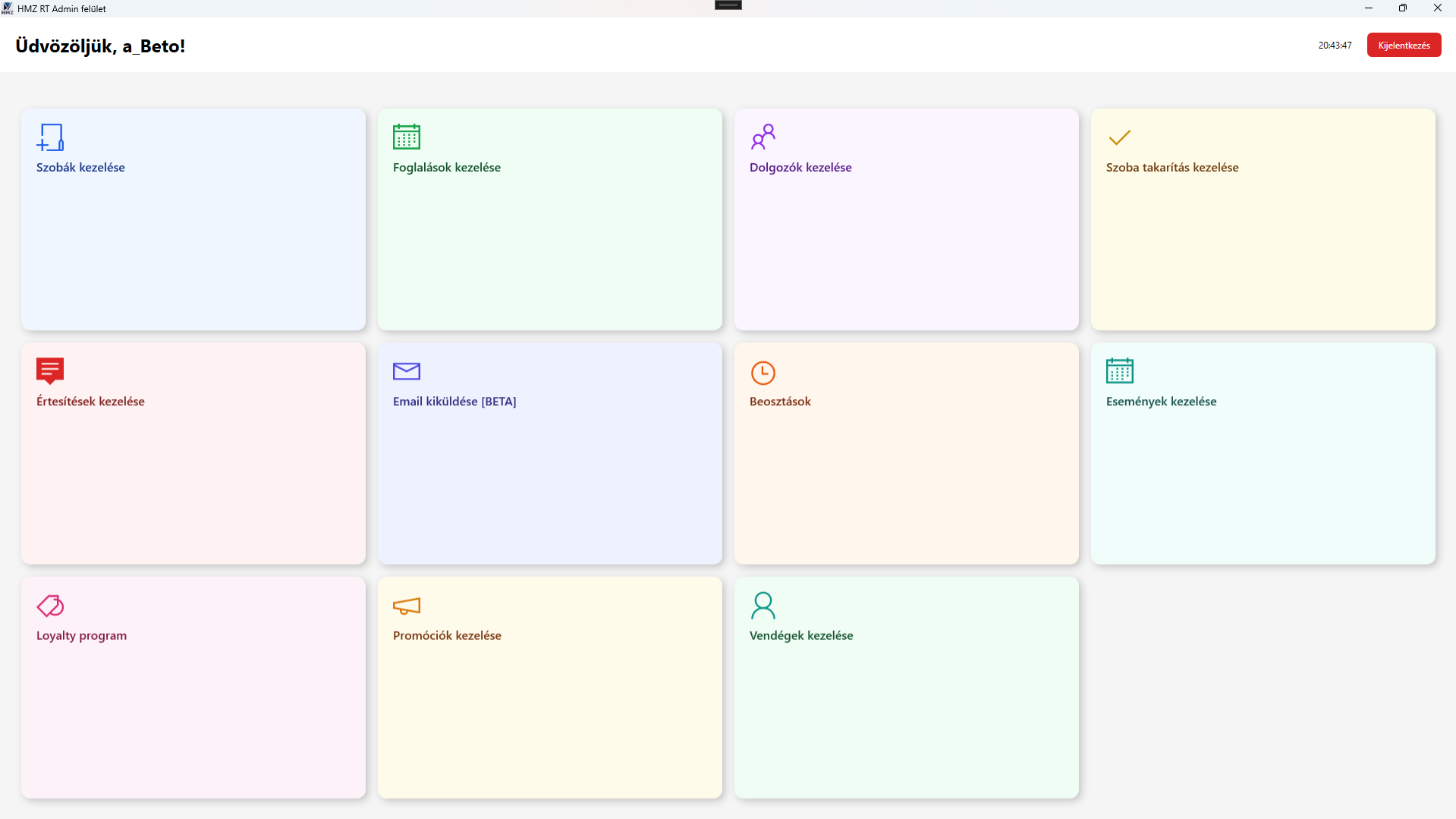
## 2.4 WPF Admin Kliens bemutatása

A HMZ rendszer adminisztratív feladatait egy különálló, Windows asztali alkalmazás látja el, amely Windows Presentation Foundation (WPF) technológiával készült, a projektben meghatározott .NET keretrendszer (lásd 1.1 szakasz) alatt. Ez a kliens kizárólag az adminisztrátorok( pl: Recepciós) és a 'System' jogosultságú felhasználók számára készült, hogy kezelni tudják a szálloda működésének kulcsfontosságú elemeit.

### 2.4.1 Bejelentkezés és Hitelesítés

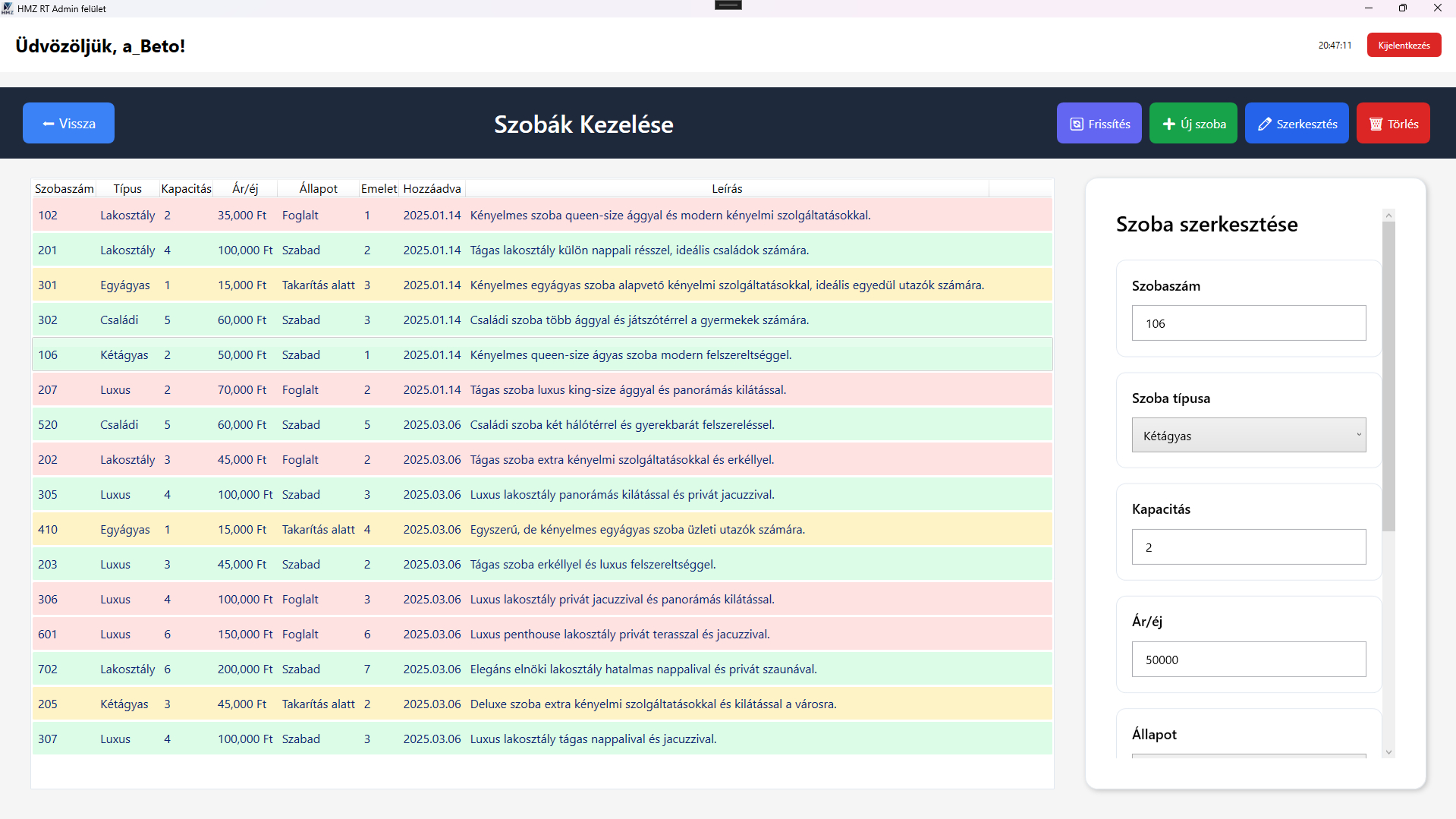
Az alkalmazás indításakor a `LoginWindow` fogadja a felhasználót. A bejelentkezéshez érvényes felhasználónév és jelszó szükséges. A backend API ellenőrzi a megadott adatokat és a felhasználó szerepkörét; kizárólag 'Admin' vagy 'System' jogosultsággal lehet belépni. Sikeres hitelesítés esetén a rendszer eltárolja a kapott JWT tokeneket (`TokenStorage`), amelyeket a további API kommunikációhoz használ. A 'Jegyezz meg' opcióval a bejelentkezési adatok (titkosítva) elmenthetők a könnyebb későbbi belépés érdekében. Az 'Elfelejtett jelszó' funkció segítségével a felhasználók emailben kapott kód alapján új jelszót állíthatnak be maguknak.

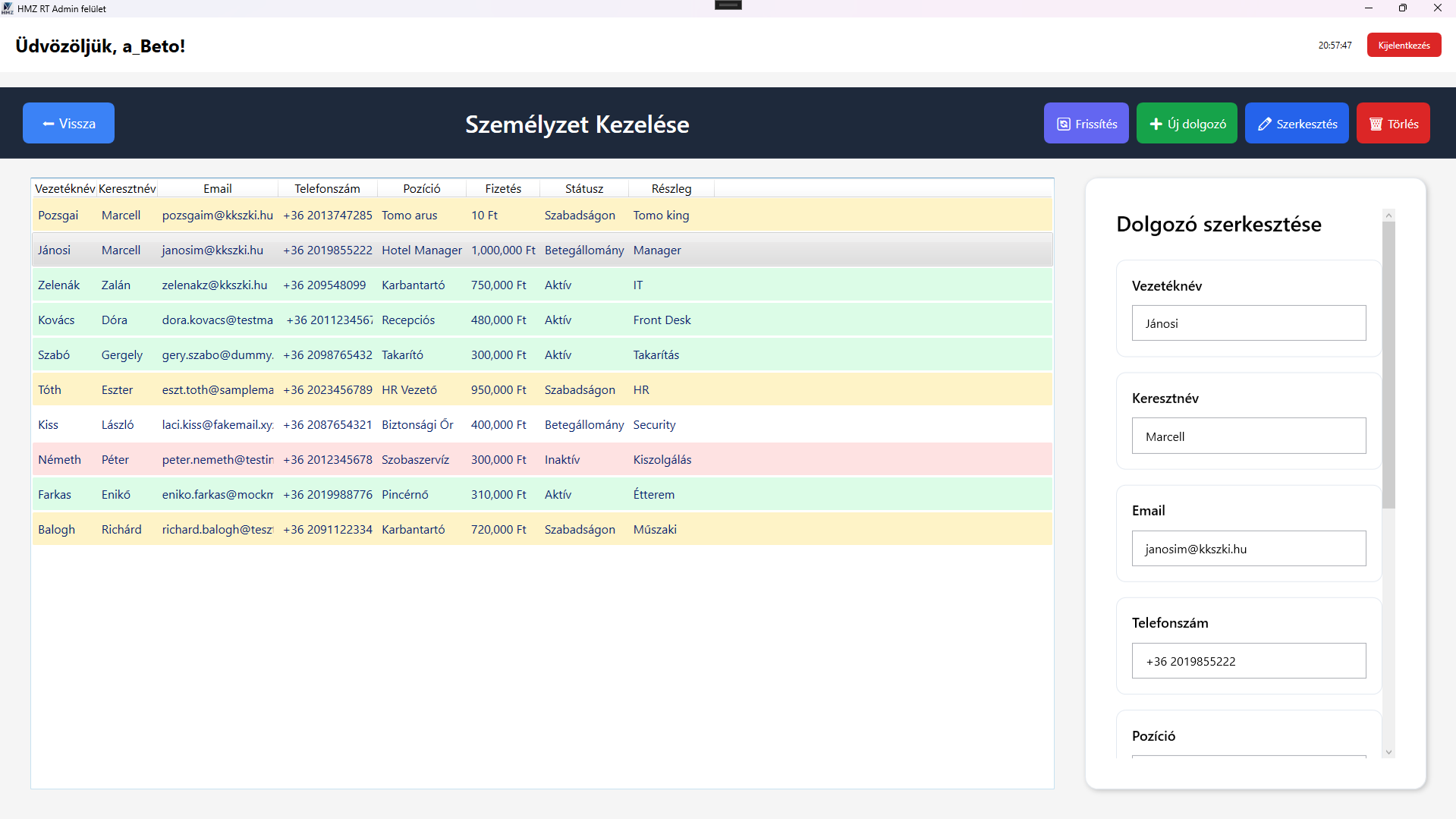
### 2.4.2 Fő ablak és Navigáció (MainWindow)

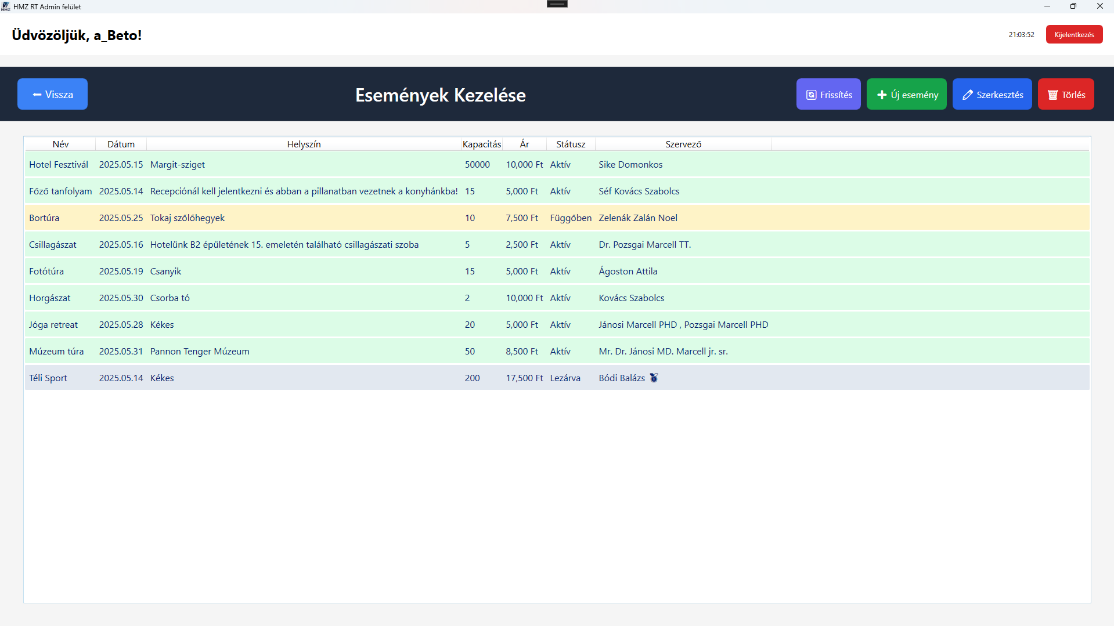
Sikeres bejelentkezés után a `MainWindow` jelenik meg, amely az alkalmazás központi vezérlőpultja. A fő területen elhelyezett kártyák segítségével lehet navigálni a különböző menedzsment modulok (Szobák, Dolgozók, Foglalások stb.) között. A felület megjeleníti a bejelentkezett felhasználó nevét az aktuális időt, valamint lehetőséget biztosít a kijelentkezésre. 

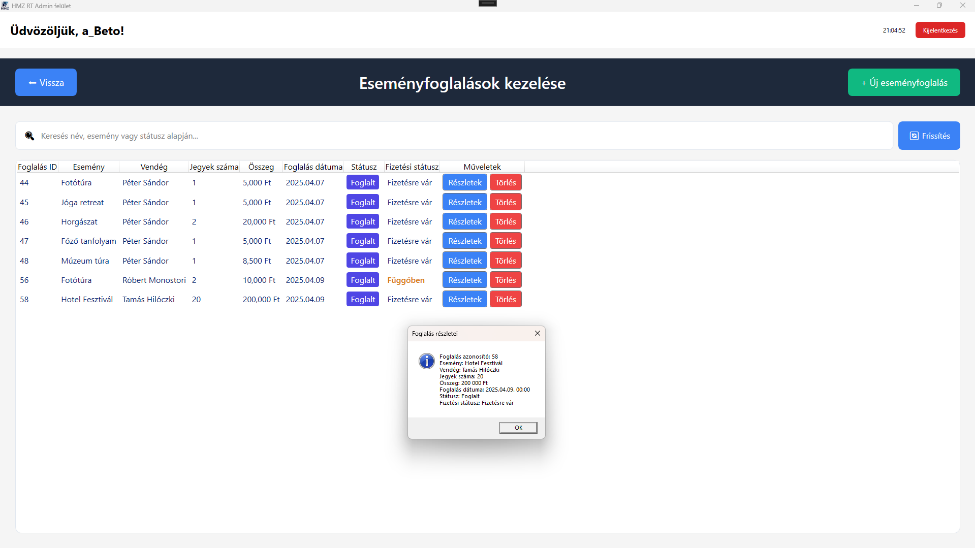
### 2.4.3 Főbb Adminisztrációs Funkciók

Az adminisztrációs kliens számos modult tartalmaz a szálloda napi műkésének kezelésére, melyeket az alábbi alpontokban részletezünk:

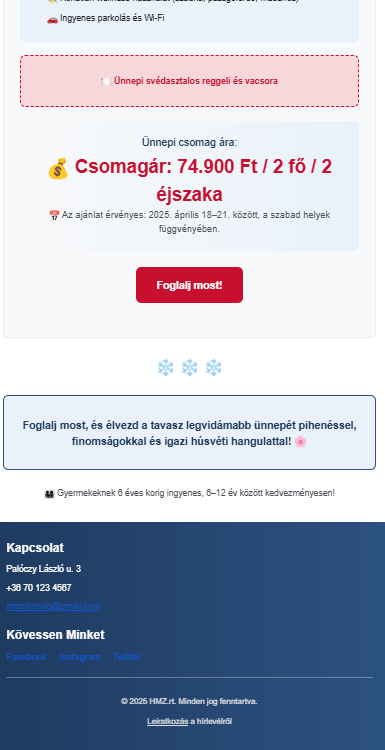
2.4.3.1 Szobakezelés:  
Lehetőség van a szobák listázására, új szoba hozzáadására, meglévő szoba adatainak (szám, típus, kapacitás, ár/éj, státusz, emelet, leírás) szerkesztésére és törlésére. A felület validálja a bemeneti adatokat (pl. kapacitás maximuma, ár maximuma).

2.4.3.2 Dolgozók Kezelése:   
Adminisztrálhatók a szálloda dolgozói: új dolgozó felvétele, meglévők adatainak (név, email, telefon, pozíció, fizetés, részleg, státusz) módosítása és dolgozó törlése. A fizetés mező csak számokat fogad el és maximális értéke korlátozott.

2.4.3.3 Események és Eseményfoglalások Kezelése:  
Szállodai események (pl. konferencia) létrehozása, adatainak szerkesztése, listázása, törlése. Az eseményekre érkezett foglalások kezelése: résztvevők listázása, új foglalás rögzítése (vendég, esemény, jegyek száma), törlése. A jegyek száma alapján a rendszer kiszámolja a fizetendő összeget.



2.4.3.4 Email Küldő Modul:  
Lehetőség van HTML sablonok alapján emailek küldésére. A rendszer betölti a 'Templates' mappában található `.html` fájlokat, megjeleníti a bennük lévő változókat (pl. `{GuestName}`), amelyeket az adminisztrátor kitölthet. Az elkészült email a hírlevélre feliratkozott email címekre.



## 2.5. Projektmenedzsment (Trello)

A fejlesztési feladatokat Trello [9] segítségével, Kanban táblán követtük nyomon, oszlopok (pl. Backlog, Folyamatban, Kész) és feladatkártyák segítségével.



## 2.6. Verziókezelés (GitHub)

A forráskódot Git segítségével kezeltük és a GitHub [10] platformon tároltuk. Feature branch modellt és code review folyamatot alkalmaztunk.

A projekt GitHub repository-ja:

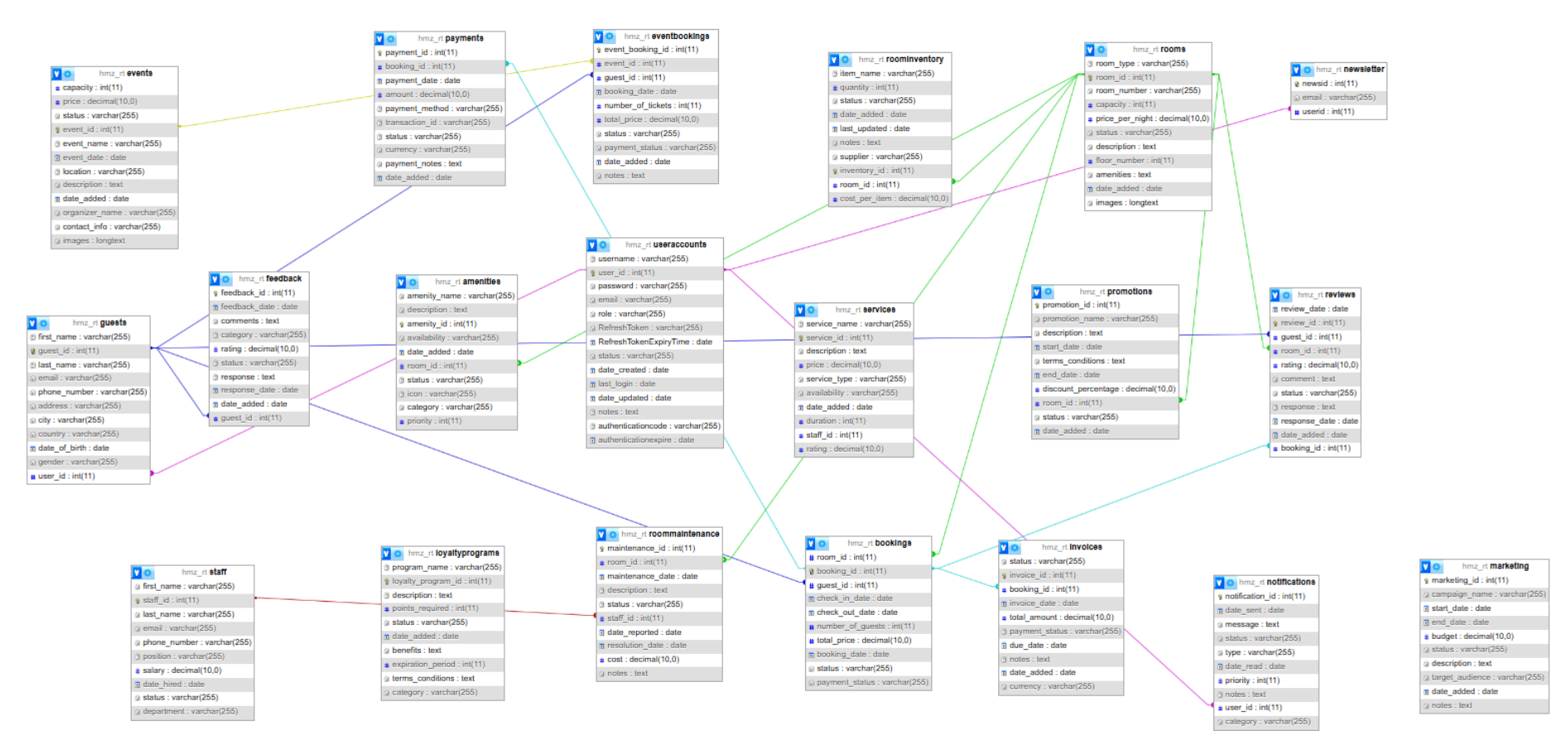
https://github.com/MrTahil/hotel (Hozzáférés szükséges lehet)

## 2.7. Adatbázis Terv

MySQL relációs adatbázist (hmz\_rt séma) használtunk, melynek struktúráját EF Core Code First megközelítéssel hoztuk létre.

### 2.7.1. Főbb Táblák és Kapcsolataik

Az adatbázis séma a szálloda működéséhez szükséges entitásokat modellezi.



Főbb táblák: Guests, Rooms, Bookings, UserAccounts, Staff, Events, EventBookings, Payments, Services, RoomMaintenance, stb. A kapcsolatok biztosítják az adatintegritást.

# 3. Telepítés és Konfiguráció

Ez a fejezet útmutatást nyújt a HMZ rendszer komponenseinek alapvető telepítéséhez és konfigurálásához a projekt jelenlegi futtatási környezetében.

## 3.1. Rendszerkövetelmények

Szerver (Otthoni Gép): Windows OS, .NET 8 Runtime, MySQL Server, cloudflared kliens [11].

Frontend (Felhasználó oldali): Modern webböngésző.

WPF Kliens (Adminisztrátor oldali): Windows OS, .NET 8 Desktop Runtime.

## 3.2. Adatbázis Beállítása

MySQL szerver telepítése és konfigurálása a szerver gépen.

hmz\_rt adatbázis létrehozása.

hmz\_rt.sql szkript futtatása az adatbázison.

## 3.3. Backend Konfiguráció és Futtatás

appsettings.json konfigurálása (ConnectionStrings, SmtpSettings, Jwt kulcs, CORS).

Backend alkalmazás indítása a szerver gépen (pl. dotnet run).

## 3.4. Frontend Build és Kiszolgálás

Frontend projekt buildelése (npm install, npm run build).

A build mappa tartalmát a Cloudflare Tunnel szolgálja ki (lásd 3.6).

## 3.5. WPF Kliens Telepítése

WPF projekt publikálása.

API cím módosítása a kódban (TokenService.cs-ben baseAddress = "https://api.hmzrt.eu/"), majd újrafordítás/publikálás.

Publikált fájlok másolása a kliens gépre, .NET 8 Desktop Runtime telepítése.

## 3.6. Cloudflare Tunnel Beállítása

A rendszer nyilvános elérését (hmzrt.eu, api.hmzrt.eu) Cloudflare Tunnel [11] biztosítja, amely az otthoni gépen futó szolgáltatásokat (Backend API, Frontend kiszolgáló) köti össze a Cloudflare hálózatával. A cloudflared klienst és a hozzá tartozó konfigurációs fájlt a szerver gépen kell beállítani a megfelelő átirányításokkal (hostname-ek a helyi portokra).

## 3.7 Összegzés, fejlesztési lehetőségek:

(DB) marketing tábla

(DB) invoices tábla

(DB) Képek tárolása

(DB) loyaltprograms tábla

(DB) services tábla

[Backend] RoomInventory\_controller

[Backend] Invoices\_Controller

[Backend] Loyaltyprograms\_Controller

[Backend] Services\_controller

[Backend] Marketing\_Controller

[WPF] Értesítések kezelése.

[WPF] Beosztások kezelése.

[WPF] Loyalty program.

#### Irodalomjegyzék

www.learn.microsoft.com/aspnet/core, Cím: ASP.NET Core documentation, Letöltés dátuma: 2025.02.10.

www.learn.microsoft.com/ef/core, Cím: Entity Framework Core documentation, Letöltés dátuma: 2025.02.10.

www.learn.microsoft.com/dotnet/desktop/wpf/, Cím: Windows Presentation Foundation (.NET), Letöltés dátuma: 2025.02.10.

www.github.com/domaindrivendev/Swashbuckle.AspNetCore, Cím: Swashbuckle.AspNetCore GitHub Repository, Letöltés dátuma: 2025.02.15.

www.react.dev, Cím: React Official Website, Letöltés dátuma: 2025.02.11.

www.tailwindcss.com, Cím: Tailwind CSS Official Website, Letöltés dátuma: 2025.02.11.

www.materialdesigninxaml.net, Cím: Material Design in XAML Toolkit Official Website, Letöltés dátuma: 2025.02.18.

www.mysql.com, Cím: MySQL Official Website, Letöltés dátuma: 2025.01.20.

www.github.com/scalar/scalar, Cím: Scalar API References GitHub Repository, Letöltés dátuma: 2025.03.05.

www.trello.com, Cím: Trello Official Website, Letöltés dátuma: 2025.01.15.

www.github.com, Cím: GitHub Official Website, Letöltés dátuma: 2025.01.15.

developers.cloudflare.com/cloudflare-one/connections/connect-apps/, Cím: Cloudflare Tunnel Documentation, Letöltés dátuma: 2025.04.09.

# Mellékletek

