Medicine Hospital Information System

Baracsi Kinga

Böröcz Evelin Tilda

Nagygéci Noémi

2022.

Szeged Vasvári Pál Technikum

Tartalom

[Bevezetés 4](#_Toc101305769)

[Asztali alkalmazás 5](#_Toc101305770)

[Tervezési folyamat 5](#_Toc101305771)

[Bejelentkezés 5](#_Toc101305772)

[Alkalmazás kinézete 5](#_Toc101305773)

[Alkalmazott programok, nyelvek 6](#_Toc101305774)

[MVVM 6](#_Toc101305775)

[Fejlesztési folyamat 6](#_Toc101305776)

[Mapparendszer 10](#_Toc101305777)

[Gombok 11](#_Toc101305778)

[Fontosabb metódusok 14](#_Toc101305779)

[Mobil alkalmazás 16](#_Toc101305780)

[Tervezési folyamat 16](#_Toc101305781)

[Kezdőlap 16](#_Toc101305782)

[Bejelentkezés 16](#_Toc101305783)

[Menü 17](#_Toc101305784)

[Orvosok keresése 17](#_Toc101305785)

[Alkalmazott programok, nyelvek 18](#_Toc101305786)

[Fejlesztési folyamat 18](#_Toc101305787)

[Webes alkalmazás 25](#_Toc101305788)

[Tervezési folyamat 25](#_Toc101305789)

[Kezdőoldal 25](#_Toc101305790)

[Menü 25](#_Toc101305791)

[Személyes adatok megtekintése 26](#_Toc101305792)

[Gyógyszerek megtekintése 26](#_Toc101305793)

[Gyógyszerek hozzáadása 26](#_Toc101305794)

[Orvosok keresése 27](#_Toc101305795)

[Alkalmazott programok, nyelvek 27](#_Toc101305796)

[Fejlesztési folyamat 27](#_Toc101305797)

[Web API 33](#_Toc101305798)

[Tervezési folyamat 33](#_Toc101305799)

[Alkalmazott programok, nyelvek 33](#_Toc101305800)

[Fejlesztési folyamat 33](#_Toc101305801)

[Adatbázis 37](#_Toc101305802)

[Egyedek 37](#_Toc101305803)

[Tulajdonságok 37](#_Toc101305804)

[Egyedek és tulajdonságaik (adattípusok) 37](#_Toc101305805)

[GitHub 40](#_Toc101305806)

[Csapatmunka 40](#_Toc101305807)

Bevezetés:

MHIS (=Medicine Hospital Information System) egy kórházi információs rendszer, amely lehetőséget nyújt a szakorvosoknak, illetve a háziorvosoknak betegek adatainak nyilvántartására a betegek leleteinek, diagnózisainak, gyógyszereinek felírására, beállítására, valamint a kezelési, ápolási javallatainak megtekintéseire, írásaira. Ezen kívül a páciens számára lehetővé teszi saját leleteinek megtekintését webes felületen.

Jelenleg a MedSol szoftverrendszert használják leginkább, ami egy kórházi információs rendszer, amely lehetőséget nyújt a szakorvosoknak, illetve a háziorvosoknak az orvosi asszisztensek segítségével betegek adatainak nyilvántartására a betegek leleteinek, diagnózisainak, anamnéziseinek, gyógyszereinek felírására, beállítására, valamint a kezelési, ápolási javallatainak megtekintéseire, írásaira. De a betegek a leleteket csak az asszisztenstől (háziorvos esetén) tudják kikérni a leleteket. Ezeket a leleteket könnyen elveszíthetik vagy megsérülhetnek, elázik vagy elszakad, így a

A mi rendszerünk legfontosabb jellemzője, az, hogy felhasználó-barát: könnyen átlátható egyaránt a szakkifejezéseket ismerő felhasználóknak számára és a laikus felhasználóknak, azaz a páciensek részére is. Ezen kívül modern technika az egységességért az egészségügyben: a háziorvosoknak a szakorvosoknak, ápolóknak egy csapat munkát kell végezni a beteg felépülése érdekében. A mi szoftverünk ezt az irányt támogatja.

Asztali alkalmazás:

Az asztali alkalmazásunk orvosok, illetve egészségügyben dolgozók számára készült. Az alkalmazásban az orvosnak lehetősége van új páciensek, orvosok, gyógyszerek, betegségek, leletek felvételére, törlésére, illetve adataiknak módosítására.

Tervezési folyamat:

Az alkalmazás kinézetének tervét készítettük el, az elemek elhelyezkedését, illetve az alap fukciókat terveztük meg.

Bejelentkezés:

A bejelentkezés során az orvos a kapott azonosítójával és a hozzá tartozó jelszóval tud bejelentkezni.

A képen szöveg, rajztábla látható

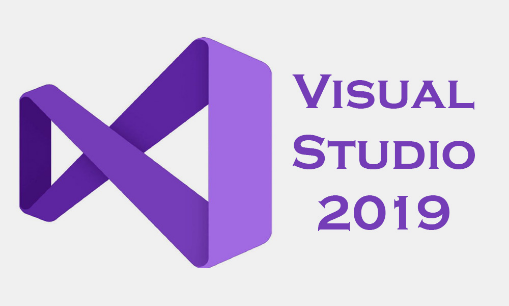
Automatikusan generált leírás

Alkalmazás kinézete:

Az alkalmazás felső sarkában a logó helyezkedik el mellette pedig az alkalmazás neve. Bal oldalon található a menü.

A képen szöveg, rajztábla látható

Automatikusan generált leírás

Alkalmazott programok, nyelvek:

Az asztali alkalmazásunk fejlesztéséhez a Visual Studio 2019 szoftvert alkalmaztuk, amely ideális fejlesztőkörnyezetet biztosított számunkra. A kódot C# nyelven írtuk és a felhasználói felülethez a WPF keretrendszert használtuk. A WPF egy asztali alkalmazások grafikus felhasználói felületének .NET alapokon történő kialakítására szolgáló alkalmazásprogramozási felület. A projektünk az MVVM programozás alapelvei alapján készült.

MVVM:

Modell (Model): itt találjuk az alkalmazásunk "logikáját".

Nézet (View): itt találjuk a (grafikus) felületünk vezérlőit.

Nézetmodell (ViewModel): itt követjük nyomon a modellben lévő változásokat.

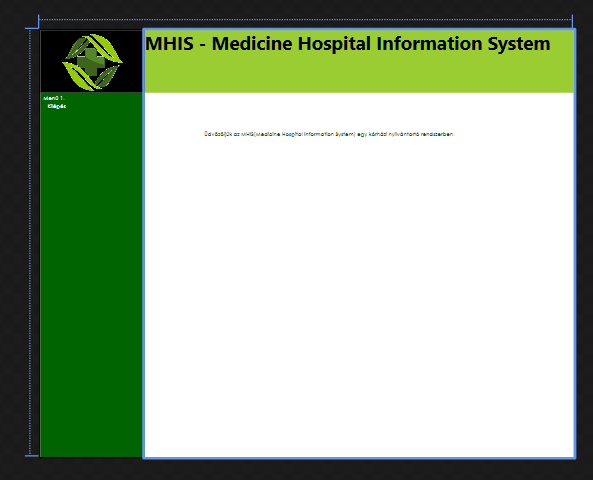
Fejlesztési folyamat:

1. Visual Studio 2019 segítségével létrehoztunk egy üres WPF alkalmazást.

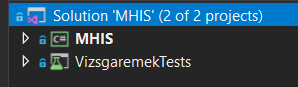
A képen szöveg látható

Automatikusan generált leírás

1. Elkészítettük az alkalmazás kezdőoldalát a tervek alapján, azaz a MainWindow.xaml fájlt.



1. Elkészítettük a teszt projektet.



1. A következő lépésben az első menüpontot készítettük el, amelyen a szoftver adatait lehet megtekinteni, illetve egy gomb segítségével a fejlesztők nevét is lehet kérdezni. A visszalépés ikon a jobb felső sarokban a kezdőoldalra visz vissza, a kilépés menüpont pedig bezárja a programot.

A képen szöveg látható

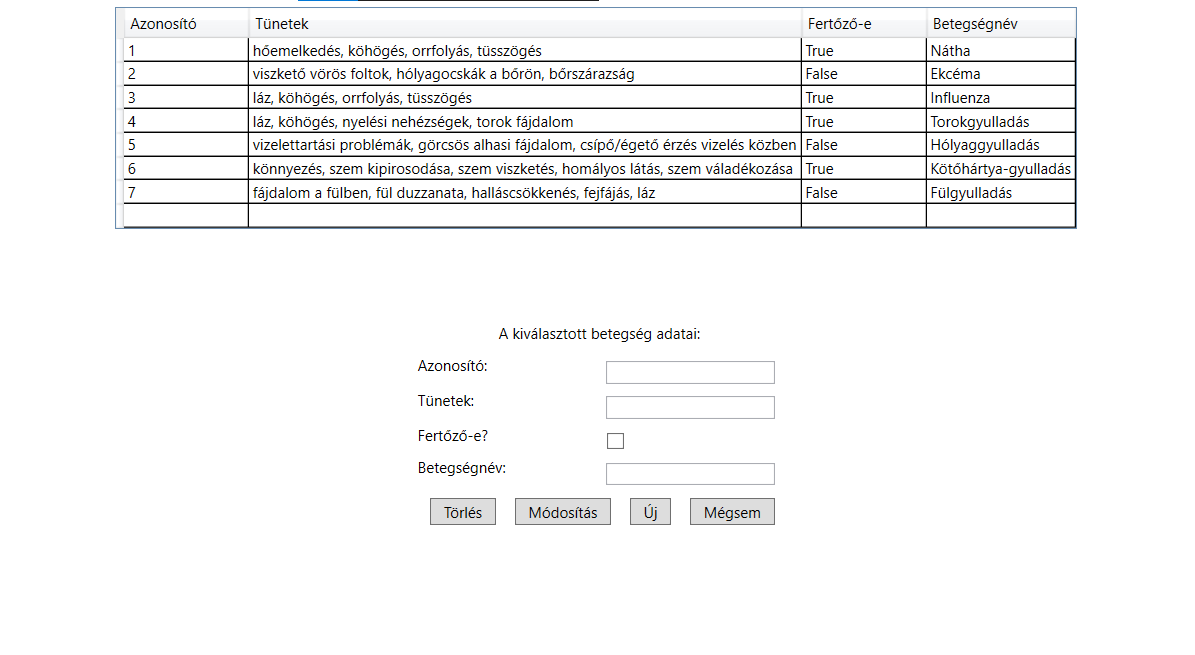
Automatikusan generált leírás

1. Lehetőséget adunk az adatbázis helyének kiválasztására.

A képen szöveg látható

Automatikusan generált leírás

1. Az adatokat egy táblázatban jelenítjük meg minden osztály esetén.

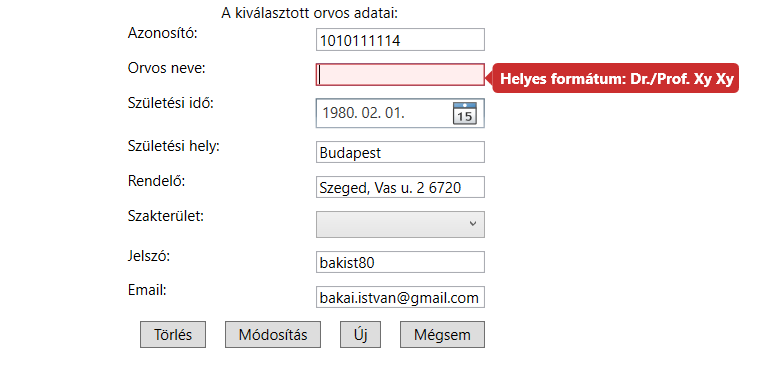


1. A kiválasztott rekord esetén megjelennek az adatok a megfelelő helyen.

A képen szöveg látható

Automatikusan generált leírás

1. A helyes formátumok miatt validációt is készítettünk a szükséges helyekre. Helyes formátum esetén a piros üzenet eltűnik.

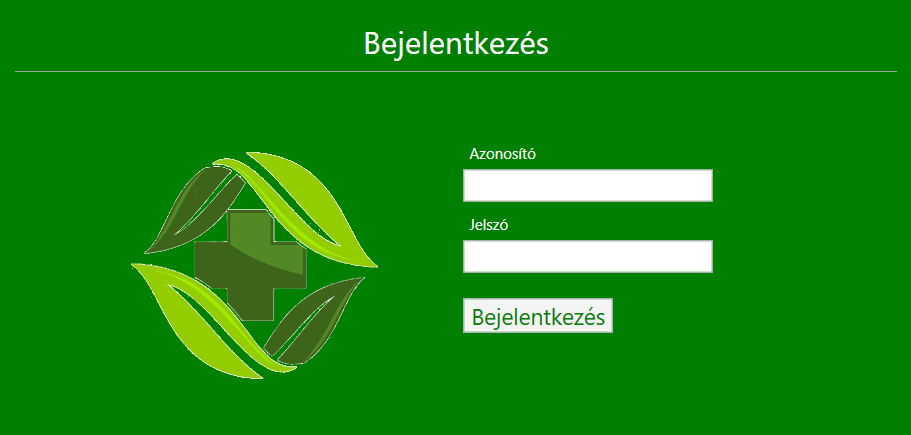


1. Az alkalmazást összekötjük a helyi adatbázissal.

A képen szöveg látható

Automatikusan generált leírás

1. A program a bejelentkezéssel kezd, az orvost vagy az adminisztrátort a megfelelő azonosító és jelszó megadása után automatikusan tovább irányítja az alkalmazás kezdőoldalára. Nem létező azonosító vagy hibás jelszó esetén hibaüzenetet küld.



Mapparendszer:

Mapparendszerünket igyekeztünk minél átláthatóbbra csinálni. Minden mappának egyértelmű nevet adtunk, adott mappában csak olyan fájlok vannak, amelyek a mappa nevének megfelelő funkciókat látnak el, így minden fájl könnyen megtalálható.

A képen szöveg látható

Automatikusan generált leírás

Gombok:

1. Kilépés gomb: amennyiben a felhasználó rákattint a program automatikusan bezárul.

A képen szöveg látható

Automatikusan generált leírás

1. Menü navigáció: a felhasználó által kiválasztott oldalt nyitja meg.

A képen szöveg látható

Automatikusan generált leírás

1. Módosítás gomb: a kiválasztott elem adatainak módosításáért felel, különböző adatbázis források esetén máshogy viselkedik.

A képen szöveg látható

Automatikusan generált leírás

1. Törlés gomb: kiválasztott adat törléséért felel, különböző adatbázis források esetén máshogy viselkedik.

A képen szöveg látható

Automatikusan generált leírás

1. Beillesztés gomb: új rekordot hoz létre, különböző adatbázis források esetén máshogy viselkedik.

A képen szöveg látható

Automatikusan generált leírás

Fontosabb metódusok:

1. MakeTestData(): ez a metódus felelt azért, hogy helyi adatbázis kapcsolat nélkül is tesztelhessük az alkalmazásunk, ehhez szükséges volt egy Model osztály amely a osztályok adatainak felépítését tárolja, ez a metódus a Repository rétegben készült el.

A képen szöveg látható

Automatikusan generált leírás

1. Update(): az adatok frissítéséért felel, ha hozzáadunk egy új rekordot, vagy törlünk, akkor ez a metódus frissíti az addig megjelenített adatokat, a metódus a Repository rétegben található.

A képen szöveg látható

Automatikusan generált leírás

1. Navigáció: azért felel, hogy a megfelelő oldalt nyissa meg a felhasználó számára.

A képen szöveg látható

Automatikusan generált leírás

*Szerző: Nagygéci Noémi*

Mobil alkalmazás:

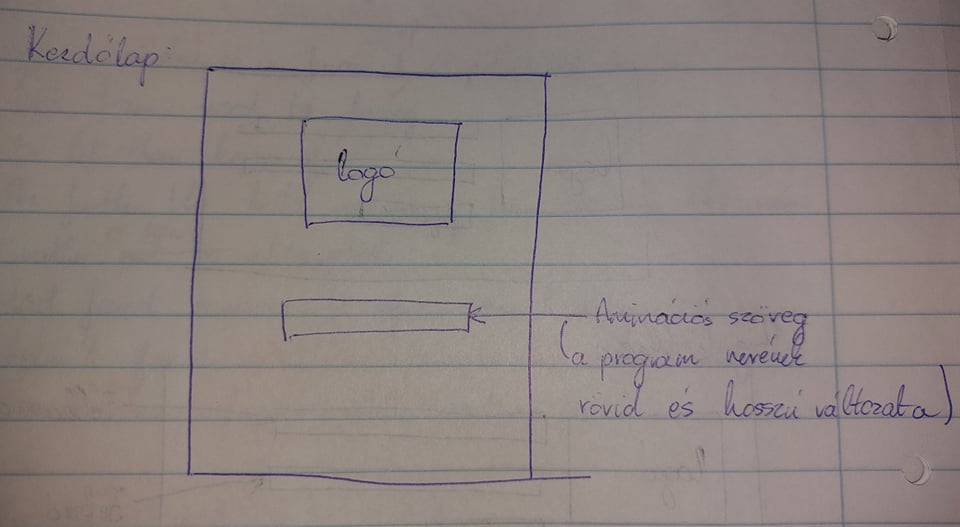
Az mobil alkalmazásunk páciensek, hétköznapi emberek számára készült. Az alkalmazásban a felhasználónak lehetősége van az adatbázisban tárolt orvosok nevét, irodáját, illetve elérhetőségét megtekinteni.

Tervezési folyamat:

Az alkalmazás kinézetének tervét készítettük el, az elemek elhelyezkedését, illetve az alap fukciókat terveztük meg.

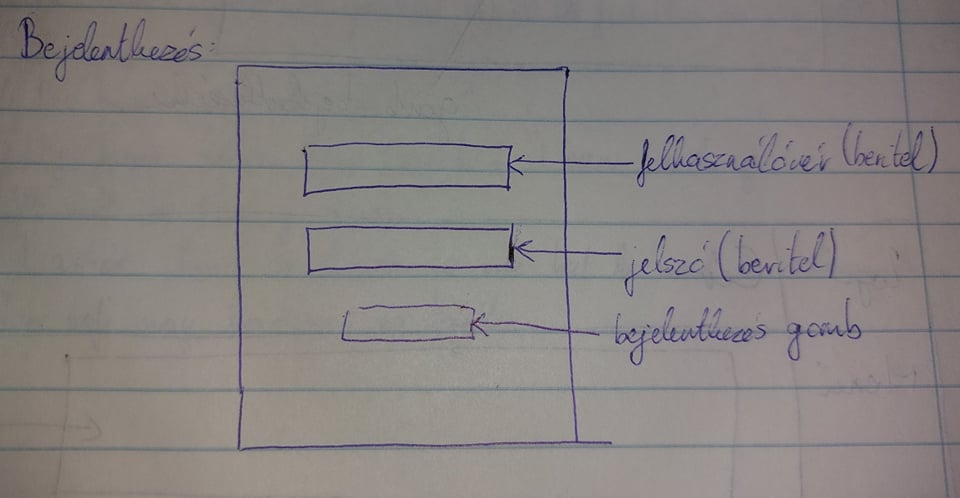
Kezdőlap:

A kezdőlapon középen megjelenik az applikáció logója, alatta pedig animációként a program rövidített és teljes neve.



Bejelentkezés:

A bejelentkezés során a felhasználó a megfelelő felhasználónév-jelszó kombinációjával tud bejelentkezni.



Menü:

A képen szöveg, rajztábla látható

Automatikusan generált leírás

Orvosok keresése:

A felhasználó megtekintheti az adatbázisban tárolt orvosok nevét, rendelőjének címét és e-mailcímét.

A képen szöveg, rajztábla látható

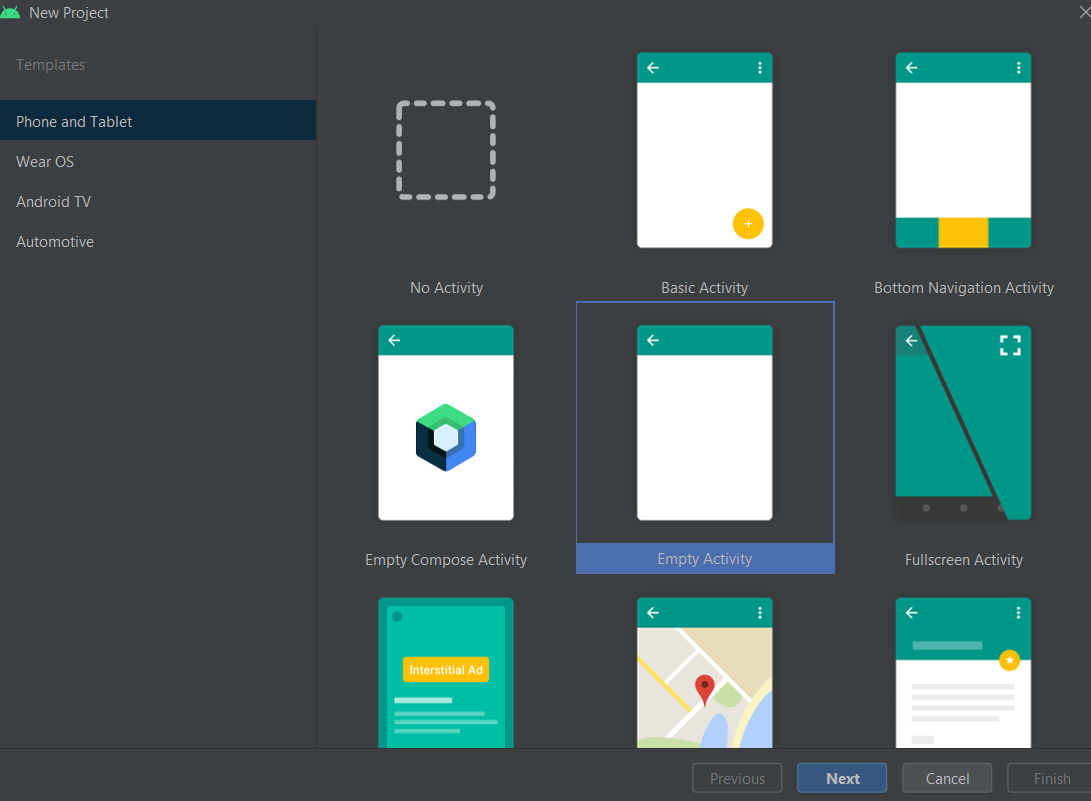
Automatikusan generált leírás

Alkalmazott programok, nyelvek:

Mobilalkalmazásunkat Android Studio segítségével készítettük, Java nyelven. Az Android Studio egy integrált fejlesztőkörnyezet Android alkalmazások fejlesztéséhez.

Fejlesztési folyamat:

1. Létrehoztunk egy Empty Activity-t Android Studio-ban.



1. A main\_activity.xml kapott egy színátmetes hátteret, amelyet egy külön drawable fájlban kellett létrehozni és háttérképnek beállítani. Az oldal közepére beillesztettük a logót.



1. Egy új array.xml fájlban létrehoztuk az animációhoz szükséges szövegeket.

A képen szöveg látható

Automatikusan generált leírás

1. FaidingTextView létrehozása az animáció megjelenítéséhez.

A képen szöveg látható

Automatikusan generált leírás

1. Kattintás után megjelenik a bejelentkezés oldala.

A képen szöveg látható

Automatikusan generált leírás

1. Bejelentkezés oldal elkészítése a tervek alapján.

A képen szöveg látható

Automatikusan generált leírás

1. A bejelentkezés lebonyolításáért egy PHP fájl felel, ami kapcsolódik az adatbázishoz és csak a megfelelő felhasználónév-jelszó kombinációval engedi a felhasználót az alkalmazásba.

A képen szöveg látható

Automatikusan generált leírás

1. Menü és elemeinek létrehozása.

A képen szöveg látható

Automatikusan generált leírás

1. Menü navigációjának elkészítése.

A képen szöveg látható

Automatikusan generált leírás

1. Az oldalak feltöltése a megfelelő tartalommal, az orvosok kereséséhez szintén egy PHP fájlt készítettünk. A fájl az adatbázistól kapott információkat átalakítja egy JSON objektummá.

A képen szöveg látható

Automatikusan generált leírás

1. Az orvosok adatainak megjelenítése egy-egy külön ListView-ban.

A képen szöveg látható

Automatikusan generált leírás

Szerző: Böröcz Evelin Tilda, Nagygéci Noémi

Webes alkalmazás:

A webes alkalmazás a páciensek számára bejelentkezés után lehetőséget nyújt vizsgálatainak, személyes adatainak megtekintésére, gyógyszereinek megtekintésére, feltöltésére, törlésére, illetve orvosok adatainak megtekintésére.

Tervezési folyamat:

Kezdőoldal:

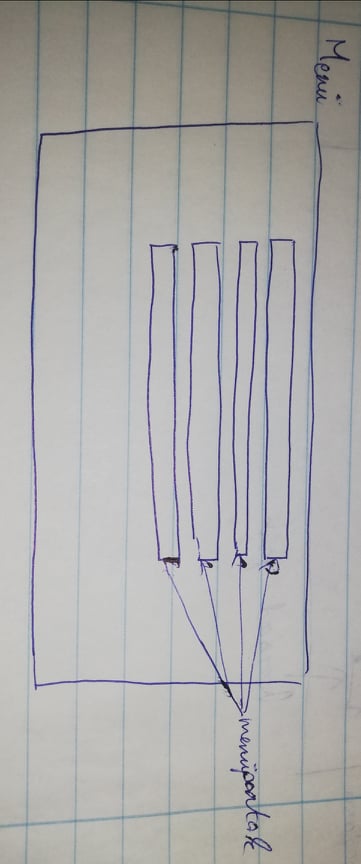
A kezdőoldalon a felhasználó be tud jelentkezni saját fiókjába.

A képen szöveg, rajztábla látható

Automatikusan generált leírás

Menü:

A bejelentkezés után egy menü jelenik meg ahol a funkciók nevei egymás alatt jelennek meg.



Személyes adatok megtekintése:

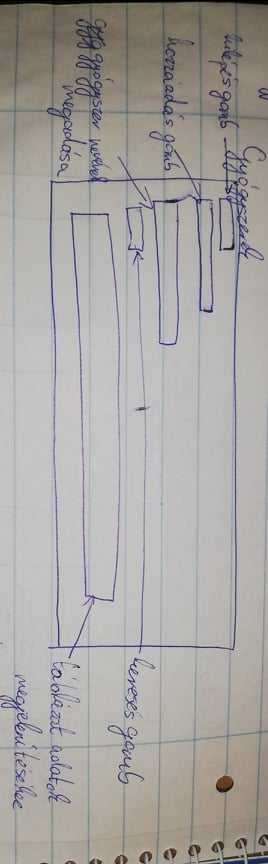
Itt tekintheti meg a felhasználó személyes adatait, TAJ számának megadásával.

A képen szöveg, rajztábla látható

Automatikusan generált leírás

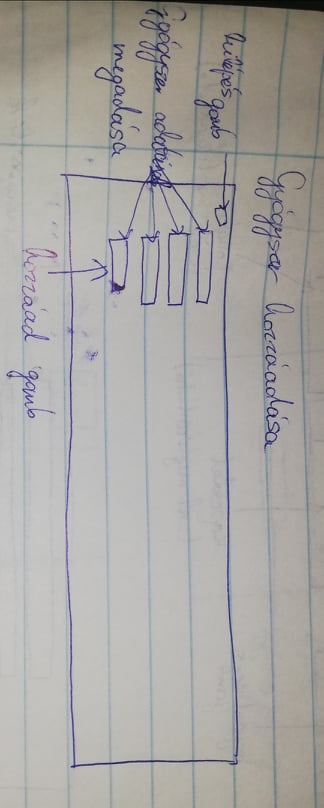
Gyógyszerek megtekintése:

A felhasználó megtudja nézni gyógyszereit egy táblázatban.



Gyógyszerek hozzáadása:

A felhasználó gyógyszereit vagy vitaminjait is hozzáadhatja az adatbázishoz.



Orvosok keresése:

Orvosok adatai jelennek meg egy táblázatban.

A képen szöveg, rajztábla látható

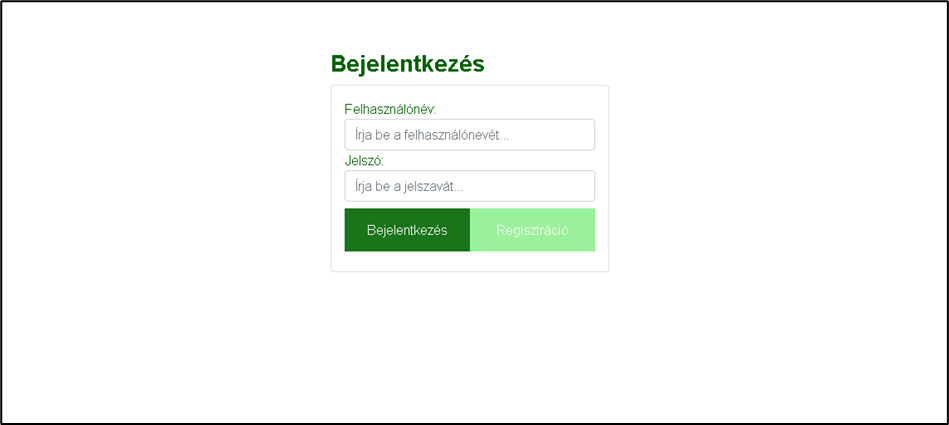
Automatikusan generált leírás

Alkalmazott programok, nyelvek:

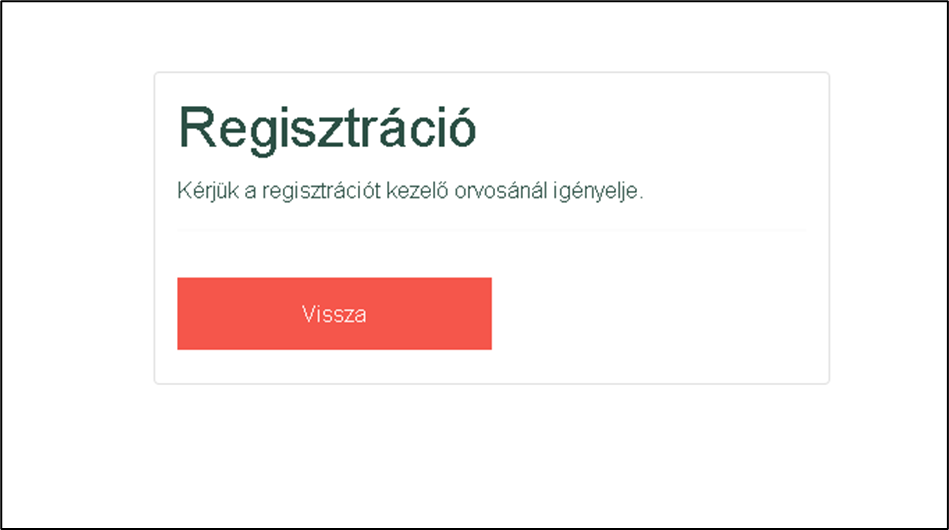
A webes alkalmazásunkhoz Visual Studio Code-ot használtunk és Vue.js alapú könyvtárt alkalmaztunk, amelyet JavaScript nyelven írtunk. A Visual Studio Code egy hatékony forráskódszerkesztő. A Vue.js egy JavaScript könyvtár, amely lényegesen segíti a JavaScript alapú oldalak forráskódjának megírását.

Fejlesztési folyamat:

1. Visual Studio Code-ban létrehoztuk egy Vue.js keretrendszer alapú projektet.
2. Bejelentkezés oldal elkészítése.



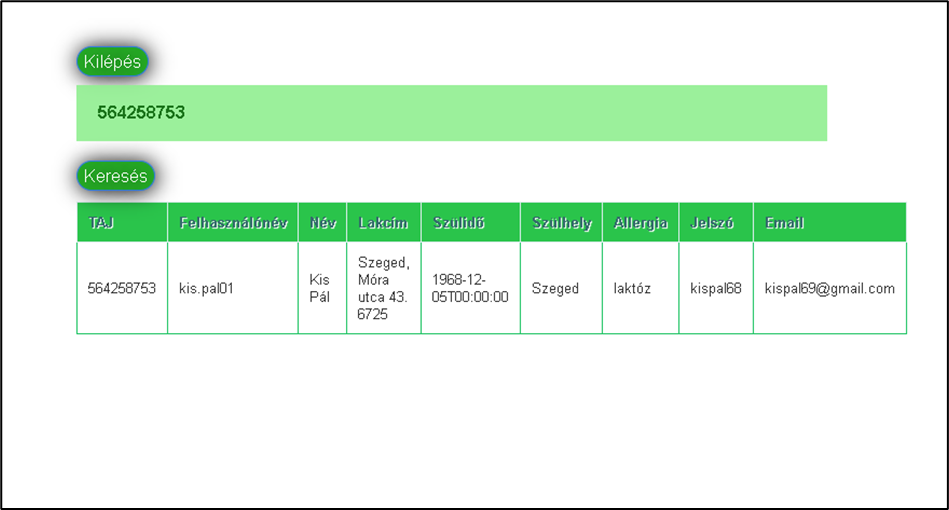
1. Regisztráció gombra kattintva egy oldalra visz amely tájékoztatja a felhasználót a regisztráció menetéről.



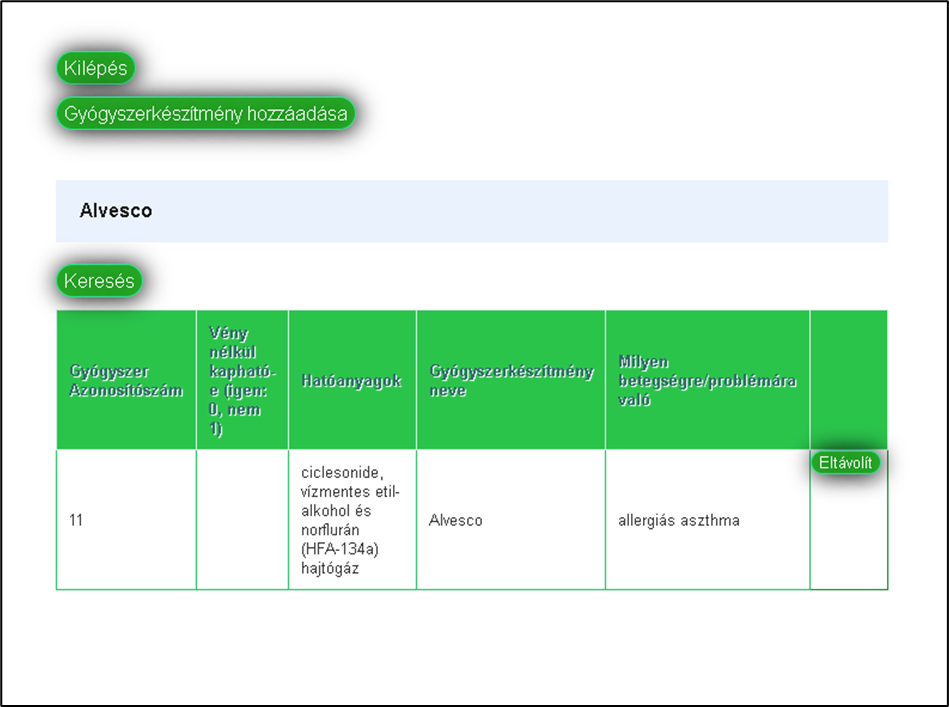
1. Sikeres bejelentkezés után a felhasználó kiválaszthatja, hogy milyen funkciót szeretne használni.



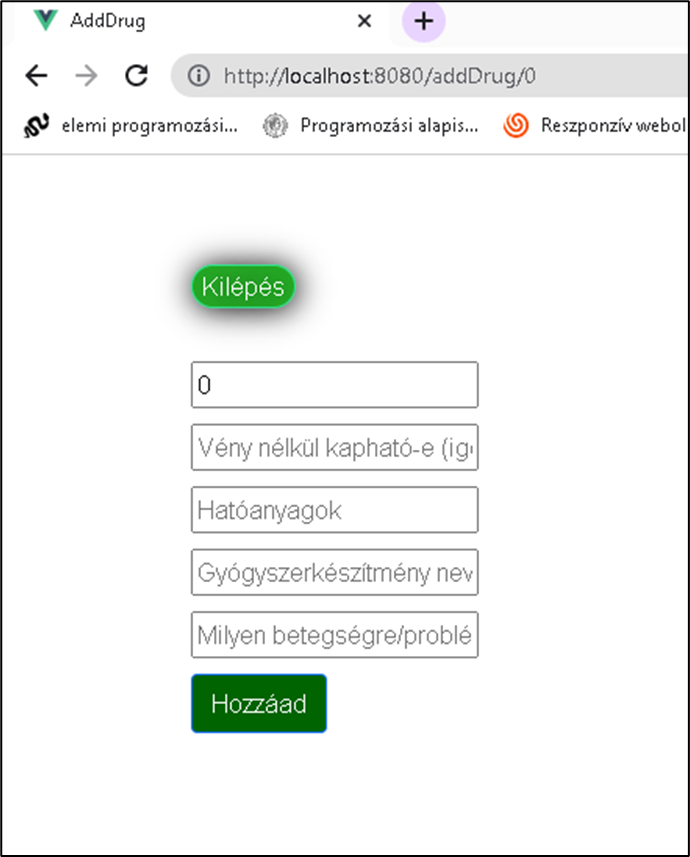
1. Személyes adatainak megtekintése TAJ szám alapján történik.



1. Gyógyszereit megtekintheti a gyógyszer nevének megadása után, itt lehetősége van a gyógyszer eltávolítására is.



1. A felhasználó új gyógyszert is tud hozzáadni, ez a funkció azért jött létre, hogy ha a beteg vitaminokat vagy táplálékkiegészítőket szed azt is tárolhassa az adatbázisban.



1. Orvosok adatainak megtekintése, itt megjelenik az orvos neve, rendelője és emailcíme.

A képen asztal látható

Automatikusan generált leírás

*Szerző: Böröcz Evelin Tilda, Baracsi Kinga*

Web API:

Tervezési folyamat:

A Web API tervezés közben figyelembe kellett vennünk, hogy milyen funkciókat szeretnénk megvalósítani a webes alkalmazásunkban.

Alkalmazott programok, nyelvek:

Visual Studio 2019-et használtunk a API elkészítéséhez, a kódot pedig C# nyelven írtuk.

Fejlesztési folyamat:

1. A felhasználó bejelentkeztetése Basic Authentication-nel, így a felhasználót könnyen és biztonságosan be lehet léptetni a webes felületre.

A képen szöveg, beltéri látható

Automatikusan generált leírás

1. Adatbázis lemodellezése paranccsal.

A képen szöveg, beltéri, képernyőkép látható

Automatikusan generált leírás

1. Minden tábla lemodellezve.

A képen szöveg látható

Automatikusan generált leírás

1. Repository rétegben authentikáció.

A képen szöveg látható

Automatikusan generált leírás

1. Kontrollerek megírása, minden kontroller a megfelelő adatokat kéri, küldi vagy éppen törli.

A képen szöveg látható

Automatikusan generált leírás

Szerző: Böröcz Evelin Tilda

Adatbázis:

Egyedek: Páciens, Orvos, Betegség, Gyógyszer, Vizsgálat

Tulajdonságok:

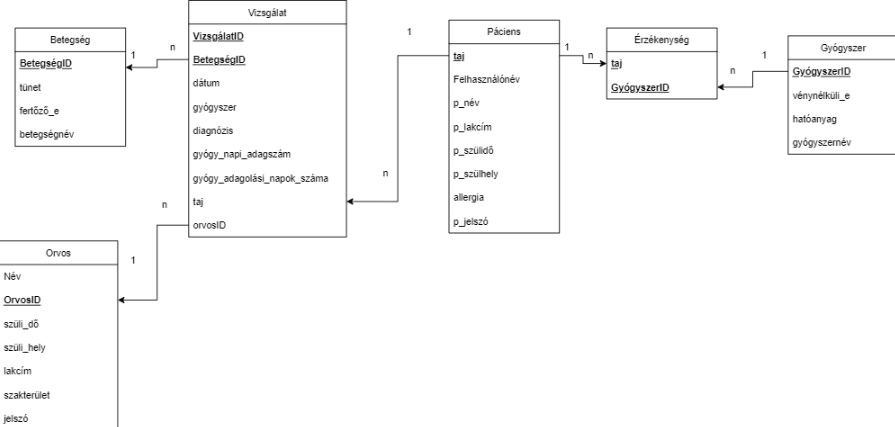
Páciens (**taj**, szul\_ido, szul\_hely, felhasznalonev, jelszo, nev, lakcim, allergia, email)

Orvos (**orvosID**, nev,szul\_ido, szakterulet, jelszo, szul\_hely, lakcim)

Betegség (**betegsegID**, tunet, betegsegnev, fertozo\_e)

Gyógyszer (**gyogyszerID,** gyogyszernev, hatoanyag, veny\_nelkuli\_e)

Vizsgálat (**vizsgalatID,** datum, diagnozis, gyogy\_napi\_adag, gyogy\_adagolasi\_napok\_szama)



Egyedek és tulajdonságaik (adattípusok):

*Páciens*

**taj** INT – elsődleges kulcs

felhasznalonev VARCHAR(20) – UNIQUE, a páciens felhasználóneve a programhoz

p\_nev VARCHAR(30) – a páciens neve

p\_lakcim VARCHAR(50) – a páciens lakcíme

p\_szulido DATE – a páciens születési dátuma

allergia VARCHAR(100) – a páciens allergiái

p\_jelszo VARCHAR(20) – UNIQUE, a páciens jelszava

email VARCHAR(10) – UNIQUE, a páciens email-je

nehéz

adag p\_szulhel

*Orvos*

**orvosID** INT – AUTO\_INCREMENT – elsődleges kulcs

nev szöveg VARCHAR(30) – az orvos neve

szul\_ido DATE – az orvos születési dátuma

szul\_hely VARCHAR(30) – az orvos születési helye

lakcim VARCHAR(50) – az orvos lakcíme

szakterulet VARCHAR(30) – az orvos szakterülete

jelszo VARCHAR(20) – UNIQUE, az orvos jelszava

mértékegység szul\_ido (

*Betegség*

**betegsegID** INT – AUTO\_INCREMENT – elsődleges kulcs

tunet VARCHAR(100) – a betegség tünetei

fertozo\_e BINARY(2) – a betegség fertőző-e (0: igen, 1:nem)

betegsegnev VARCHAR(30) - betegség megnevezése

felhasználónév fertozo\_e (

*Gyógyszer*

**gyogyszerID** INT – AUTO\_INCREMENT – elsődleges kulcs

venynelkuli\_e BINARY(2) – a gyógyszer kapható-e recept nélkül (0:igen, 1:nem)

hatoanyag TEXT – a gyógyszer összetevői

gyogyszernev VARCHAR(30) – a gyógyszer neve

alapanyagID

hatoanyag (

*Vizsgálat*

**vizsgalatID** INT – AUTO\_INCREMENT – elsődleges kulcs

**betegsegID** INT – idegen kulcs

datum DATE – a vizsgálat dátuma

gyogyszer VARCHAR(30)– a gyógyszer neve

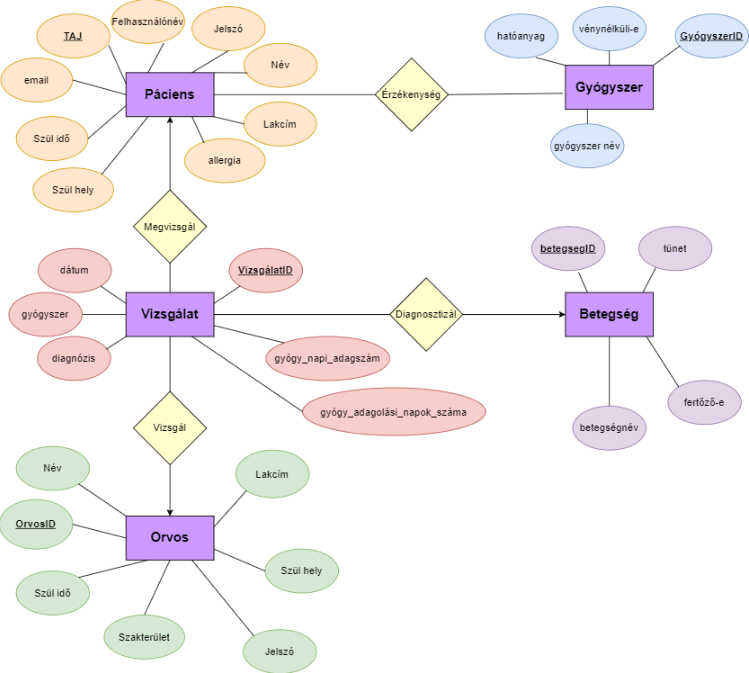
diagnozis TEXT – a vizsgálat során az orvos által felállított diagnózis a páciensre

gyogy\_napi\_adagszam – INT – azokon a napokon amikor be kell szednie a betegnek a gyógyszert, ez tárolja a napi gyógyszer adagszámot

gyogy\_adagolasi\_napok\_szama – INT – időintervallum, ami alatt a betegnek szednie kell a gyógyszert

taj INT – idegen kulcs

orvosID INT – idegen kulcs



*Szerző: Baracsi Kinga, Böröcz Evelin Tilda*

GitHub:

A GitHub egy szoftverfejlesztési verziókövetési-szolgáltatás, amely a Git-en alapul. Verziókövetést és forráskódkezelést is lehetővé tesz a fejlesztők számára.

Csapatmunka:

A GitHub meglehetősen egyszerűvé és hatékonnyá teszi a csapatban történő fejlesztést, segítségével nyomon lehet követni, hogy ki milyen új fájlt adott hozzá, fejlesztést, módosítást vagy törlést hajtott végre az adott forráskódban. Amennyiben a fejlesztők egyértelmű neveket adnak branch-eiknek a csapat többi tagja könnyen be tudja azonosítani, hogy mit készített.

A képen szöveg látható

Automatikusan generált leírás

*Szerző: Baracsi Kinga*