

**SZEGEDI SZAKKÉPZÉSI CENTRUM
VASVÁRI PÁL GAZDASÁGI ÉS INFORMATIKAI
SZAKGIMNÁZIUMA**
Az 54 213 05 számú Szoftverfejlesztő szakképzés záródolgozata

Medi-2020

TURCSÁNYI ÁRON BÁLINT
SZEGED
2020

TARTALOMJEGYZÉK	
HALLGATÓI NYILATKOZAT.....	2
BEVEZETÉS.....	3
1. FELHASZNÁLÓI DOKUMENTÁCIÓ.....	3
1.1 AZ ASZTALI ALKALMAZÁS DOKUMENTÁCIÓJA.....	4
1.2 A WEBALKALMAZÁS DOKUMENTÁCIÓJA.....	15
2. FEJLESZTŐI DOKUMENTÁCIÓ.....	23
2.1 AZ ADATBÁZIS.....	24
2.2 AZ ASZTALI ALKALMAZÁS DOKUMENTÁCIÓJA.....	26
2.3 WEBALKALMAZÁS DOKUMENTÁCIÓJA.....	28
3. FELHASZNÁLT KERETRENDSZEREK.....	29
4. MELLÉKLET.....	30
KÖSZÖNETNYILVÁNÍTÁS.....	32

HALLGATÓI NYILATKOZAT

Alulírott *Turcsányi Áron Bálint*, a Szegedi Szakképzési Centrum Vasvári Pál Gazdasági és Informatikai Szakgimnáziuma hallgatója kijelentem, hogy a *Medi-2020 Betegfelvétel* című záródolgozatom a saját munkám.

A záródolgozat mellékletét képezi egy CD lemez, amely tartalmazza jelen dokumentációt elektronikus formában (docx és pdf), valamint az asztali és webalkalmazás projektjeit és az adatbázis állományt.

.....

Kelt: Szeged, 2020-08-30

BEVEZETÉS

Szakedolgozatom témájának egy fiktív, *Medi-2020* nevű magánkórház betegnyilvántartó rendszerének létrehozását választottam. Általában, akik ennek a képzésnek a záródolgozatát fejlesztik a szakmájukhoz igencsak közel álló témát választanak, én valami mást szerettem volna implementálni.

A szoftverfejlesztés világában a legkülönbébb rendszereket fejlesztik vagy tartják karban napról-napra úgy, mint számlázóprogramok, hivatali adminisztrációs rendszerek, banki applikációk és még sorolhatnánk. Egy ilyen általánosan szükséges rendszer, mint egy magánkórházi adminisztrációs rendszer ötlete fogalmazódott meg bennem, ezt az alkalmazás-rendszert igyekeztem kifejleszteni, amely magába foglal egy asztali applikációt, amely a kórház belső adminisztratív feladatait hivatott automatizálni; egy webalkalmazást, amely a kórház regisztrált betegeinek nyújt szolgáltatásokat.

Ennek a rendszernek a dokumentációja olvasható a továbbiakban fejlesztői és felhasználói oldalról megközelítve a kész funkciók használatát, illetve azok implementációs mikéntjét.

1. FELHASZNÁLÓI DOKUMENTÁCIÓ

Az alábbiakban olvasható a *Medi-2020* számára kifejlesztett alkalmazásrendszer asztali és webalkalmazásának a felhasználói dokumentációja.

Az asztali szoftver a rendszer adminisztrációs feladatait ellátó alkalmazás, amely a magánkórház dolgozói számára nyújtanak segítséget a napi adminisztráció elvégzéséhez, beleértve az ügyfelek és a kórház orvosainak pontos nyilvántartását, illetve a cég pénzügyi nyilvántartását hivatott nyomon követni.

A felhasználói dokumentáció részletes tájékoztatást ad a kifejlesztett rendszer eddigi megfelelő használatához.

1.1 AZ ASZTALI ALKALMAZÁS DOKUMENTÁCIÓJA

A program elindításakor egy ablak jelenik meg a képernyőhöz viszonyítva teljes méretben. A szoftverben az ablak tetején elhelyezkedő eszköztár lehet a legnagyobb segítségünkre, annak használatával lehetséges a menüpontok között navigálnia a felhasználónak. (1. ábra)



Az első menüpont a betegek kezelésére szolgáló *Betegfelvétel* menüpont. Amennyiben az adminisztrátor ezt megnyitja, a fő ablakban egy kisebb ablak jelenik meg, amelyben táblázatos formában megjelenik a Medi-2020 minden eddigi páciense. Ezen funkció kifejlesztését az indokolta, hogy bármely dolgozó gyorsan és érthetően kapjon információkat egy adott páciens adatairól, ha szüksége van rá. (2. ábra)

Medi-2020 Bt. adminisztrációs szoftver

Betegfelvétel Orvosok Orvosi rendelések Pénzügyek Kélpés

Betegfelvétel Keresés Mentés

TAJ szám	Vezetéknév	Keresztnev	Irányítószám	Település	Lakcím	E-mail cím	Telefon
000-112-344	Harkovszky	Péter	6722	Szeged	Kossuth Lajos sgt. 18.	peterhark@gmail.com	06203219987
110-987-444	Halmosi	József	6726	Szeged	Féau utca 56.	jozsefhal@gmail.com	06204449872
123-456-789	Turcsányi	Áron	6725	Szeged	Példa utca 45.	turcsanyi@gmail.com	06202345783
200-198-333	Halász	Amarilla	6710	Szeged	Szentmihály út 51.	amarilla@yahoo.com	06207891813
221-009-788	Hámos	Gábor	6753	Szeged	Buda Nagy Antal u. 12.	hamosgabor@gmail.com	06208774443
228-003-112	Balogh	Károly	6791	Szeged	Szémalom utca 57.	karezbalo@gmail.com	06303255609
312-899-379	Prónai	András	6727	Szeged	Ladvánasky József utca 40.	ladvannad@yahoo.com	06306328465
423-353-811	Maczeka	Judit	6727	Szeged	Fekete János utca 21.	feketeja93@gmail.com	06207803330
427-383-390	Lukács	Tamás	6771	Szeged	Szőregi út 131.	lkacsam@gmail.com	06304959666
432-881-920	Harkai	Gyula	6725	Szeged	Hajnal utca 6.	gyulaharka@gmail.com	06303342892
438-211-064	Mihály	Péter	6710	Szeged	Alsónyomás sor 5.	pemil@citromail.hu	06204869922
438-571-597	Koronkai	hett	6791	Szeged	Barátság utca 22/8	baratsag@gmail.com	06704719678
455-122-344	Hornváth	Géza	6722	Szeged	Jósika utca 11.	gezahornvath@gmail.com	06205559871
462-855-211	Kőnya	Ágnes	6721	Szeged	Óthalom utca 53.	othalom@gmail.com	06704385721
472-571-371	Rácz	Béla	6727	Szeged	Ortutay utca 47.	ortutay@yahoo.com	06306318347
480-682-157	Módr	Ferenc	6725	Szeged	Göz utca 38.	folgae923@gmail.com	06306307779
483-150-463	Gelencsér	Sándor	6722	Szeged	Szentháromság utca 13.	sandorgelencser@yahoo.com	06304062022
488-201-486	Magda	István	6722	Szeged	Roosevelt tér 7-11	magdaist019@gmail.com	06306524565
499-443-217	Gera	Ágnes	6728	Szeged	Ladvánasky utca 7.	agnesgera@citromail.hu	06208739544
554-332-021	Ipacs	Béla	6723	Szeged	Söllér sor 6/A	belapacs@citromail.hu	06209012227

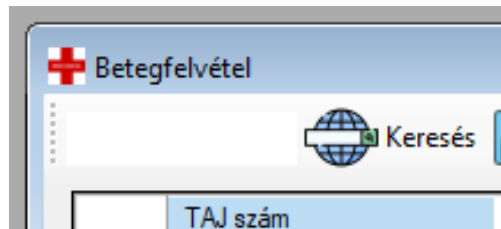
Összesen: 31

<< < 1/2 > >>

2. ábra Betegfelvétel funkció

A beteg nyilvántartáson többféle művelet végezhető, úgy mint a keresés, adatok mentése, valamint a könnyebb olvashatóság érdekében nem egy oldalon jelenik meg az összes ügyfél, hanem csak egy része. A bal alsó közepén elhelyezkedő nyilakkal tudunk lapozni a páciensek adatai közül. A bal, két külső dupla nyíllal az ügyfél táblázat elejére ugorhat a felhasználó, a jobb oldali dupla nyíllal ellátott gombbal pedig a nyilvántartás legvégére. A két belső nyíllal – értelemszerűen az irányokat figyelembe véve – előre és hátra lapozható egyesével a táblázat. Középen egy számláló jelzi az aktuális oldalszámot baloldalon, a jobb oldali szám az összes oldalak számát jeleníti meg. A jobb alsó sarokban lévő címke a nyilvántartásból lekérdezett összes sorok számát jelenti.

Keresést indítani a kisablak bal felső sarkában lévő szöveges mezőjébe írva lehetséges. Bármely beírandó szórészletre tud keresni az alkalmazás. A folyamat a mellette lévő Keresés gomb lenyomásával hajtható végre. (3. ábra)



3. ábra Keresés funkció

A további funkciók az új beteg hozzáadása, meglévő adatainak módosítása, illetőleg bizonyos feltételek mellett meglévő beteg törlése a páciens nyilvántartásból. Ezen művelet kifejlesztését az indokolta, hogy az olyan páciensek, úgymint idősebb ügyfelek nem rendelkeznek esetleg interneteléréssel, és nincs lehetőségük regisztrálni a Medi-2020 weboldalán, hanem a magánkórházban személyesen megjelenve a szoftvert kezelő alkalmazott ott helyben regisztrálhassa a cég adatbázisába.

Új ügyfél hozzáadásához a táblázaton, kattintva bárhova egy jobb egérklikkel előhívható egy párbeszéd ablak, amelyen az előbbieken felsorolt funkciók megtalálhatók. (4. ábra)

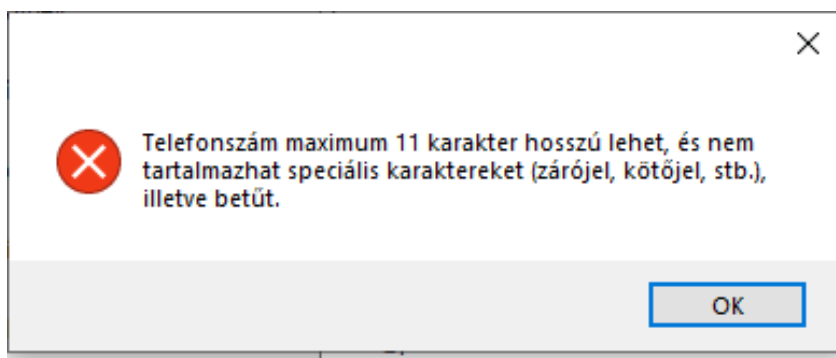
Vezetéknév	Keresztnév
Hankovszkv	Péter
Halm	sef
Turc	n
Halá	anilla
Hámos	Gábor

4. ábra Párbeszéd ablak
menüpontjai

A hozzáadása menüpontra kattintva egy újabb ablak jelenik meg, amely az új ügyfél hozzáadásához szükséges űrlapot jeleníti meg, amely értelemszerűen töltendő ki. (5. ábra)

5. ábra Betegfelvételi űrlap

Az űrlap kitöltése után az OK feliratú gombbal nyugtázható az új páciens felvétele. Előfordulhat a kitöltés során néhány hibalehetőség: az alkalmazás nem fogad el üresen kitöltött mezőt, illetve helytelenül kitöltött adatokat. A teljesség igénye nélkül ilyen hibalehetőség az üresen hagyott beviteli mezők, kisbetűvel kezdődő keresztnév és vezetéknév, település név és lakcím; a rossz formátummal megadott e-mail cím; a telefonszám, amelyet mindenféle speciális karakter nélkül kell megadni és csak számokat tartalmazhat. Az OK gomb lenyomására ellenőrzés fut végig a bevitt adatokon, így elkerülhetők a formailag hibás adatok felvitele. Ezeket a hibákat egy felugró hibaüzenet ablak közli a felhasználóval. (6. és 7. ábrák)



6. ábra Telefonszám hibalehetőség

The screenshot shows a 'Beteg' (Patient) registration window. It contains several input fields: 'Taj szám', 'Jelszó', 'Vezetéknév', 'Keresztnév', 'Irányítószám', 'Település', 'Lakcím', 'E-mail cím', and 'Telefonszám'. At the bottom, there are 'OK' and 'Mégse' buttons. A modal error dialog box is displayed in the center, featuring a red 'X' icon and the text 'Üres bemenet.' (Empty input). The dialog has an 'OK' button.

7. ábra Üres beviteli mezők hibalehetőség

Kivételes eset, amikor már a rendszerben létező TAJ számot adunk meg az ügyfél felvitelénél – esetleg véletlenül másét – olyankor, az OK gomb mellett lévő kis felkiáltójeles ikon fogja jelezni a hibalehetőséget. (8. ábra)

The screenshot shows the same 'Beteg' (Patient) registration window, but now with data entered in all fields: 'Taj szám' (123-456-789), 'Jelszó' (*****), 'Vezetéknév' (Turcsányi), 'Keresztnév' (Aron), 'Irányítószám' (6725), 'Település' (Szeged), 'Lakcím' (Példa u. 46.), 'E-mail cím' (taron@pelda.hu), and 'Telefonszám' (06902944351). The 'OK' button is highlighted with a blue border and has a small red warning icon next to it. A tooltip with the text 'Létező taj szám!' (Existing TAJ number!) is visible below the 'OK' button. The 'Mégse' button is also present.

8. ábra TAJ szám hibalehetőség ikon

Abban az esetben, ha az ellenőrzés folyamata nem talált kivétlnivalót a bevitt adatokban, úgy az OK gomb lenyomása után eltűnik a kisablak és visszatér a táblázatos nézet. Amennyiben nem látható az újonnan felvitt ügyfél sora, úgy a táblázat legvégére kell lapozni, mivel új beteg felvitelkor minden esetben oda fog kerülni. Az oszlopokra kattintva rendezhetők bármilyen tulajdonság alapján a páciensek. A felvitt ügyfelet a táblázatnézet ablakán lévő Mentés gombra kattintva tudjuk véglegesen elmenteni a cég adatbázisába. (9. ábra)



9. ábra Mentés funkció

Lehetőség van a már felvitt beteg adatainak módosítani is. Ekkor a módosítandó ügyfelet ki kell jelölni egy bal egérgattintással a táblázatban, majd azon jobb egérgéppel kiválasztani a kis párbeszédablakból a *Szerkesztés* menüpontot. Ekkor a korábbihoz hasonló űrlap jelenik meg, azzal a különbséggel, hogy minden szöveges mező ki van töltve az ügyfél korábban megadott adataival; illetve a TAJ szám és a jelszó nem módosítható.

A módosítandó adatot tartalmazó szöveges mezőből ki kell törölni a már meglévő adatot és az újat beírni. (10. ábra)

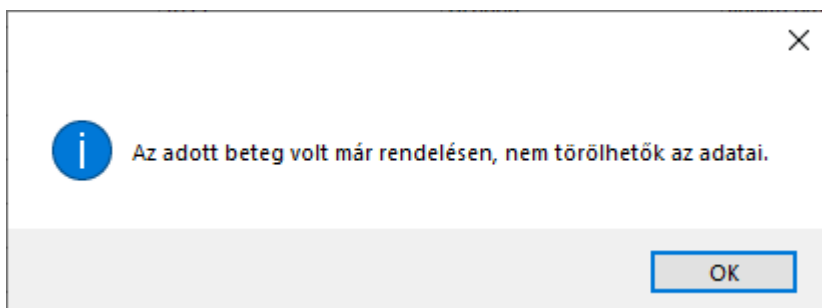
A screenshot of a web form titled "Beteg" (Patient) with a red close button in the top right corner. The form contains several input fields with labels on the left and values on the right. The fields are: TAJ szám (000-112-344), Jelszó (*****), Vezetéknév (Hankovszky), Keresztnév (Péter), Irányítószám (6722), Település (Szeged), Lakcím (Kossuth Lajos sgt. 18.), E-mail cím (peterhank@gmail.com), and Telefonszám (06203219987). At the bottom, there are two buttons: "OK" and "Mégse" (Cancel).

10. ábra Beteg módosítási űrlap

Szintén előfordulhatnak hibalehetőségek a kitöltésnél, itt is felugró ablakok jelzik az ellenőrzés végeredményét. A hibalehetőségek itt kivétel nélkül ugyanazok lehetnek, amelyek az új páciens felvétele esetén – kivételt képez a TAJ szám hibalehetőség, mivel az nem módosítható, illetve a jelszóra vonatkozó megkötések.

Az OK gombra kattintva és hibáktól mentesen az ablak itt is szintén bezáródik, majd az eszköztáron lévő Mentés gombra kattintva menthetők el a módosított adatok.

Az utolsó lehetőség ebben a menüpontban a páciens törlése. Páciens törlése akkor végezhető el, ha még egyetlen egy szolgáltatást sem vett igénybe a Medi-2020-tól, így a neve nem szerepel a vizitek listájában. A későbbi visszakereshetőség végett olyan páciens adatai, akik már legalább egy szolgáltatást igénybe vettek, az adataik nem törölhetők! A törlési folyamat elindításához egy bal egérgéppel ki kell jelölni azt az ügyfelet, aki törölhető és egy jobb egérgéppel előhívva a párbeszéd ablakban a *Törlés* menüpontot kell kiválasztani. Amennyiben sikeres a művelet úgy a táblázatból az adott páciens sora eltűnik és a Mentés gombbal zárható le véglegesen a folyamat. Ellenkező esetben, ha olyan páciensen történik meg a törlés kívánalma, aki nem törölhető, egy felugró ablak hívja fel erre a hibalehetőségre a figyelmet. (11. ábra)



11. ábra Nem törölhető vendég hibalehetőség

Későbbi fejlesztési lehetőség részét képezi a fentmaradó kiskapuk befoltozása, úgy, mint TAJ szám módosítása téves felvitelből. Jelen esetben ez úgy kísérelhető meg – ha még időben kiderül a téves felvitel ténye –, hogy a teljes felhasználót törölni kell, mielőtt szolgáltatást venne igénybe, majd újra fel kell vinni a felhasználót.

Jelszó az asztali alkalmazásban nem módosítható, de az ügyfél felületén a webalkalmazásban igen – az is akkor, ha bejelentkezett munkamenetben eléri a profil módosítási funkciót a beteg – így ez a kiskapu módosításra szorul.

Amennyiben olyan ügyfél kerül az asztali alkalmazásba regisztrálva, akinek nincs internetelérése, úgy az e-mail cím mezőbe kitölthető a magánkórház e-mail címe is, vagy közeli hozzátartozó e-mail címe. Asztali alkalmazásban létrehozott felhasználó automatikus webes felhasználói fiókot von maga után, amely a későbbiekben bármikor használható.

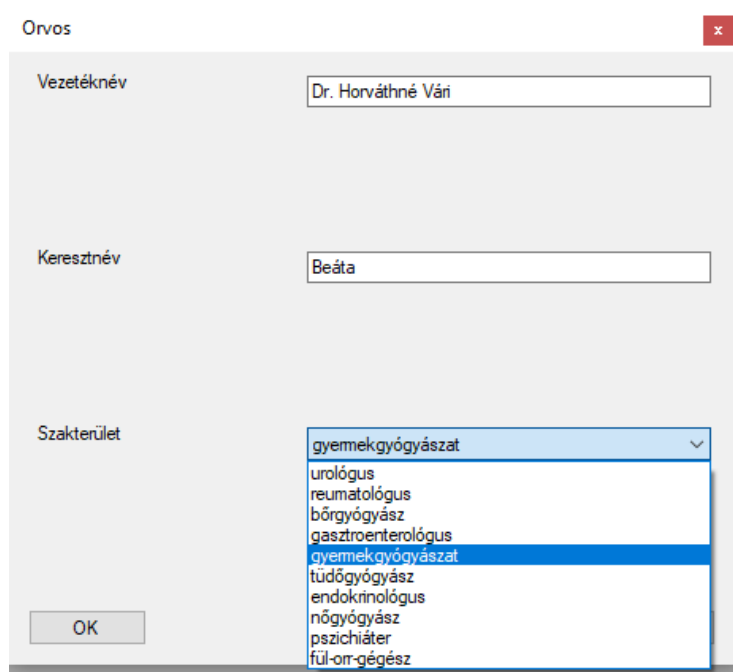
A következő menüpont az *Orvosok* funkció a fő ablak eszköztárában. Ez a Medi-2020-nál dolgozó orvosok nyilvántartása. A menüpontra kattintva egy a betegfelvételhez hasonló táblázatos nézet jelenik meg majd egy kisebb ablakban, ugyanolyan táblázatos nézetben. A táblázat kereshető, menthető, lapozható. Az előzőkhez hasonlóan itt is ugyanúgy működik az új orvos felvétele, törlése és módosítása, ahogy a páciensek esetében: bármikor felvihető új orvos egy kisebb ablakban, kötelezően kitöltendő vezetéknév és keresztnév szöveges mezőkkel, illetve kötelezően kiválasztandó szakterület megnevezés egy legördülő listában (12. ábra)

The screenshot shows a window titled "Orvos" with a close button (X) in the top right corner. Inside the window, there are three input fields: "Vezetéknév" (Surname), "Keresztnév" (First Name), and "Szakterület" (Specialty). The "Szakterület" field is a dropdown menu with a list of medical specialties: urológus, reumatológus, bőrgyógyász, gasztroenterológus, gyermekgyógyász, tüdőgyógyász, endokrinológus, nőgyógyász, pszichiáter, and fül-orr-gégész. The "OK" button is located at the bottom left of the form.

12. ábra Orvos hozzáadása űrlap

Szintén OK gombbal végezhető el a felvitel, ha esetleg hibalehetőség adódik (kisbetűs vezetéknév vagy keresztnév, úgy azt felugró hibaüzenet ablak jelzi a felhasználónak), majd a fő ablakon Mentés gombbal zárható le a folyamat.

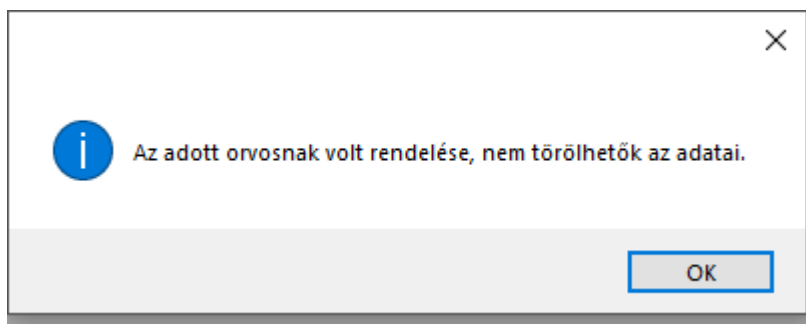
Meglévő orvos szerkesztéséhez itt is ki kell jelölnünk egy bal egérgattintással a módosítandó orvost és a *Szerkesztés* menüpontra kattintva az orvos kitöltött adataival megjelenő ablak jelenik meg. (13. ábra)



13. ábra Orvos szerkesztése űrlap

Az orvosnak bármely adata módosítható. A kis ablakon OK gombbal, a fő ablakon Mentés gombbal zárható le a folyamat.

Az orvosok törlése szintén ahhoz kötött, ami a páciensek törléséhez: még szolgáltatást nem nyújtó orvos törölhető; szolgáltatást már nyújtó orvos a későbbi visszakereshetőség okán nem törölhető. A törölhető orvos szintén kijelölendő egy bal egérgattintással és jobb klikkel a párbeszéd ablakon a *Törlés* funkciót kiválasztva. Nem törölhető orvosra történő kísérlet esetén hibaüzenet jelenik meg (14. ábra)



14. ábra Nem törölhető orvos hibalehetőség

A következő menüpont a fő ablakon az orvosi vizitek nyilvántartása, az *Orvosi rendelések* menü. Erre kattintva a felhasználó a már megismert táblázatos nézetű kis ablakot fogja kapni, megjelenítve minden eddigi Medi-2020 által lebonyolított vizitről. A táblázat kereshető, menthető, lapozható és szintén működik a hozzáadás, törlés, módosítás funkció.

Hozzáadáshoz bárhova kattintva egyszer, jobb egérklikkel elő kell hívni a párbeszéd ablakot és a *Hozzáadás* menüpontot kiválasztani. Egy szokásos kisablakos űrlap jelenik meg és kitölthetők a vizit adatai: már regisztrált páciensnek lehet rendelési időt rögzíteni egy legördülő listából, meglévő orvost lehet kiválasztani szolgáltatás nyújtáshoz egy legördülő listából, valamint a szolgáltatások közül választani – szintén legördülő listából. A rendelés ideje egy naptárból választható ki. (15. ábra)

Orvosi rendelés

Beteg: Balogh Károly (228-003-112)

Orvos: Dr. Horváthné Vári Beáta (gyermekgyógyászat)

Szolgáltatás: kontroll

Időpont: 2020. augusztus 29., szombat

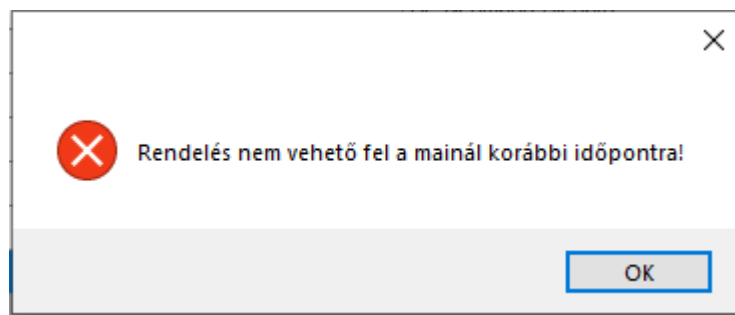
OK Mégse

2020. augusztus						
H	K	Sze	Cs	P	Szo	V
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Ma: 2020. 08. 29.

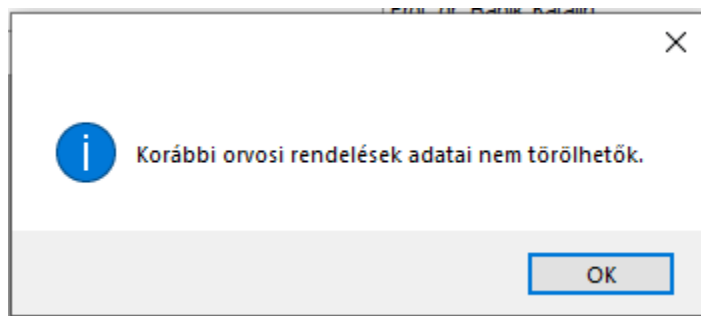
15. ábra Vizit rögzítési űrlap

Hibalehetőség az időben keletkezhet, mindig a mai naptól számítva korábbi időpontra nem vehető fel orvosi rendelés (a múltba). Az ilyen kísérletre való törekvés esetén hibaüzenet ablak jelenik meg a képernyőn. A sikeres rendelés felvételt a kis ablakon az OK gombbal, a nagyobb ablakon a Mentés gombbal hajtható végre. (16. ábra)



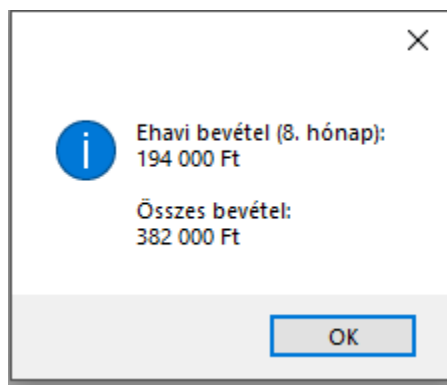
16. ábra Korábbi időpontra történő rendelés felvétel

Módosítás esetén szintén figyelni kell az időbeliségre. A mai dátumnál korábbi időpontra nem módosítható már felvett vizit, illetve a mai dátumnál korábbi vizit nem módosítható, illetőleg nem is törölhető! Azonban a jövőben felvett időpontokra vizitek módosíthatók és törölhetők értelemszerűen. Módosításkor a párbeszéd ablakon a *Módosítás* menü, törlés esetén a vizit kijelölése vizit után a párbeszéd ablakon *Törlés* menüvel történhet. OK gomb és Mentés gomb minden esetben a folyamat végleges lezárása. Szintén felugró hibaüzenet ablakok jelenítik meg az adott hibára futó ellenőrzés végeredményét. (17. ábra)



17. ábra Nem törölhető vizit hibalehetőség

Utolsó menüpont a fő ablak eszköztárában a *Pénzügyek* menüpont, ahol egy felugró információs ablak közli a Medi-2020 aktuális havi és összes eddigi bevételének összegét forintban. (18. ábra)



18. ábra Bevétel ablak

Jelenleg nincsen lehetőség új szakterület, vagy szolgáltatás felvitelére, igény esetén ilyen funkciók is készülhetnek az asztali alkalmazáshoz az eljövendőkből. Későbbiekben kifejtésre kerül a pénzügyek táblázatos formában történő kiolvasásának lehetősége.

1.2 A WEBALKALMAZÁS DOKUMENTÁCIÓJA

A Medi-2020 weboldala a cég reklám- illetve felhasználók számára nyújtott szolgáltatások felülete.

A weboldal használata végtelenül egyszerű, egy bemutatkozó oldal látható az elindításkor, amely a Medi-2020 kórházról mutat fényképeket, illetve pár szóban az igénybe vehető szolgáltatásokról ír. A kórház elérhetőségei a weboldal alján látható láblécen olvashatók. (19. ábra)



19. ábra Webalkalmazás nyitóoldala

A fő oldalon található navigációs sávval lehetséges menüpontok között böngészni. A navigációs sávon látható szöveges mezők segítségével lehet az alkalmazásba belépni és az általa nyújtott szolgáltatásokat használni. Amennyiben még nincs regisztrációja és nem ügyfele a Medi-2020-nak, úgy a *Regisztráció* menüpontra kattintva van lehetőség regisztrálni a magánkórház ügyfelei közé.

Ebben az esetben egy regisztrációs űrlap jelenik meg a weboldalon, amelyet értelemszerűen kell kitölteni. (20. ábra)

Regisztrációs űrlap

Vezetéknév
Keresztnév
TAJ szám
Jelszó
Írányítószám
Település
Lakcím
Telefonszám
E-mail cím
Regisztráció

Adatainak megadásával elfogadja az Adatvédelmi Irányelveket

20. ábra Regisztrációs űrlap

Amennyiben valamely mező kitöltése nem egyértelmű, úgy a Regisztráció gombra kattintva egy ellenőrzés fut le, amely az esetleg rosszul bevitt adatokra hívja fel a figyelmet. Olyan hibalehetőségek fordulhatnak elő, mint kisbetűs vezetéknév- vagy keresztnév, településnév, rossz e-mail cím formátum, nem négy karakterből álló és betűt tartalmazó irányítószám, vagy nem hármassával, kötőjellel elválasztott kilenc karakternyi a megadott TAJ szám. További hiba, amennyiben már létező – feltehetően más TAJ számát adtuk meg a regisztráció során véletlenül. Hibalehetőség még, hogy nem lehet üresen szöveges mezőket elküldeni. Ezek esetében is a gomb után piros színnel kiírt hibaüzenet figyelmeztetik a regisztráló felhasználót. (21., 22. és 23. ábra)

Regisztrációs űrlap

Vezetéknév
Keresztnév
TAJ szám
Jelszó
Irányítószám
Település
Lakcím
Telefonszám
E-mail cím
Regisztráció

Adatainak megadásával elfogadja az Adatvédelmi Irányelveket

Üres mezők!

Tajszám nem 11 karakter vagy nem tartalmaz kötőjelet!

Irányítószám nem 4 karakter!

E-mail nem tartalmaz @ karaktert!

21. ábra Regisztrációs hibalehetőségek

taron@pelda.hu
Regisztráció

Adatainak megadásával elfogadja az Adatvédelmi Irányelveket

Vezetéknév vagy keresztnév nem lehet kisbetűs!

22. ábra Kisbetűs vezeté- vagy keresztnév hibalehetőség

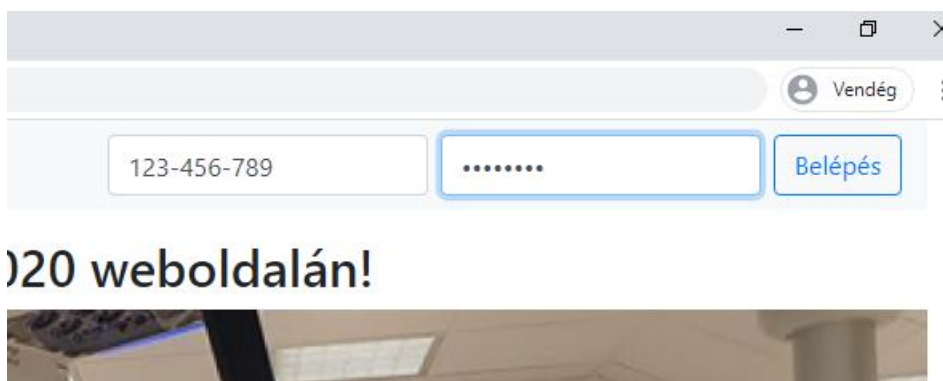
Regisztráció

Adatainak megadásával elfogadja az Adatvédelmi Irányelveket

Létező tajszám!

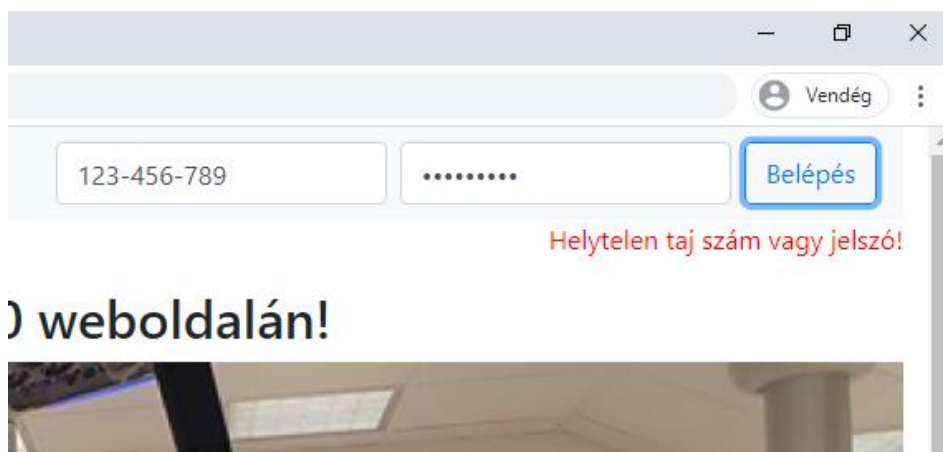
23. ábra Létező TAJ szám hibalehetőség

Amennyiben sikeres a regisztráció, úgy egy piros „Sikeres regisztráció” szöveg jelenik meg a Regisztráció gomb alatt. Az alkalmazásba való bejelentkezéshez a már korábban említett navigációs sávon található szöveges mezők segítségével, a TAJ-szám és a regisztráció során megadott jelszó begépelésével tudunk bejelentkezni. (24. ábra)



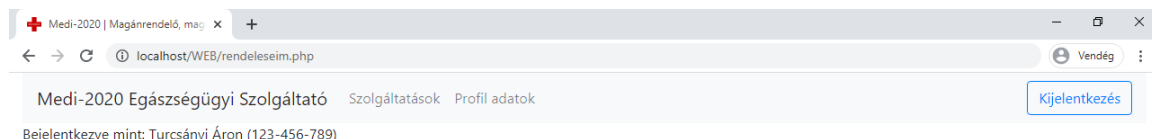
24. ábra Bejelentkezés funkció

Amennyiben rosszul kerülnek bevitelre a bejelentkezéshez szükséges adatok, úgy az alkalmazás piros színű hibaüzenettel jelzi a felhasználónak és nem lépteti be őt az alkalmazásba. (25. ábra)



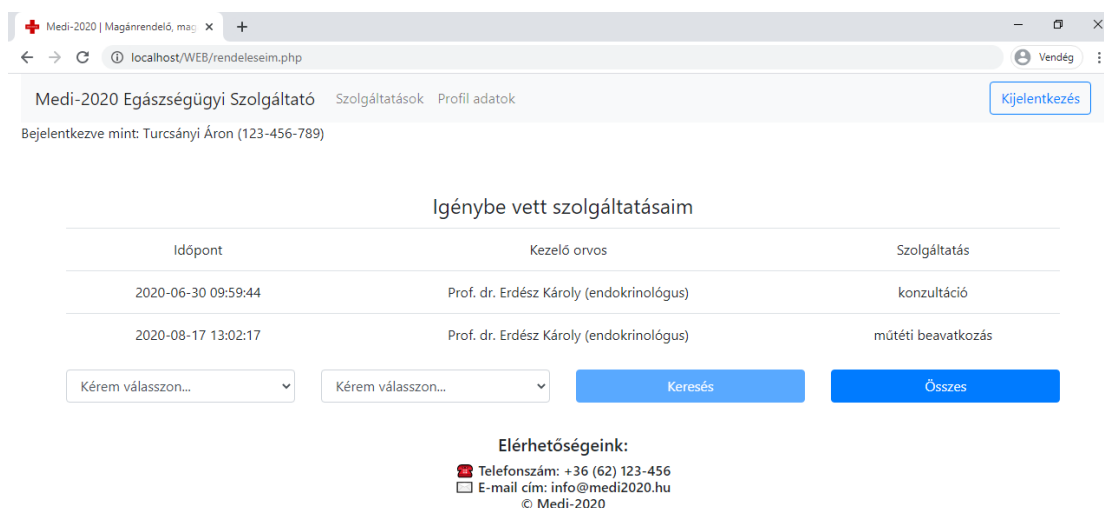
25. ábra Helytelen belépési adatok

A sikeres bejelentkezést követően a navigációs sáv jelzi a leendő ügyfél számára, hogy bejelentkezett állapotban van és a munkamenetét az alkalmazás nyilvántartja. Ez abban nyilvánul meg, hogy a navigációs sáv tartalma megváltozik a használható szolgáltatások menüpontjaira, a belépés gomb helyett kijelentkezés gomb jelenik meg, valamint a bal felső sarokban megjelenik a bejelentkezett felhasználó neve és mellette zárójelben a TAJ száma. (26. ábra)



26. ábra Bejelentkezett felhasználó munkamenete

Elsők között megnyílik a *Szolgáltatások* menüpont, ahol a páciens adott betegélettudját tudja nyomon követni: mikor, melyik orvosnál milyen szolgáltatást vett igénybe a Medi-2020 magánkórház keretein belül. Ez egy táblázatos megjelenítés a weboldalon, amelyen lehet keresést is végezni. A táblázat első oszlopában a szolgáltatás időpontja látható, a középső oszlopban a kezelő orvos neve és szakterülete, az utolsó oszloppedig a szolgáltatás nevét jeleníti meg. (27. ábra)



27. ábra igénybe vett szolgáltatások funkció

A táblázat alatt látható kettő legördülő lista, amellyel szűrhetők a vizitekből az orvosok neve és a szolgáltatás típusa. A mellette lévő keresés gombra kattintva indítható el az adott

paraméterekkel a keresés folyamata. Alapértelmezetten a Keresés gomb nem működik, amíg valamelyik legördülő lista nincsen kiválasztva, legalább az egyik kiválasztására kattinthatóvá válik a keresés gomb. Értelemszerűen így szűrhetünk csak az orvos nevére, csak a szolgáltatás típusára, illetve egyidejűleg mindkettő paraméterre. Ekkor a táblázatban a szűkebb keresztmetszetű, szűrt táblázat jelenik meg. (28., 29. és 30. ábra)

Időpont

2020-06-30 09:59:44	Prof. dr.
2020-08-17 13:02:17	Prof. dr.

Kérem válasszon...

Kérem válasszon...

Dr. Horváthné Vári Beáta (gyermekgyógyász)

Dr. Juhász Györgyi (reumatológus)

Dr. Kiss Péter (bőrgyógyász)

Dr. Kiss László (gasztroenterológus)

Dr. Kovács Béla (urológus)

Dr. Neumann Richárd (fül-orr-gégész)

Dr. Széll Miklós (tüdőgyógyász)

Prof. dr. Erdész Károly (endokrinológus)

Prof. dr. Privóczki Noémi (pszichiáter)

Prof. dr. Rábik Katalin (nőgyógyász)

Kérem válasszon...

Kérem válasszon...

Te
 E-

28. ábra Orvosra szűrés

Igénybe vett szolgáltatás

Időpont	Kezelő orvos
2020-06-30 09:59:44	Prof. dr. Erdész Károly (endokrin)
2020-08-17 13:02:17	Prof. dr. Erdész Károly (endokrin)

Kérem válasszon...

Kérem válasszon...

kontroll

konzultáció

műtéti beavatkozás

Kérem válasszon...

Kérem válasszon...

Összegeink:

hívás díja: +36 (62) 123-4567

☐ E-mail cím: info@medi2020.hu

© Medi-2020

29. ábra Szolgáltatásra szűrés

Medi-2020 Egészségügyi Szolgáltató Szolgáltatások Profil adatok [Kijelentkezés](#)

Bejelentkezve mint: Turcsányi Áron (123-456-789)

Igénybe vett szolgáltatásaim

Időpont	Kezelő orvos	Szolgáltatás
2020-08-17 13:02:17	Prof. dr. Erdész Károly (endokrinológus)	műtéti beavatkozás

Flérhetőségeink:

30. ábra Szűrt adatok táblázatban

Az Összes gombra kattintva visszaállítható a szűrt táblázat az összes igénybe vett szolgáltatásra. Másik funkció a *Profil adatok* menüpont, ahol későbbi adatváltozás során lehetősége van a felhasználónak ezeket módosítani a Medi-2020 felé. Ehhez egy űrlap áll rendelkezésre, amely igencsak hasonló a regisztrációs űrlaphoz. A kitöltés itt is értelemszerű, bármely személyes adat módosítható a TAJ számon kívül. Az űrlap alatt található kék, módosítás gombbal végezhető el az adatok módosítása. (31. ábra)

Profil adatok

123-456-789

.....

Turcsányi

Áron

6725

Szeged

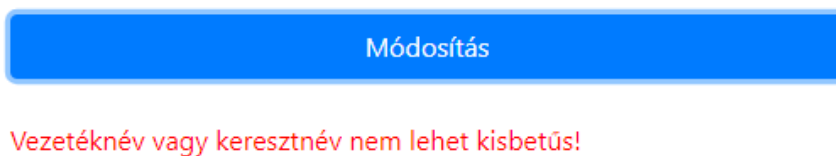
Példa utca 45.

turcsanyi@gmail.com

06202345783

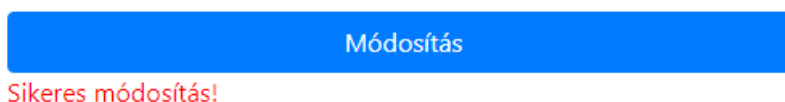
31. ábra Profil módosítása űrlap

A regisztrációs űrlaphoz hasonlóan a rosszul vagy üresen hagyott szöveges mezők esetén szintén piros hibaüzenetek jelzik a rosszul megadott adatok hibáit. (32. ábra)



32. ábra Módosítás esetén hibalehetőség

Sikeres módosítás esetén egy piros „Sikeres módosítás!” felirat nyugtázza a műveletet. (33. ábra)



33. ábra Sikeres módosítás visszajelzése

Az alkalmazásból kilépni a jobb felső sarokban található kilépés gombra kattintva lehetséges, amely rögtön a fő oldalra irányítja vissza a felhasználót.

2. FEJLESZTŐIDOKUMENTÁCIÓ

A Medi-2020 betegfelvételi és magánkórházi alkalmazásrendszer A rendszer funkcióit az alkalmazáshoz készült use-case diagram szemlélteti a legjobban, amelyet a Melléklet 1. számú ábrája szemléltet.

Az alkalmazásrendszer az alábbi szoftverek segítségével készült el:

- Az adatbázis: ApacheXAMPP programcsomagban foglalt, Oracle által gyártott MySQL egy fajtája, a MariaDB adatbáziskezelő-rendszerben készült el,
- Az asztali alkalmazás C# nyelven készült Visual Studio 2019 fejlesztőkörnyezetben, 4.7.2 verziójú .NET Framework platformon; az egységtesztet MSTestFramework segítségével készültek.
- A webalkalmazáshoz használt nyelvek: HTML, CSS, JavaScript (nagyreszt jQuery API) és PHP, NetBeans 8.2 fejlesztőkörnyezet segítségével

- A rendszertervek diagramjai elkészítéséhez a <https://www.diagrams.net> weboldalon elérhető rajzeszközök nyújtottak segítséget.

2.1 AZ ADATBÁZIS

Tanulmányaim során adatbázis-kezelés témakörben a relációs adatbázisokkal ismerkedtem meg, így nem volt kérdés, hogy a fejlesztendő rendszer alapját is egy ilyen típusú adatbázis fogja adni. A relációs adatbázisokon belül is a jelenleg az *Oracle* tulajdonában lévő *MySQL* adatbázis-kezelő rendszerrel dolgoztam a fejlesztés során.

Az adatbázis megtervezése alapvetően egy kreatív munkafolyamat volt számomra. Ezen munkafolyamat során a szokásos, standard eljárást alkalmaztam és az egyed-kapcsolat diagram elkészítését végeztem el első körben, miután a követelmények feltárásaként tudtam milyen adatbázis sémát szeretnék kialakítani. Egészségügyi nyilvántartói rendszerről lévén szó értelemszerűen el kellett tárolni a Medi-2020 orvosainak és pácienseinek az adatait, az igénybe vehető szolgáltatásokat és rendeléseket továbbá az orvosok szakterületeinek az adatait.

Az egyed-kapcsolat diagram készítése során kell kialakítani a logikai adatbázist: a tárolandó adatokat csoportosítani kell attribútumok formájában, amelyek egy egészként majd egyedeket alkotnak – leírjuk azokat, amely dolgokról adatokat szeretnénk tárolni; valamint ezen egyedek közötti kapcsolatokat kell feltérképezni és a diagrammon jelölni őket. Minden egyednél ki kell jelölni egy elsődleges kulcs attribútumot, hogy a későbbiekben minden letárolt egyed egyedileg és egyértelműen azonosítható legyen.

Végezetül a normalizálást is elvégezhetjük a logikai adatbázis tervezése során. Ez egy rendkívül fontos eljárás a tervezési szakaszban, hogy a későbbiekben a relációs adatbázis konzisztenciáját megőrizve redundanciáktól mentesen tarthassam. Az általános ajánlásokat figyelembe véve a harmadik normálformáig végeztem el az egyedek és kapcsolatok rendszerezését. A harmadik normálforma szerinti normalizálás röviden annyit tesz, hogy minden attribútum a saját egyedéhez tartozik, valamint egyetlen egyedben sincs olyan attribútum, amely a kulcstól funkcionálisan vagy tranzitívan függene. Minden adat a megfelelő egyedhez van kapcsolva, amelyhez leginkább tartozik. Ezáltal elkerülhetők a hivatkozási integritás megsértésére irányuló törlési, beszúrási és módosítási anomáliák.

Az egyed-kapcsolat diagram elkészülte után azokat átírtam relációsémává, amellyel el is kezdődött a fizikai, *MySQL* adatbázis implementációja. Az adatbázis neve *medi_2020*, utf-8 magyar karakterkódolás szerinti. Az alábbiakban a relációs adatbázis tábláinak és kapcsolatainak dokumentációját írom le:

- *doctor* tábla: a Medi-2020-nál dolgozó orvosok adatai.
Id, automatikusan növekvő, sorszámozott azonosító, ***elsődleges kulcs*** (11 hosszú egész szám)
Vezeteknev, (50 hosszú szöveg)
Keresztnev, (50 hosszú szöveg)
SzakteruletId, orvoshoz tartozó szakterület, amely a *specialization* táblához idegenkulcs megkötésben áll.
- *patient* tábla: a Medi-2020 ügyfelének adatai.
Taj, ***elsődleges kulcs*** (9 hosszú egész szám)
Jelszo, (10 hosszú szöveg)
Vezeteknev, (50 hosszú szöveg)
Keresztnev, (50 hosszú szöveg)
Iranyitoszam, (4 hosszú egész szám)
Telepules, (50 hosszú szöveg)
Lakcim, (50 hosszú szöveg)
Email, (75 hosszú szöveg)
Telefon, (11 hosszú szöveg)
- *service* tábla: az orvosi rendelés során igénybe vehető szolgáltatások adatai.
Id, automatikusan növekvő, sorszámozott azonosító, ***elsődleges kulcs*** (11 hosszú egész szám)
Nev, (50 hosszú szöveg)
Egysegar, (11 hosszú egész szám)
- *specialization*: Orvosok szakterületeinek adatai.
Id, automatikusan növekvő, sorszámozott azonosító, ***elsődleges kulcs*** (11 hosszú egész szám)
Nev, (25 hosszú szöveg)

- *visit*: Orvosi rendelések adatai: mikor, melyik páciens, melyik orvosnál, milyen szolgáltatást vett igénybe.

Id, automatikusan növekvő, sorszámozott azonosító, ***elsődleges kulcs*** (11 hosszú egész szám)

Idopont, (időbélyegző, *current_timestamp* függvény alapértelmezett értékkel)

BetegTaj, (11 hosszú egész szám) beteg, aki részt vett a rendelésen, idegenkulcs megkötésben áll a *patient* tábla *Taj* kulcsmezőjével

OrvosId, (50 hosszú szöveg) orvos, aki rendelt, idegenkulcs megkötésben áll a *doctor* tábla *Id* kulcsmezőjével

SzolgaltatasId, (11 hosszú egész szám) szolgáltatás, amelyet a beteg az adott orvosnál vett igénybe, idegenkulcs megkötésben áll a *service* tábla *Id* kulcsmezőjével

2.2 AZ ASZTALI ALKALMAZÁS DOKUMENTÁCIÓJA

A Medi-2020 rendszer asztali alkalmazás implementációja C# nyelven készült 4.7.2 verziószámú .NET platformon. A grafikus felület kialakítása *Windows Forms* grafikus fejlesztői eszköztár segítségével történt.

A szoftver többretegű, a *Microsoft* által kifejezetten ehhez a grafikus fejlesztői eszköztárhoz kifejlesztett *Model-View-Presenter* architektúra sémáján alapul. Jól elkülönül benne minden felelősségi kör, külön réteg végzi a grafikus vezérlők megjelenítését (View), az alkalmazás interakciós logikáját (Presenter), az adatbázis egyedeket pedig a Model réteg képezi le.

A szoftver felhasználja a repository tervezési mintát, amely megvalósított réteg pedig az üzleti logikáért és az adattárolásért felelős. Ezek a szoftverkomponensek a saját, konzekvencia szerint elnevezett mappájukban találhatók. Az alkalmazás teljes UML diagramját a projekt gyökérkönyvtárában található *UML.cd* állomány tartalmazza, amelyet a Visual Studio fejlesztőkörnyezet *ClassDesigner* kiegészítője automatizált módon generál ki.

A grafikus felületet a View mappa tartalmazza. Ezen réteg az adatok megjelenítéséért felelős a felhasználó felé az adatbázisból. A *Windows Forms* jellegéből adódóan az interakciós logika a grafikus ablakok code-behind-jában is fellelhetők események formájában, azonban a dinamikus adatokat a következő réteg az, ami szolgáltatja a View felé – interfészeken keresztül.

Ezek az interfészek a *ViewInterfaces* mappában találhatók. Az interfészek a grafikus ablakokat írják le absztrakt módon, a Presenter azon keresztül kommunikál csak a View-val. A Presenter réteg az, aki minimális interakciós logikát és dinamikus adatok kötését végzi el az előbb említett interfészeken keresztül a grafikus felületnek.

A Presenter közvetlenül a Repository réteggel tart kapcsolatot, aki az üzleti logikát és az adattárolást végzi el. Fontos megjegyezni, hogy a Repository egy programtervezési minta, amelynek a legfontosabb szerepe az adattárolás leválasztásán túl, hogy ha egy projektben alkalmazunk perzisztens keretrendszert, akkor az az adatokat magába rejtse a többi modul/réteg elől, így csökkentve a függőséget egymás között. Így a későbbiekben egy továbbfejlesztés során, ha a perzisztens keretrendszert le kell cserélni másikra, akkor a többi modul kódja nem sérül, mivel ők a repository osztályokról tudnak. Jelen esetben az EntityFramework 6-os verziója van alkalmazva, mint perzisztens keresztrendszer a projektben. Az EntityFramework objektum-relációs leképzést végez az adatbázison, amelyből kigenerálódnak az egyedek, mint modellek, valamint egy Context osztályban menti el őket DbSet adatszerkezetben. Ez biztosítja az alkalmazásnak az adatelérést az adatbázishoz. Értelemszerűen minden modellnek van repository osztálya, amely róla szolgáltat adatokat vagy engedélyez adatmanipuláló műveleteket a project számára az EntityFramework közreműködésével. Ezáltal minden repository osztály kapcsolatot tart a Context osztály egy példányával. Jelen projektben a Context osztály szerepét a Medi2020Context osztály látja el. A modelleket a *Model* mappa tárolja, ezek azok a kódok, amelyet az EntityFramework generált ki az adatbázis sémából, egy tábla egy modell osztály elven.

Az EntityFramework az adatbázis csatlakozás során az adatbázis elérésnek karakterláncát elmenti az *App.config* konfigurációs fájlba, XML formátumban. Az EntityFramework a MySQL által .NET platformra kiadott *MySQL Connector*-ja valósítja meg a csatlakozást, mivel alapértelmezetten MS SQL szabványt preferál a keretrendszer. Ez mind könnyedén beállítható a Visual Studio ServerExplorer menüjében, ahol a fejlesztőkörnyezetben is menedzselhető az adatbázis kapcsolat.

Egyéb képi erőforrásokat a *Resources* mappa tartalmaz. A projekt részét képezi továbbá egy egységteszt projekt is, amely a *Microsoft* saját egységtesztelő keretrendszerével - *MSTestFramework* – valósítja meg a tesztelhetőséget. Az egységtesztetek adatok ellenőrzését megvalósító függvényt tesztelnek.

Az alkalmazás futtatásához szükséges rendszerkövetelmények:

- minimum Windows 7 operációs rendszer,
- minimum 4 GB RAM,
- minimum két rendszermaggal ellátott processzor
- 7-es vagy annál nagyobb verziószámú .NET platform telepítése

szükséges

2.3 WEBALKALMAZÁS DOKUMENTÁCIÓJA

A webalkalmazás kliensoldalán a grafikus felület vázát a HTML nyelv írja le, annak formázását a CSS nyelv. A HTML DOM struktúra megváltoztatása pedig JavaScript nyelven történik, jQuery keretrendszer felhasználásával. A HTML felület továbbá felhasználja a Bootstrap 4 CSS keretrendszert, amely az oldal reszponzivitását adja. A szerveroldal PHP nyelven került leprogramozásra. Ezenfelül a webalkalmazás az adatbázis adatait aszinkron technikával, úgynevezett AJAX technológiával olvassa be és jeleníti meg az alkalmazásban.

A mappaszerkezet kialakítása a következő: a statikus HTML tartalmat a *html* mappa tartalmazza, úgymint navigációs sávok és a HTML oldalak fejrészeit. Az *img* mappa az oldalon használt képi erőforrásokat tartalmazza. A *config* mappa az adatbázishoz csatlakozó szerveroldali kódot tartalmazza a kapcsolatléíróval inicializálva. A *css* mappa a formázáshoz szükséges stíluslapot és a Bootstrap stíluslapot tartalmazza letöltve, *bootstrap.min.css*.

A *js* mappa a kliensoldali JavaScript (jQuery-ben megírt) nyelvű kódjait foglalja magába. A mappa tartalma továbbá a *jquery-3.4.1.min.js*, amely maga a jQuery, valamint a *bootstrap.min.js* és a *popper.js*. A saját JavaScript kód minden szerveroldali adatbázis lekérdezést tartalmaz, amely erőforrásként szolgál a *main.js* kódjainak, amely a kliensoldali aszinkron hívást végrehajtja és annak megfelelően valamely HTML kimenetet generál.

Mint minden webalkalmazás esetében, itt is az *index.php* a weblapindító weboldal, a többi weboldal *.php* kiterjesztésben a projekt gyökérkönyvtárban foglal helyet.

A felhasználói munkamenet szerveroldalon történik beállításra, SESSION szuperglobál tömb segítségével. A legtöbb adat dinamikusan, aszinkron jön létre AJAX technológia segítségével, amelyben a jQuery volt nagy segítségére.

3. FELHASZNÁLT KERETRENDSZEREK

A zárodolgozatban felhasznált keretrendszerek, API-k:

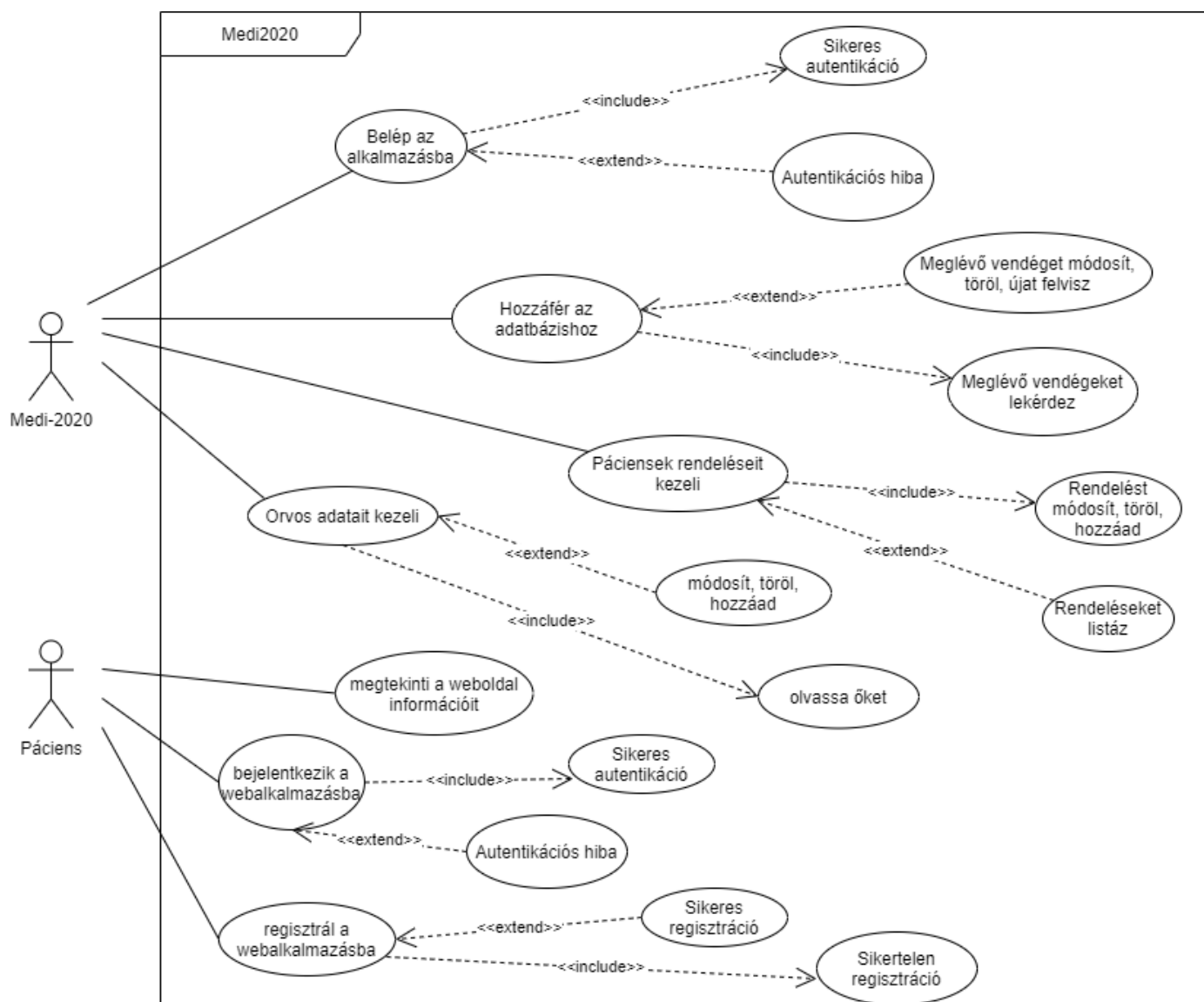
Bootstrap 4.5.1: <https://getbootstrap.com/>

jQuery 3.4.1: <https://jquery.com/download/>

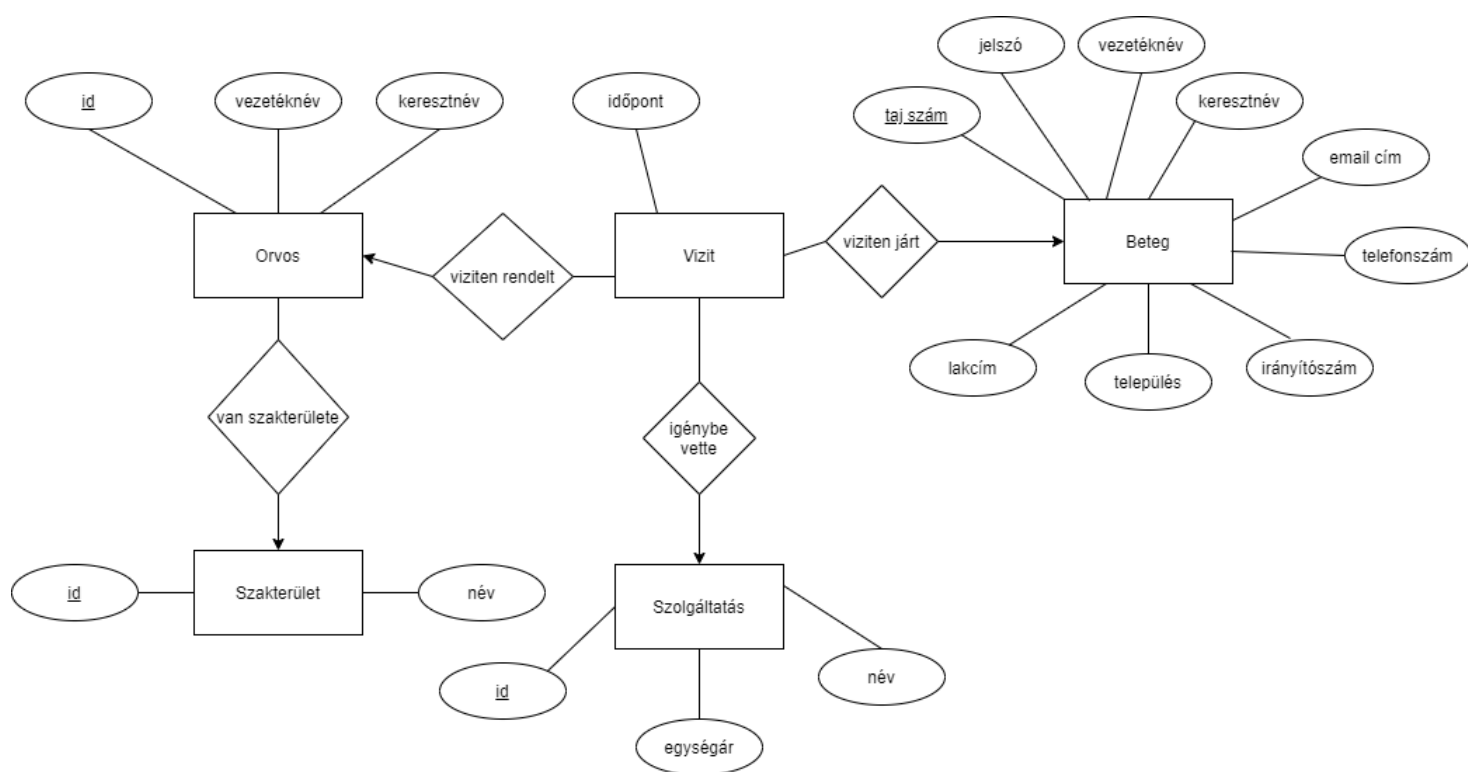
MySQL Connector 8.0.21: <https://dev.mysql.com/downloads/connector/net/>

EntityFramework 6.4.4: <https://www.nuget.org/packages/EntityFramework/>

4. MELLÉKLET



34. ábra A rendszer use-case diagramja



35. ábra Az adatbázis egyed-kapcsolat diagramja

KÖSZÖNETNYILVÁNÍTÁS

Köszönetet szeretnék mondani a Vasvári Pál Gazdasági és Informatikai Szakgimnáziumában dolgozó tanároknak, akik a tanulásomat segítették a két esztendő folyamán, név szerint Bálint Róbert Tanár Úrnak és Boros Bence Tanár Úrnak, akik a záródolgozatomat értékelték. Külön köszönetet szeretnék mondani első évestanáromnak Gyuris Csaba Tanár Úrnak!

Turcsányi Áron Bálint

Szeged, 2020-08-30