



Eötvös Loránd Tudományegyetem
Informatikai Kar
Információs Rendszerek Tanszék

Egészségügyi nyilvántartó

Témavezető:

Szalai-Gindl János Márk
adjunktus

Szerző:

Habzda Fruzsina Mária
Programtervező informatikus BSc.

Budapest, 2020

Tartalom

1.	Bevezetés	4
1.1.	Témaválasztás	4
1.2.	Témaválasztás indoklása	4
1.3.	Megvalósítás.....	4
2.	Felhasználói dokumentáció.....	6
2.1.	Rendszerkövetelmények	6
2.1.1.	Szerveroldali követelmények	6
2.1.2.	Kliensoldali követelmények	7
2.1.3.	Végfelhasználó követelményei	7
2.2.	Telepítés	8
2.2.1.	Konfiguráció	8
2.2.2.	Függőségek telepítése	9
2.2.3.	Alkalmazás fordítása.....	10
2.2.4.	Alkalmazás telepítése és futtatása.....	10
2.3.	Felhasználói felület bemutatása.....	11
2.3.1.	Alkalmazás elérése	11
2.3.2.	Alkalmazás felépítése	11
2.3.3.	Rendszerüzenetek	25
3.	Fejlesztői dokumentáció	28
3.1.	Tervek.....	28
3.1.1.	Fő funkcionális körök (szerepekre bontva):.....	28
3.1.2.	Use Case diagramok	29
3.1.3.	Nem funkcionális követelmények	32
3.1.4.	User Story-k.....	34
3.2.	Spring Boot program.....	46

3.2.1.	Fejlesztői környezet.....	46
3.2.2.	Package-ek.....	47
3.2.3.	Osztályok.....	47
3.2.4.	Kivételek	58
3.2.5.	Függőségek	58
3.3.	Adatbázis.....	61
3.4.	Kliensoldal	63
3.4.1.	Fejlesztői környezet.....	63
3.4.2.	Projekt felépítése	63
3.4.3.	Domain mappa.....	64
3.4.4.	Services mappa	65
3.4.5.	Komponensek	65
3.4.6.	Függőségek	67
4.	Tesztelés.....	69
4.1.	Junit tesztelés	69
4.2.	Manuális tesztelés	71
5.	Továbbfejlesztési lehetőségek	79
6.	Ábrajegyzék.....	82
7.	Összegzés	81
8.	Források	84
8.1.	Felhasznált képek	86
9.	Mellékletek.....	87

1. Bevezetés

1.1. Témaválasztás

Egészségügyi nyilvántartót hoztam létre, mely képes tárolni páciensenként az egészségügyi papírokat (recepteket, beutalókat és leleteket). Ezen dokumentumokat az egészségügyi dolgozók képesek feltölteni a rendszerbe. A dolgozóknak lehetőségük van a felhasználók között keresni is, és ezáltal hozzá tudnak férni a már feltöltött dokumentumaikhoz, továbbá itt tudnak feltölteni újakat is. A rendszer lehetővé teszi, hogy a páciensek háziorvost válthassanak, lakhelyük szerint. A dolgozó megkapja a kérelmet, majd miután jóváhagyja, értesítést küldünk a felhasználónak, amennyiben rendelkezik email címmel. A dolgozók megadhatják, hogy a regisztrált intézmények közül hol rendelnek, ezek az intézmények pedig megtalálhatóak egy térképen, a gyógyszertárakkal együtt, ami biztosítja a könnyebb keresést a felhasználók számára.

1.2. Témaválasztás indoklása

Az egészségügynek szüksége van egy olyan integrált rendszerre, ahol a betegek összes adata elérhető egy helyen, annak érdekében, hogy a diagnózisok könnyebben felállíthatók legyenek. Pár évvel ezelőttig nem használt ilyen jellegű alkalmazást az egészségügy, és a betegeknek maguknak kellett hordaniuk az összes egészségügyi papírjukat az egyes vizsgálatokra. Úgy gondoltam, hogy sokat lehetne ezen korszerűsíteni, hogy ezzel mind a betegek, mind az orvosok dolgát megkönnyítsük.

1.3. Megvalósítás

A szerver Spring Boot-ban lett megírva, Java nyelven, több könyvtárat is használ, melyeket maven repositoryból szed le. A program RESTful api, tartalmaz security filtert, spring data-t használ, és az adatokat adatbázisba, a fájlokat egy külső tárhelybe menti, onnan kéri le. A Spring-nek köszönhetően könnyen átírható a program, és ennek következtében bármilyen sql alapú adatbázis használható hozzá. Jelenleg PostgreSQL-t használ, mely a Heroku-n fut. A szerver ezen kívül használ Amazon Web Service-t - pontosabban S3-at - a fájlok tárolásához.

A kliens oldal Javascript nyelven lett megírva, alapja a React könyvtár. Ezen kívül pedig több könyvtárat is használ. Valamely a megjelenítésben segít: Bootstrap,

FontAwesome, ReactCalendar, más a program részében: React-Router, és a térkép használatában: Leaflet.

A kliens és a szerver http kérésekkel kommunikál, az adatot többnyire JSON-ban küldi. A szerver többségében Entity-t küld a frontendnek módosítva a kérés szerint, de alkalmanként előfordul, hogy Data Transfer Object-etben szállítja az adatokat. A megvalósításról bővebben a „Fejlesztői dokumentáció” alatt olvashat a kedves Olvasó.

2. Felhasználói dokumentáció

A főoldalt mindenki látja, és térképen bárki kereshet gyógyszertárat, kórházat, illetve rendelőt, valamint azok adatlapjait is megtekintheti. A kliens oldali alkalmazás bizonyos funkcióinak használatához viszont az adminnak regisztrálnia kell a felhasználókat, illetve a dolgozókat. Ezen funkciók nagy része szerepkörhöz kötött. A felhasználó csak megtekinteni tudja a saját dokumentumait, míg egy dolgozó bárkinek feltölthet dokumentumot és le is töltheti őket.

A szerver alkalmazás feladata fogadni az adatokat a kliens alkalmazástól http protokollon, RESTful interfészen keresztül. A felhasználók regisztrálás után admini beavatkozás nélkül is módosíthatják bizonyos adataikat, és használhatják a számukra engedélyezett funkciókat. Ezeket a parancsokat HTTP kéréseken keresztül kommunikálja le a kliens a szervernek, ami végrehajtja a szükséges változtatásokat az adatbázisban.

2.1. Rendszerkövetelmények

2.1.1. Szerveroldali követelmények

A szerver oldali alkalmazás fejlesztése és tesztelése Windows 10 operációs rendszeren, Spring Tool Suite környezetben, Spring Boot keretrendszerben történt. Az adatbázis, amit használ, Herokun fut, a fájl tárolója pedig Amazon Web Service, és Amazon szerveren van. Ezeknek tehát nincsenek további rendszer követelményeik.

A futtatáshoz javasolt technológiák a következők:

- Hardware (minimum):
 - Processzor: 1 gigahertz
 - Memória: 2 gigabájt
 - Háttértár: 50 gigabájt (tartalmazza az operációs rendszer igényét is)
- Internetkapcsolat
- Operációs rendszer:
 - Windows 10 (JVM-nek köszönhetően bármilyen modern operációs rendszeren fut)
- IDE (Integrated Development Environment) – nem kötelező:
 - Spring Tool Suite 4 (>= 4.7.2)

- Eclipse (>= 2020-09)
- NetBeans (>= 12)
- SE JDK (Java Standard Edition Development Kit):
 - Java SE 1.8
- Java JRE
 - >=14
- Java verzió:
 - >= 1.8

2.1.2. Kliensoldali követelmények

A kliens oldali alkalmazás fejlesztése és tesztelése Windows 10 operációs rendszeren, Microsoft Visual Studio Code környezetben, Google Chrome 85-ös böngészőben történt. A kliens oldali programot le kell fordítani, majd egy tárhelyszolgáltatóhoz felrakni.

A függőségek letöltését, és a program fordítást elvégző géphez javasolt technológiák a következők:

- Hardware (minimum):
 - Processzor: 1 gigahertz
 - Memória: 2 gigabájt
 - Háttértár: 50 gigabájt (tartalmazza az operációs rendszer igényét is)
- Internetkapcsolat
- Operációs rendszer:
 - Windows 10 (de bármely más operációs rendszer is megfelel)
- IDE (Integrated Development Environment) – nem kötelező:
 - Visual Studio Code
 - bármely szövegszerkesztő

2.1.3. Végfelhasználó követelményei

A végfelhasználónak csak egy Javascript futtatására képes böngészővel kell rendelkeznie számítógépén.

A használathoz javasolt technológiák a következők:

- Kijelző felbontása:
 - 1920x1080
- Operációs rendszer:
 - bármely modern operációs rendszer megfelel
- Böngésző:
 - Google Chrome 85 <=
 - Mozilla Firefox 81 <=

2.2. Telepítés

2.2.1. Konfiguráció

Az alkalmazást szükséges konfigurálni mielőtt futtathatóvá tesszük. A Spring Boot programban az adatbázis, az Amazon fájlserver és az email elérést kell beállítani, továbbá a CORS policy miatt a kliens program elérését is meg kell adni. Kliens oldalon csak a szerverhez való hozzáférést kell beállítani, és a „homepage”-t.

Spring Boot program

Navigáljuk el az *egeszsegugyinyilvantarto\src\main\resources* mappába. Az ott szereplő *application.properties* fájlt nyissuk meg. (Bármilyen szövegszerkesztővel megnyitható.)

A *spring.datasource.url* után adja meg a saját adatbázisának elérési útját. A *spring.datasource.username* után adjuk meg az adatbázisnál használt felhasználónevét. A *spring.datasource.password* után adjuk meg az adatbázisnál használt jelszavát. (Abban az esetben, ha külső adatbázist használunk, ezeket az adatokat az adott szolgáltató generálhatja.)

Az Amazon S3 használatához is meg kell adni pár adatot, melyet az Amazon-tól kell elkérni. Az *amazonProperties.accessKey* után adjuk meg az elérési kulcsát az S3-nak. Az *amazonProperties.secretKey* után adjuk meg a titkos kulcsot. (Ezt a két adatot az Amazon egyszerre generálja, a titkos kulcsot csak egyszer engedi megnézni, ezért figyeljünk a kimásolásnál.) Az *amazonProperties.bucketName* után adjuk meg annak a létrehozott vödörnek (bucket) a nevét, melyet az adott S3-ban használni szeretnénk.

Az email funkció működéséhez szükséges egy külső email fiók. Miután ezt létrehoztuk, adjuk meg a *spring.mail.host* után a szolgáltató host-ját, majd a *spring.mail.port* után a portot. Adjuk meg a *spring.mail.username* után a felhasználónevet, és a *spring.mail.password* után a jelszót. (A jelszó nem feltétlenül az email fiókos jelszavunk, lehet, hogy a szolgáltatóval külön kell generáltatni egyet arra az esetre, ha az email szolgáltatást egy külső programból szeretnénk elérni.) Adjuk meg a használni kívánt email címet a *spring.mail.from.email* után. (Erről az email címről fogja a program küldeni a leveleket – ez lesz a feladó.)

Ahhoz, hogy a kliens el tudja érni a szerveret, külön engedély kell a szerver részéről biztonsági okok miatt. Ezt az *egeszsegugyinyilvantarto\src\main\java\hu\elte\szakdolgozat\egeszsegugyinyilvantarto\config\security* mappában lévő *SecurityConfig.java* megnyitása után (bármilyen szövegszerkesztő megfelelő) lehetséges. A *corsConfigurer* metódus alatt, az 50. sorban az *allowedOrigins*-ban írjuk be a kliens elérését (url-jét) idézőjelek között.

React program

Az *egeszsegugyinyilvantarto\src\main\webapp\reactjs\src\services* mappában található *request_service.js*-t nyissuk meg (egy szövegszerkesztő is elegendő). Itt az *api* nevű változó értékét írjuk át a szerver elérési útjára (url). Nyissuk meg a projekt gyökérkönyvtárában található *package.json*-t, majd miután lefoglaltunk egy domain címet az alkalmazásnak, ide írjuk bele a „homepage”: „{domain címe}”, –t (Például: „homepage”: „http://egészségügyinyilvantarto.hu”,).

2.2.2. Függőségek telepítése

Spring Boot program

Nyissunk meg egy konzolt! Navigáljunk az *egeszsegugyinyilvantarto* mappába, ahol az *mvnw* fájl és a *pom.xml* fájl található. Az *mvnw* segítségével fogjuk letölteni a programban használt függőségeket. A függőségek sorát a *pom.xml* tartalmazza. Írjuk be a konzolba azt, hogy: ***mvnw dependency:resolve*** vagy azt, hogy ***mvnw install***.

React program

A program függőségeinek letöltéséhez szükségünk lesz Node.js-re és npm-re. Ezeket töltsük le és telepítsük fel a gépre.¹

Nyissunk egy konzolt! Navigáljunk az *reactjs* mappába, ahol a *node_modules* mappa és a *package.json* fájl található. A függőségek a *node_modules*-ba fognak kerülni. A *package.json* tartalmazza a függőségek listáját. Futtassuk le az ***npm i*** parancsot. Ez ugyan figyelmen kívül fogja hagyni a függőségek verzió számait, és a legújabb verziókat tölt le, de ez véleményem szerint nem okoz problémát.

2.2.3. Alkalmazás fordítása

Spring Boot program

Nyissunk egy konzolt! Navigáljunk az *egeszsegugyinyilvantarto* mappába, ahol az *mvnw* fájl és a *pom.xml* fájl található. Az *mvnw* segítségével fogjuk becsomagolni a programot. Írjuk be a konzolba a következőt: ***mvnw package -Dmaven.test.skip=true***. Az így létrejött fájlok és mappák a *target* mappában megtalálhatók.

React program

Nyissunk egy terminált, navigáljunk a gyökérkönyvtárba, és futtassuk le az ***npm build*** parancsot, ez legenerál egy *build* elnevezésű könyvtárat, ami a program egy tömörített változata.

2.2.4. Alkalmazás telepítése és futtatása

Spring Boot program

Több választási lehetőség is van arra hogy hol szeretnénk futtatni a programot. Példaként a Tomcat szerveres megoldás²:

- Töltsük le, és csomagoljuk ki a Apache Tomcat-et³. Pontos leírás:
<https://www.baeldung.com/tomcat>

¹ <https://nodejs.org/en/download/>

² <https://www.baeldung.com/spring-boot-war-tomcat-deploy>

³ <http://mirror.23media.de/apache/tomcat/tomcat-8/v8.5.30/bin/>

- Másoljuk ki az *egeszsegugyinyilvantarto/target* mappából *egeszsegugyinyilvantarto-0.0.1-SNAPSHOT.war*-t és másoljuk be a *tomcat/webapps* mappába.
- Nyissunk egy konzolt, és navigáljunk el a *tomcat/bin* mappába és futtassuk le ezt a parancsot:
 - ***catalina.bat run*** (Windows környezetben)
 - ***catalina.sh run*** (Unix alapú környezetben)

React program⁴

Lépünk be a tárhelyszolgáltatónkra, navigáljunk a cPanel-re, majd a File Manager menüpontra. Az itt lévő mappaszerkezetben keressük meg a *public_html* mappát, és nyissuk meg.

Másoljuk be a projekt gyökérkönyvtárában szereplő *.htaccess* fájlt, és a *build* mappa tartalmát a *public_html* mappába.

2.3. Felhasználói felület bemutatása

2.3.1. Alkalmazás elérése

A host címét a böngésző fejlécébe írva, bármely olyan böngészővel elérhetővé válik a már telepített és futtatott alkalmazás, mely képes Javascript futtatására. A felhasználók bejelentkezés nélkül is használhatják a program bizonyos funkcióit, mint például a térképet. Megfelelő felhasználónévvel és jelszóval bejelentkezhetnek az admin által regisztrált felhasználók, és így további testreszabott funkciókat is elérhetnek.

2.3.2. Alkalmazás felépítése

Az alkalmazás egyes funkcióihoz és tartalmaihoz nem szükséges bejelentkezni a rendszerbe, ilyen például a főoldal, valamint a térkép megtekintése és használata. Más funkciók igénybevételéhez viszont elengedhetetlen a bejelentkezés. Ezek a funkciók szerepköröktől függenek, tehát mást tehet meg egy dolgozó, egy felhasználó és egy admin. Ezeket lejjebb mutatom be részletesebben.

Az alkalmazás tartalmaz egy fejléctet, egy lábléctet és a kettő közötti tartalmat.

⁴ <https://dev.to/crishanks/deploy-host-your-react-app-with-cpanel-in-under-5-minutes-4mf6>

2.3.2.1. Mindenki számára elérhető tartalmak

Egészségügyi nyilvántartó Intézmények térképe Bejelentkezés

Bejelentkezés

Taj szám / Felhasználónév

tájszám vagy felhasználónév

8 számjegyből álló szám a tájkártyán.

Jelszó

Jelszó

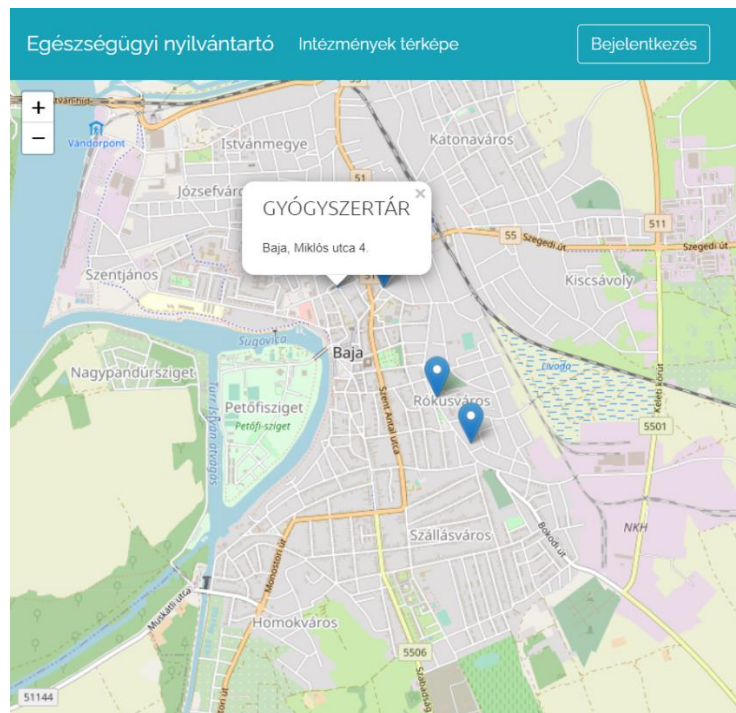
Belépés

2020 Egészségügyi nyilvántartó

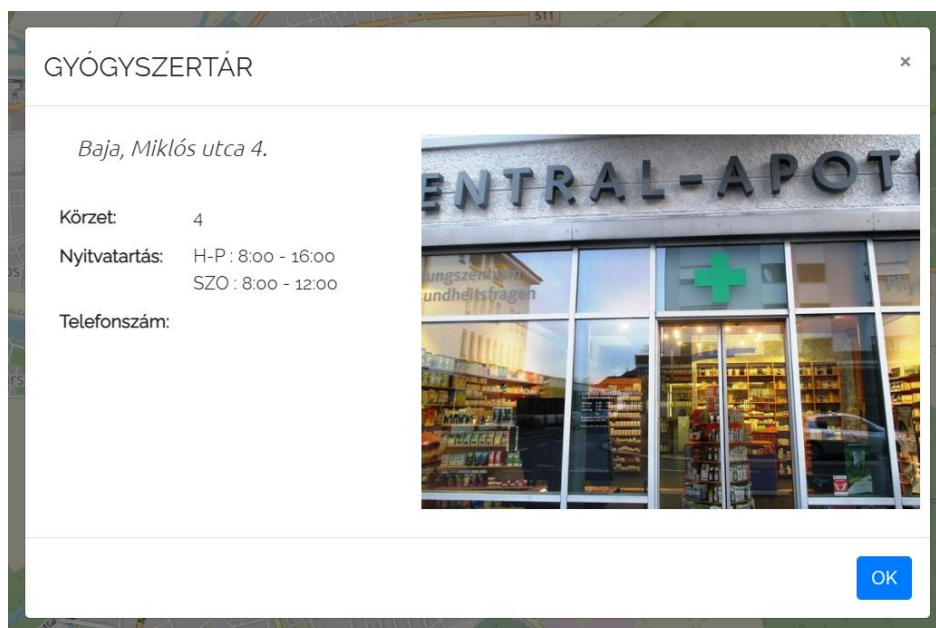
1. ábra Bejelentkezés felület

Az adminisztrátor, a felhasználó és a dolgozók is mind ezen a bejelentkező felületen tudnak belépni (ld. 1. ábra). Ha probléma van a bejelentkezési adatokkal, hibaüzenetet kapunk. (Felhasználónév/jelszó hiba, zárolva lett a profil)

A térkép (ld. 2. ábra) bárki számára elérhető. Itt megtalálhatók a rendszerbe felvitt kórházak, gyógyszerterek, és rendelők. A térkép kérheti a felhasználót, hogy engedélyezze számára a hely adatait, mely alapján azt a helyszínt tölti be, ahol a felhasználó tartózkodik. A térkép egyszerre csak annyi intézményt tölt be, amennyi a képernyőre ráfér. A térképen való navigálás közben kéri le a többi intézményt. Egy marker-re mutatva megjelennek az adott intézmény fontosabb adatai.



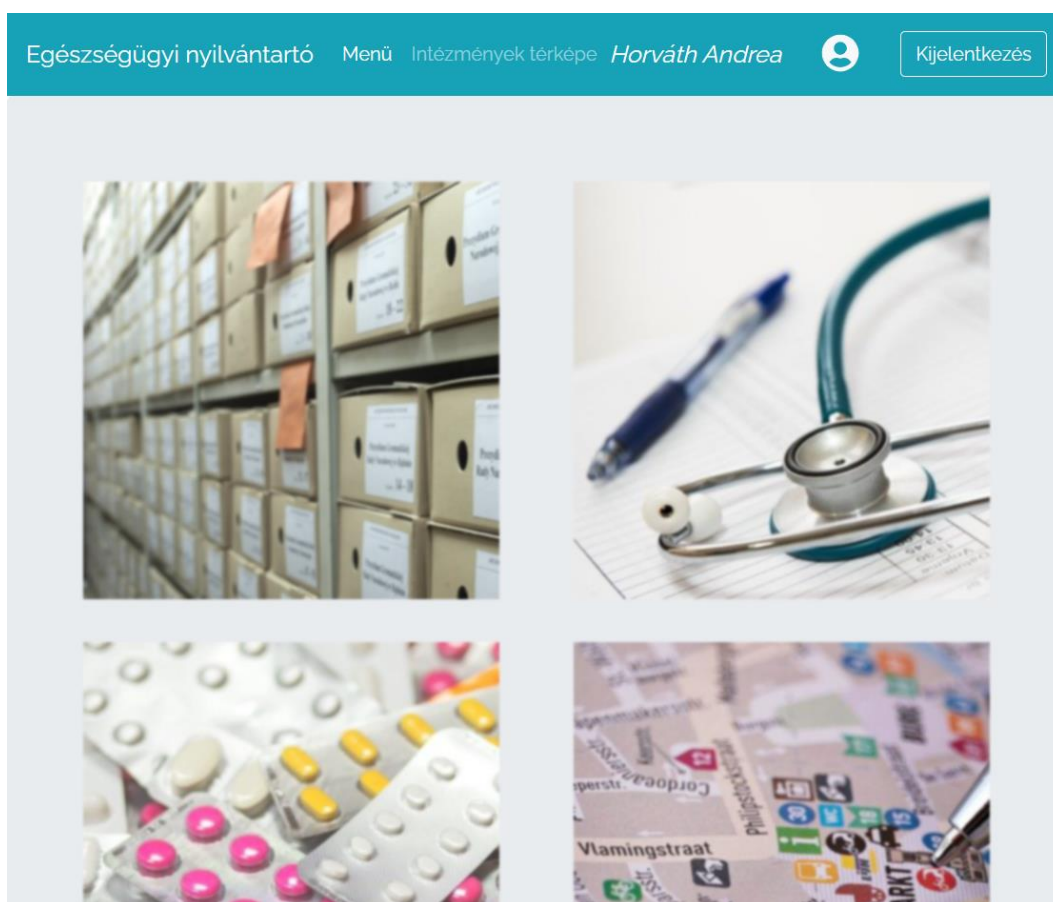
2. ábra Térkép



3. ábra Intézmény adatai

Az intézmények további adatai megtekinthetők, miután a felhasználó rákattintott az egyik marker-re (ld. 3. ábra). Az intézmény típusa, címe, körzetszáma, nyitvatartása, telefonszáma, hozzácsatolt kép, és egyéb hozzáfűzött megjegyzés is megtekinthetővé válik.

2.3.2.2. Csak páciensek által elérhető funkciók és tartalmak



4. ábra Felhasználó menüjének részlete

A menü lehetőséget biztosít az oldalak közti navigációra (ld. 4. ábra). A pácienseknek négy opciójuk van képekkel illusztrálva, ebben a sorrendben: Leletek, Beutalók, Receptek és a Térkép. Belépés után közvetlen a menü oldala jelenik meg. Ekkor a navigációs sáv is megváltozik, megjelenik rajta a bejelentkezett személy neve, és új gomb kerül rá, mellyel megnyitható a profil.

A receptek kiállításuk szerinti növekvő sorrendben jelennek meg az „Receptek” oldalon (ld. 5. ábra). A lejárt receptek világos szürkék, míg a még kiválthatók fehérek. A gyógyszer neve, javasolt szedése, a recept kiállítási dátuma, lejárat dátuma és kiállító orvos neve van feltüntetve az egyes recepteknél. A dokumentumok csökkenő és növekvő sorrendbe rendezhetők a „Gyógyszer”, illetve a „Kiállítás dátuma” szerint, a fejlécre való kattintással. (Első kattintással csökkenő - dátumoknál növekvő -, másodiknál növekvő - dátumoknál csökkenő - sorrend kapható.) Amennyiben nincs eltárolt dokumentum, ezt jelzi az oldal egy üzenettel.

Egészségügyi nyilvántartó
Menü
Intézmények térképe
Horváth Andrea
Kijelentkezés

Receptek

Tekintse meg feltöltött dokumentumait!

Gyógyszer ▼	Javasolt mennyiség	Kiállítás dátuma ▼	Lejárat dátum	Kiállító
D-vitamin	Napi 1 tabletta	2020.11.09.	2020.12.14.	Dr Kis Anikó
Rinospray	Napi 2x, 5 napig	2020.08.15.	2020.11.09.	Dr Kis Anikó
D-vitamin	Napi 1 tabletta	2020.08.02.	2020.08.15.	Dr Kis Anikó

2020 Egészségügyi nyilvántartó

5. ábra Felhasználó "Receptek" oldala

A beutalók kiállításuk szerinti csökkenő sorrendben jelennek meg az „Beutalók” oldalon (ld. 6. ábra). A felhasználó, a beutalóhoz tartozó dokumentumot a kiállító neve melletti gombbal letöltheti. Ekkor a böngészőben beállított letöltési mappájába lesz lementve a fájl, mely alap állapotban a „C” meghajtó „Letöltések” mappája.

A feltöltés dátuma, a beutaló kategóriája, annak oka és a kiállító neve van feltüntetve az egyes beutalóknál. A dokumentumok csökkenő, és növekvő sorrendbe rendezhetők a „Dátum”, illetve a „Kategória” szerint, a fejlécre való kattintással. (Első kattintással csökkenő - dátumoknál növekvő -, másodiknál növekvő - dátumoknál csökkenő - sorrend kapható.) Amennyiben nincs eltárolt dokumentum, ezt jelzi az oldal egy üzenettel.

A „Leletek” oldal (ld. 7. ábra) hasonlóképpen működik mint a „Beutalók” oldal, annyi eltéréssel, hogy itt a feltöltés dátuma, lelet kategóriája, hozzáfűzött megjegyzés és a kiállítójának neve szerepel.

Beutalók			
Tekintse meg feltöltött dokumentumait!			
Dátum ▼	Kategória ▼	Ok	Kiállító
2020.08.14.	vérvétel	koronavírus gyanú	Dr Kis Anikó

6. ábra Felhasználó "Beutalók" oldala

Leletek			
Tekintse meg feltöltött dokumentumait!			
Dátum ▼	Kategória ▼	Megjegyzés	Kiállító
2020.08.15.	szemvizsgálat	Új szemüveg javallott.	Dr Kis Anikó
2020.08.02.	vérvétel		Dr Kis Anikó

7. ábra Felhasználó "Leletek" oldala

Egészségügyi nyilvántartó
Menü
Intézmények térképe
Horváth Andrea
Kijelentkezés

Profil

Név:

Horváth Andrea

Taj szám:

12345678

Jelszó:

Születési dátum:

1970.02.03.

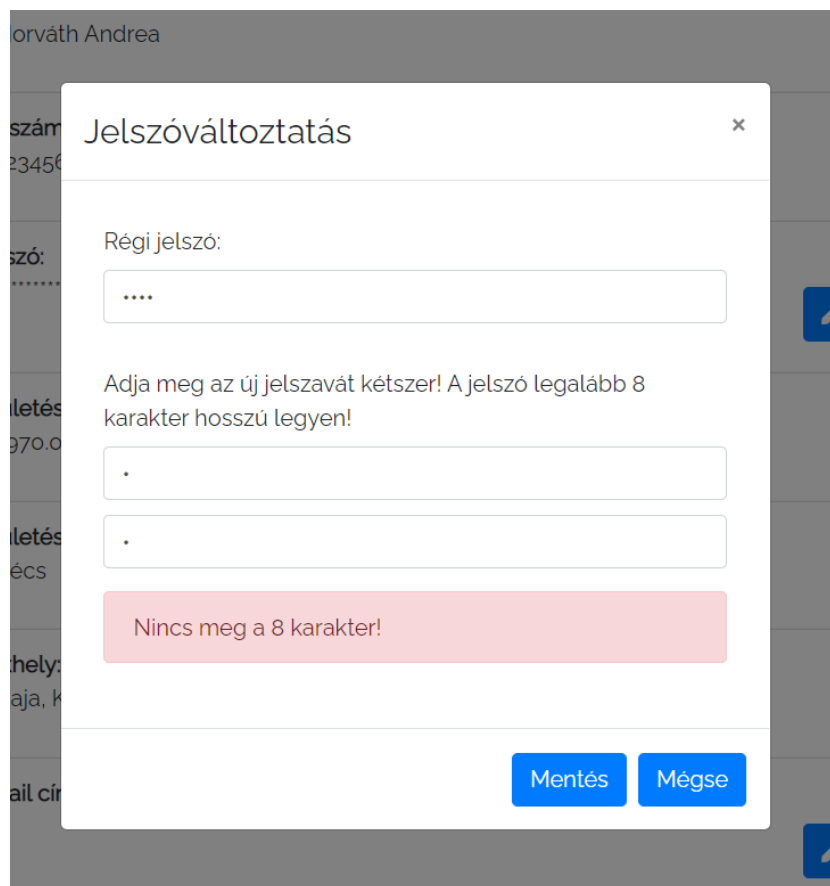
Születési hely:

Pécs

8. ábra Felhasználó "Profil" oldalának részlete

A profil a felhasználó adatait tartalmazza (ld. 8. ábra). A fejlécben szereplő név melletti ikonnal lehet megnyitni. Az adatok közül páron változtathat is.

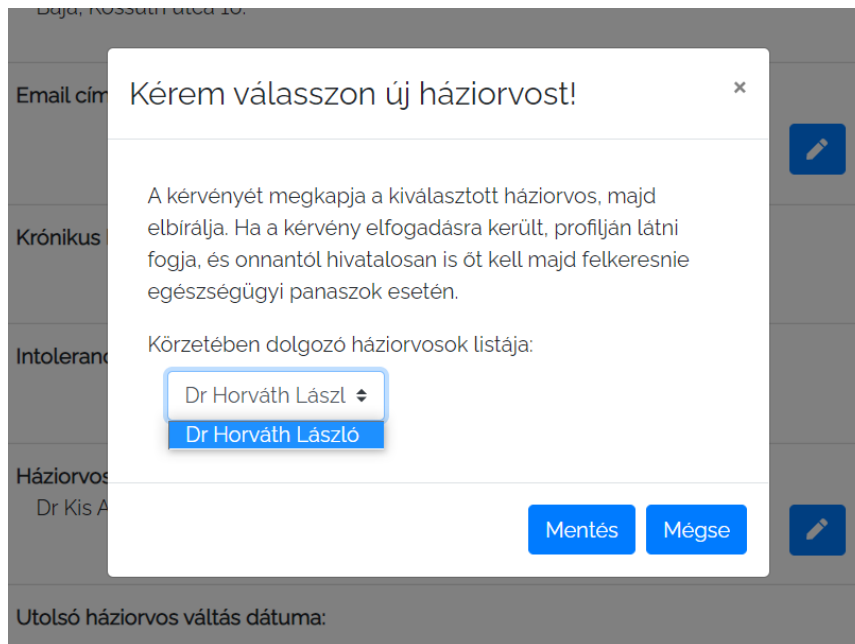
Jelszó változtatásnál meg kell adnia az eredeti jelszót és az újat kétszer. Az új jelszónak minimum 8 karakter hosszúnak kell lennie (ld. 9. ábra).



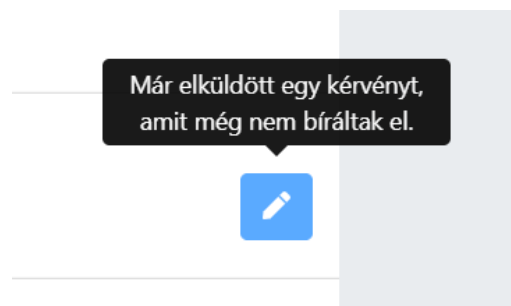
9. ábra Jelszó változtatás hibaüzenettel

Email cím megadása. Ez abban az esetben lényeges, ha a páciens szeretne értesítést kapni arról, ha elfogadják a házi orvos váltásra vonatkozó kérelmét.

Házi orvos változtatásnál a program figyelembe veszi, hogy melyik körzetben lakik a páciens, és azokat az orvosokat listázza, akik ebben a körzetben rendelnek a jelenlegi kivételével. Miután a páciens kiválasztja azt az orvost, akihez át szeretne jelentkezni, a program annak az orvosnak küld egy kérvényt (ld. 10. ábra). Egy házi orvos váltás után 3 évre lehet újra erre vonatkozó kérvényt leadni, és addig sem lehet váltani, amíg a páciensnek van leadott elbírátlan kérvénye (ld. 11. ábra).



10. ábra Új háziorvos választása

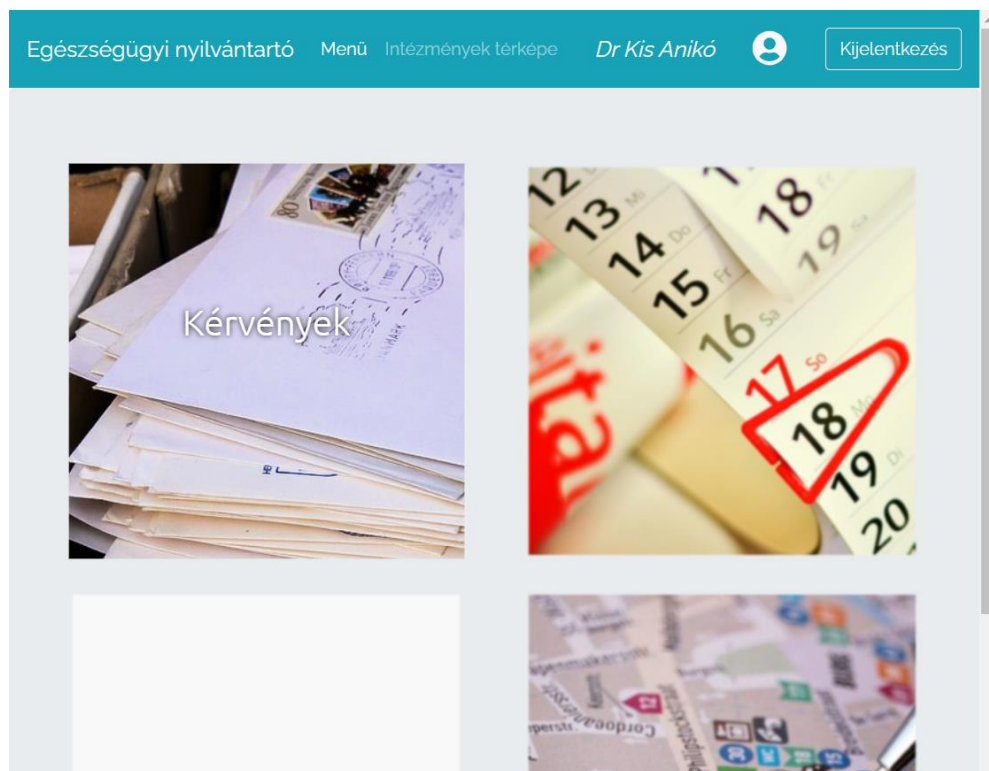


11. ábra Inaktív módosítás, háziorvos váltás után

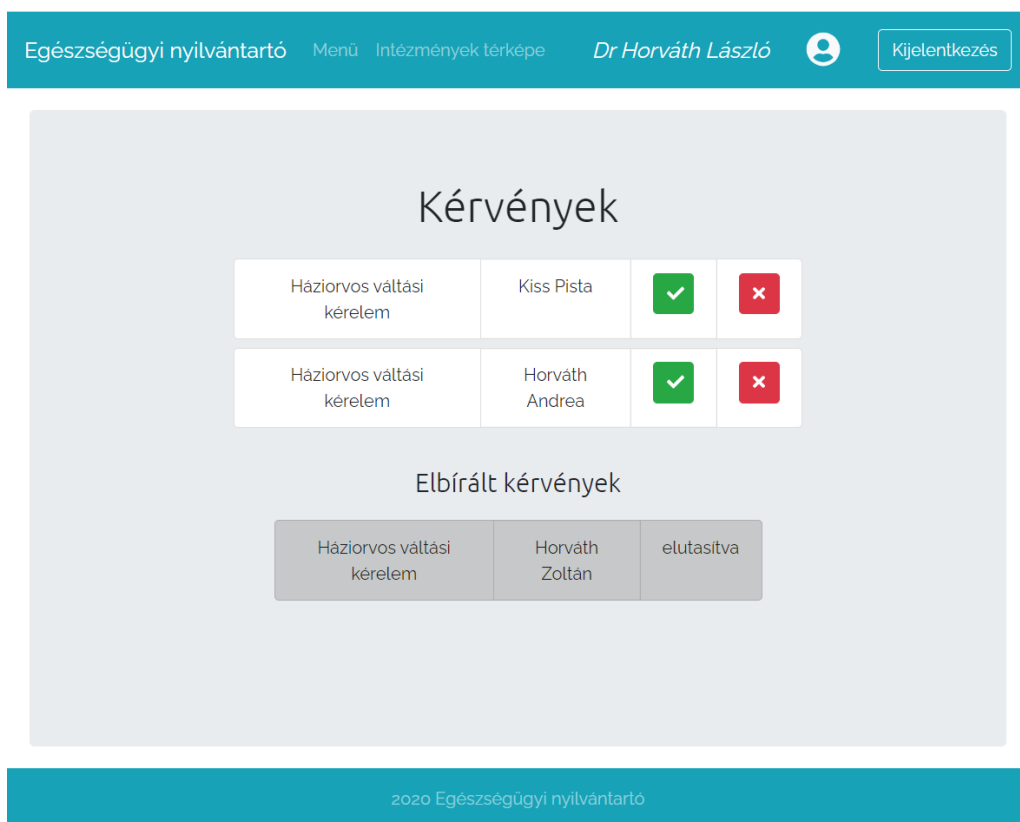
2.3.2.3. Csak dolgozók által elérhető funkciók és tartalmak

A dolgozói profil nagyon hasonló a betegek profiljához (ld. 8. árba), csak kevesebb adatot tartalmaz.

Dolgozói menüjének elemei, képekkel illusztrálva ebben a sorrendben: „Kérvények”, „Naptár”, „Páciens keresése”, „Térkép” (ld. 12. ábra).

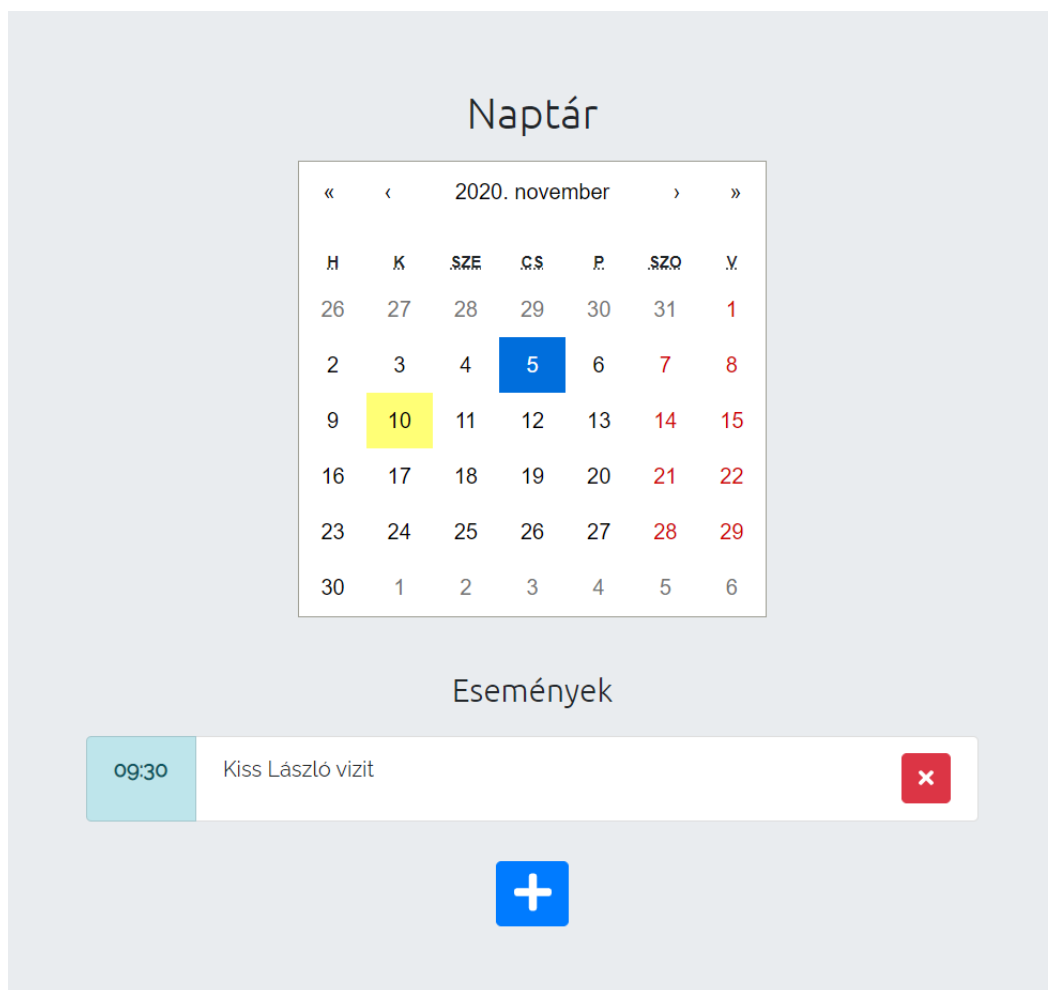


12. ábra Dolgozó menüjének részlete



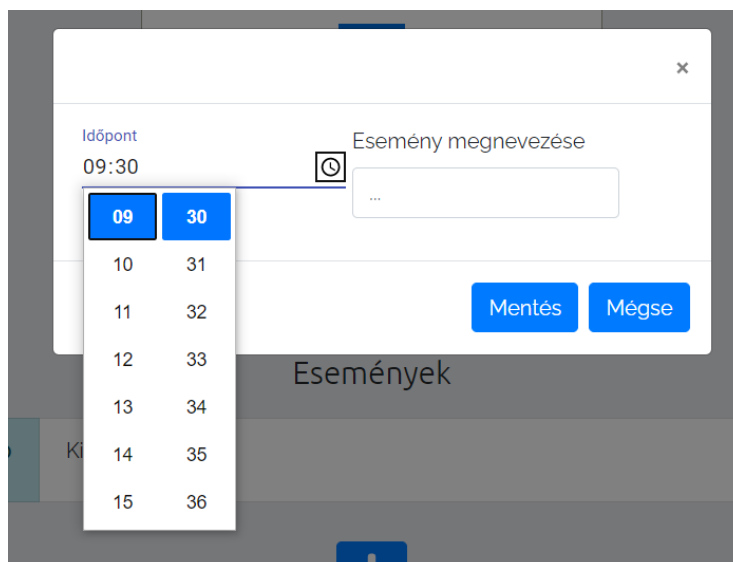
13. ábra "Kérvények" oldal

A Kérvények oldalon először az elbírátlan kérvények sorakoznak, majd az elbíráltak (ld. 13. ábra). Egy kérvény elutasításakor a páciens újra küldhet kérvényt, előtte viszont még nem, ezért javasolt a kérvényeket minél előbb elbírálni. Egy kérvény elfogadásakor a páciens a „házi orvos neve alá íródik” és a páciens adatlapján átíródik mind a házi orvos neve, mind a legutolsó házi orvos váltásának ideje. Amennyiben a páciens rendelkezik email címmel, értesítést is kap a változásról.

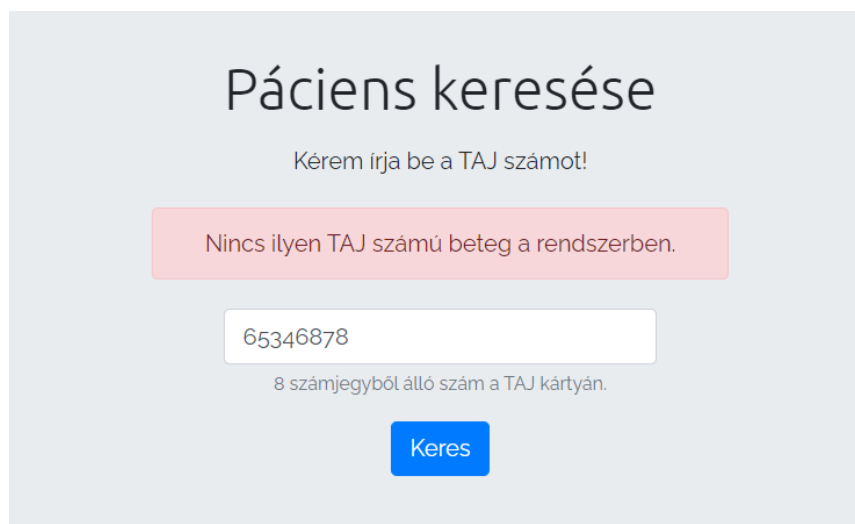


14. ábra "Naptár" oldal

A Naptár oldal események felvitelére szolgál. Az oldal megnyitásakor a naptár az adott napon áll, és annak eseményeit tölti be. A naptár egyes napjaira kattintva lehet megtekinteni az ahhoz tartozó eseményeket. Az eseményeket egyesével ki lehet törölni. Új esemény felvihető az adott napra a plusz gomb lenyomásával.



15. ábra Esemény felvitele

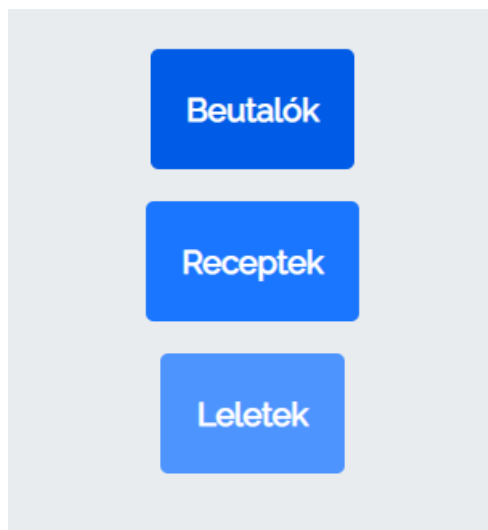


16. ábra "Páciens keresése" oldal, hibaüzenettel

A „Páciens keresése” oldalon az egészségügyi dolgozók kikereshetik a betegeket azok TAJ száma alapján. Ha az adatbázisban nincs olyan beteg, akinek a TAJ száma megfelelne a bevitt TAJ számnak, a program hibát ír ki (ld. 16. ábra). Amennyiben a bevitt adat nem felel meg TAJ számnak (nem számjegyekből álló 8 hosszúságú szöveg), akkor is hibát jelez a program.

A keresés gombra kattintva - amennyiben van ilyen beteg – a beteg adatlapjára navigál az oldal. Itt két adatot megváltoztathat az orvos: „Krónikus betegségek”, és „Intoleranciák” listáját. A páciens profiljához képest itt megjelenik három plusz gomb az

oldal alján, mégpedig: „Beutalók”, „Receptek” és „Leletek” felirattal (ld. 17. ábra). Ezek átnavigálnak a páciens adott dokumentumaihoz.



17. ábra Plusz gombok a páciens adatlapján

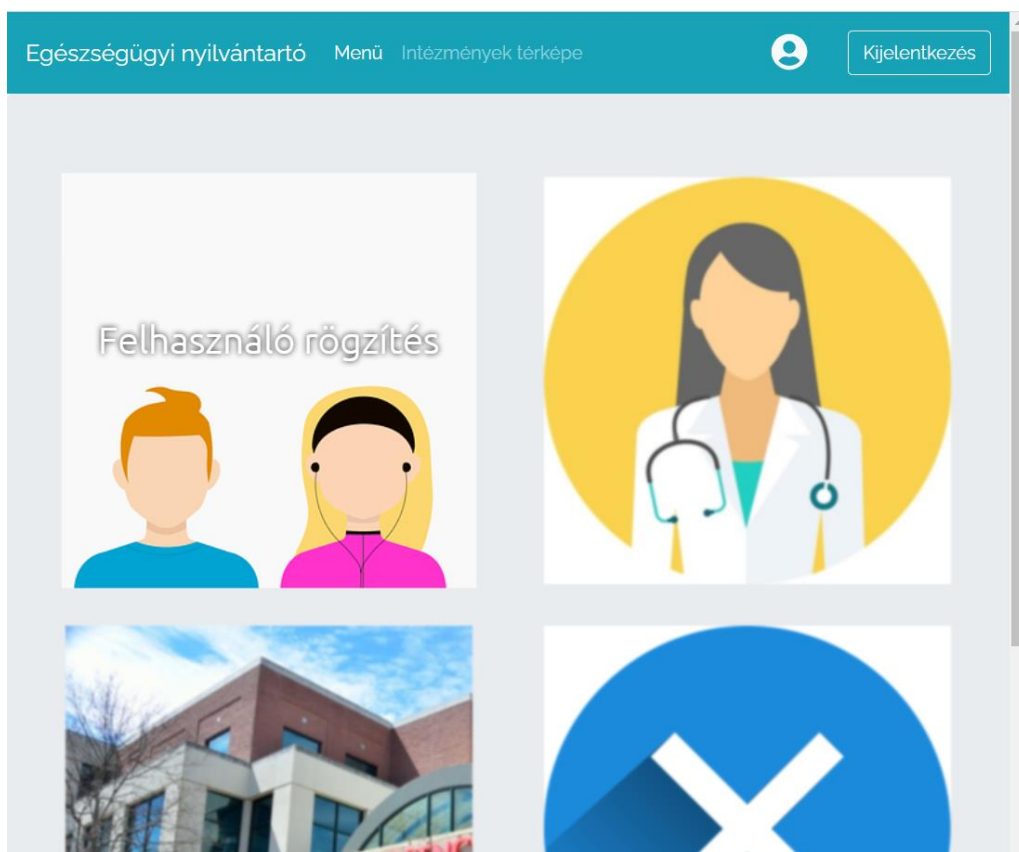
A páciensek dokumentumai ugyanolyan formában jelennek meg, mint fentebbi képernyőképeken (ld. 5.-6.-7. ábra). A dokumentumok alá kerül egy plusz gomb az új dokumentum felvitelére, amire kattintva megnyílik az ehhez szükséges ablak az űrlappal (ld. 18. ábra). A recept felvitelénél alapból 3 hónap van beállítva lejárat dátumnak.

The screenshot shows a web form titled 'Beutaló feltöltése' with a close button (X) in the top right corner. Below the title is a message: 'Az összes mezőt legyen szíves kitölteni!'. The form contains three input fields, each with a red asterisk indicating it is required. The first field is labeled 'Ok' and contains the text 'Ok'. The second field is labeled 'Kategória' and contains the text 'Kategória'. The third field is labeled 'Dokumentum' and contains the text 'dokumentum', with a 'Browse' button to its right. At the bottom of the form is a blue button labeled 'Feltölt'.

Beutaló feltöltése		
Az összes mezőt legyen szíves kitölteni!		
Ok	<input type="text" value="Ok"/>	*
Kategória	<input type="text" value="Kategória"/>	*
Dokumentum	<input type="text" value="dokumentum"/> <input type="button" value="Browse"/>	*
<input type="button" value="Feltölt"/>		

18. ábra Új dokumentum

2.3.2.4. Csak admin által elérhető funkciók és tartalmak



19. ábra Admin menüjének részlete

Az adminisztrátor menüjének elemei képekkel illusztrálva: „Felhasználó rögzítés”, „Dolgozó rögzítés”, „Intézmény rögzítés” és a „Felhasználó profiljának zárolása” (ld. 19. ábra).

Az admin szerepkörű felhasználó űrlapok kitöltésével bővítheti az adatbázist és a felhasználók körét. Az ő dolga regisztrálni a betegeket és dolgozókat. Ezzel elkerülhető lesz, hogy bárki regisztrálhasson dolgozónak, vagy hogy nem a saját TAJ számának megfelelő felhasználónévvel regisztráljon a páciens.

Az űrlapok a lentebbi ábrához (ld. 20. ábra) nagyon hasonlóan néznek ki. A megcsillagozott mezőket kötelező kitölteni, ha ez nem történik meg, a program hibát jelez.

Egészségügyi nyilvántartó Menü Intézmények térképe Kijelentkezés

Dolgozó feltöltése

Mindenhol tölts ki a formot!

Felhasználónév *

Jelszó *

Név *

Feltölt

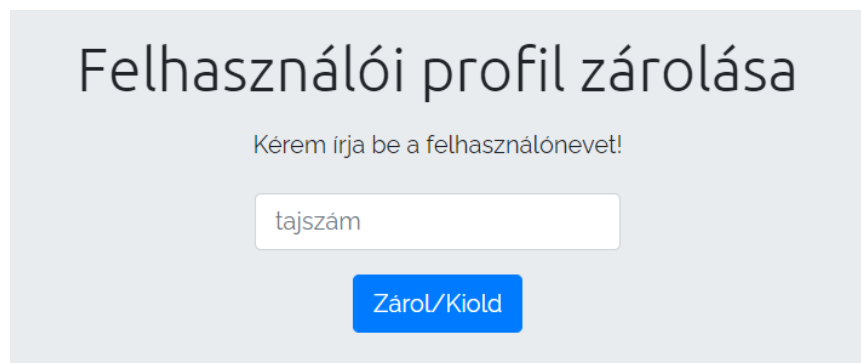
Excel fájl feltöltés

Fájl feltöltése

20. ábra Admin űrlapjai

A felhasználók excel fájlból is feltölthetők a rendszerbe. A betegek felviteléhez használandó fájlban fül neve: Felhasználók, fejléce: TAJ, JELSZÓ, NÉV, SZÜLETÉSI DÁTUM, SZÜLETÉSI HELY, JOGOSULTSÁG, LAKHELY, KÖRZET. A dolgozók felviteléhez használandó fájlban a fül neve: Dolgozók, a fejléce: FELHASZNÁLÓNÉV, JELSZÓ, NÉV. Amennyiben nem egyezik a fejléc az elvárttal, a program hibát ír. Amennyiben valahol az excel fájlban hibás formátumú adat (például dátum helyett szöveg) van, akkor a program kiírja, hogy melyik sorban történt beolvasási hiba. Amennyiben a fájl tartalmaz olyan felhasználónevet, mely már regisztrálva van a rendszerben, a program kiírja, hogy mely felhasználónév létezik már, és az adott sor előtti sorok adatait elmenti.

Intézmények felvitelénél meg kell adni az intézmény címét, körzetét, típusát, koordinátáit (amik a térképen való megjelenítéshez elengedhetetlenek) és egyéb információkat.



21. ábra Felhasználói profil zárolása

A felhasználók profiljait ezen az oldalon lehet aktiválni és zárolni (ld. 21. ábra). Olyan felhasználónév megadásával, mely nem szerepel a rendszerben, hibaüzenetet ad, ellenkező esetben viszont megváltozik a profil elérhetősége, és ki is írja a program, hogy aktiválva vagy zárolva lett.

2.3.3. Rendszerüzenetek

A program üzenetekkel reagál a felhasználók egyes cselekvéseire, és segíti őket a annak helyes használatában. Ezek az üzenetek lehetnek visszajelzések, illetve hibaüzenetek. Ha valamit nem megfelelően próbálunk elérni, a program hibaüzenetet dobhat. Ezeket az üzenetek soroltam fel alább, oldalanként csoportosítva.

Bejelentkezés

<p>Hibás felhasználónév vagy jelszó!</p> <p>Ez a felhasználó zárolva lett!</p>
--

Páciens profil

Email rögzítése	A bevitt érték nem email cím!
Háziorvos váltása	Még nem válthat háziorvost, mert még nem telt le a 1 év a legutóbbi váltása óta. (ld. 45. ábra) Már elküldött egy kérvényt, amit még nem bíráltak el. (ld. 11. ábra) Nincs orvos regisztrálva a körzetében.
Jelszó megváltoztatása	Nem helyes az eredeti jelszó! Nincs meg a 8 karakter! (ld. 9. ábra) Nem egyezik a két jelszó!

Páciens leletek, beutalók, receptek

Nincsenek feltöltött dokumentumai.

Profil zárolása

65421857 TAJ számhoz tartozó profil zárolva lett. ani123 felhasználónévhez tartozó profil aktiválva lett. Nincs a rendszerben ilyen nevű / TAJ számú felhasználó!

Kérvények

Nincs elbírálásra váró kérelem

Dolgozó profil

Rendelés helyszín váltás	Ebben a városban nincs egészségügyi intézmény
Jelszó megváltoztatása	Nem helyes az eredeti jelszó! Nincs meg a 8 karakter! (ld. 9. ábra) Nem egyezik a két jelszó!

Páciens keresése

Nincs a rendszerben ilyen TAJ számú felhasználó! (ld. 16. ábra)

A beírt érték nem TAJ szám!

Admin páciens, dolgozó felvitelénél

Már létezik felhasználó ilyen felhasználónévvel!

(ld. 35. ábra)

A fájlban nincs Felhasználók nevű munkalap

A fájlban nem helyes a fejléc! Az elvárt formátum: TAJ, JELSZÓ, NÉV, SZÜLETÉSI
DÁTUM, SZÜLETÉSI HELY, JOGOSULTSÁG, LAKHELY, KÖRZET

(ld. 31. ábra)

A fájlban nincs Dolgozók nevű munkalap

A fájlban nem helyes a fejléc! Az elvárt formátum: FELHASZNÁLÓNÉV, JELSZÓ, NÉV

Hiba történt a fájl beolvasása közben, helytelen formátum; ellenőrizze a(z) 6. sort!

(ld. 33. ábra)

Hiba történt a fájl beolvasása közben; ellenőrizze a(z) 8. sort!"

A feltöltött fájl nem .xlsx formátumú, kérem adjon meg excel fájlt!

(ld. 29. ábra)

Mindegyik mezőt töltsse ki!

Sikeres mentés!

(ld. 38. ábra)

Fájl sikeresen feltöltve, adatok sikeresen betöltve.

3. Fejlesztői dokumentáció

3.1. Tervek

A fejlesztendő program terve kezdetekben még képlékeny volt, mert nem tudtam pontosan hogy mit szeretnék, és mit tudok megvalósítani az idő és az eszközök korlátai mellett. Végül a tervek körülbelül 90%-a teljesült, teljesülhetett. Egypár kevésbé fontos dolgot kihagytam, míg más plusz funkciókat, melyek az eredeti tervekben nem szerepeltek, beleraktam. Ilyen például az email küldés a felhasználónak, ha megváltozik a háziorvosa, a profilok zárolása, vagy a dolgozó munkahely váltása. Kihagyott funkció például a dokumentumok konvertálása pdf-fé.

3.1.1. Fő funkcionális körök (szerepekre bontva):

Közös:

- Közös kezdőlap
- Térkép egészségügyi intézmények megjelenítéséhez
- Bejelentkezési felület

Páciens:

- Bejelentkezett kezdőlap
- Saját profil
- Saját dokumentumok

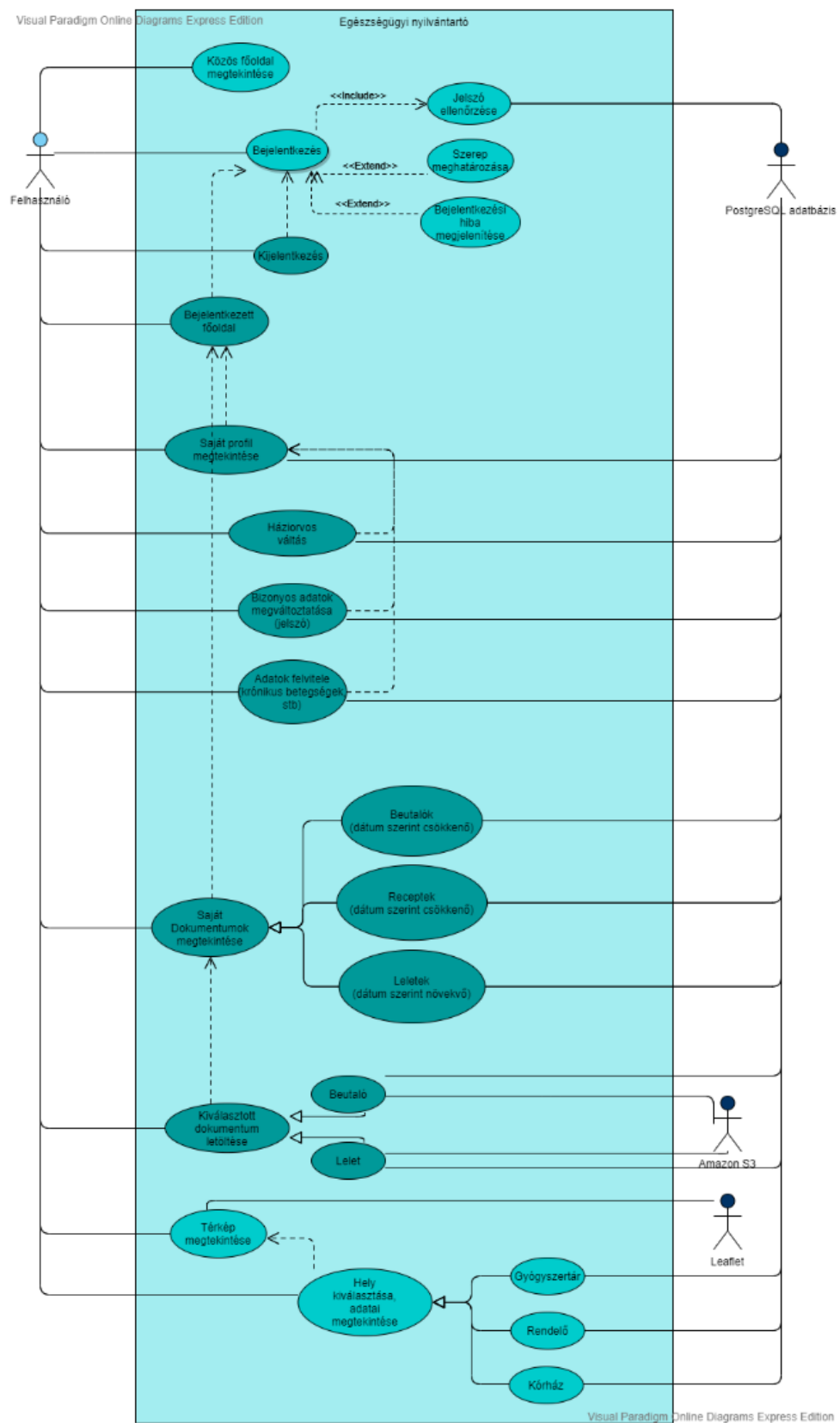
Dolgozó:

- Bejelentkezett kezdőlap
- Naptár
- Saját profil
- Beteg keresése, profilja
- Beteg dokumentumai

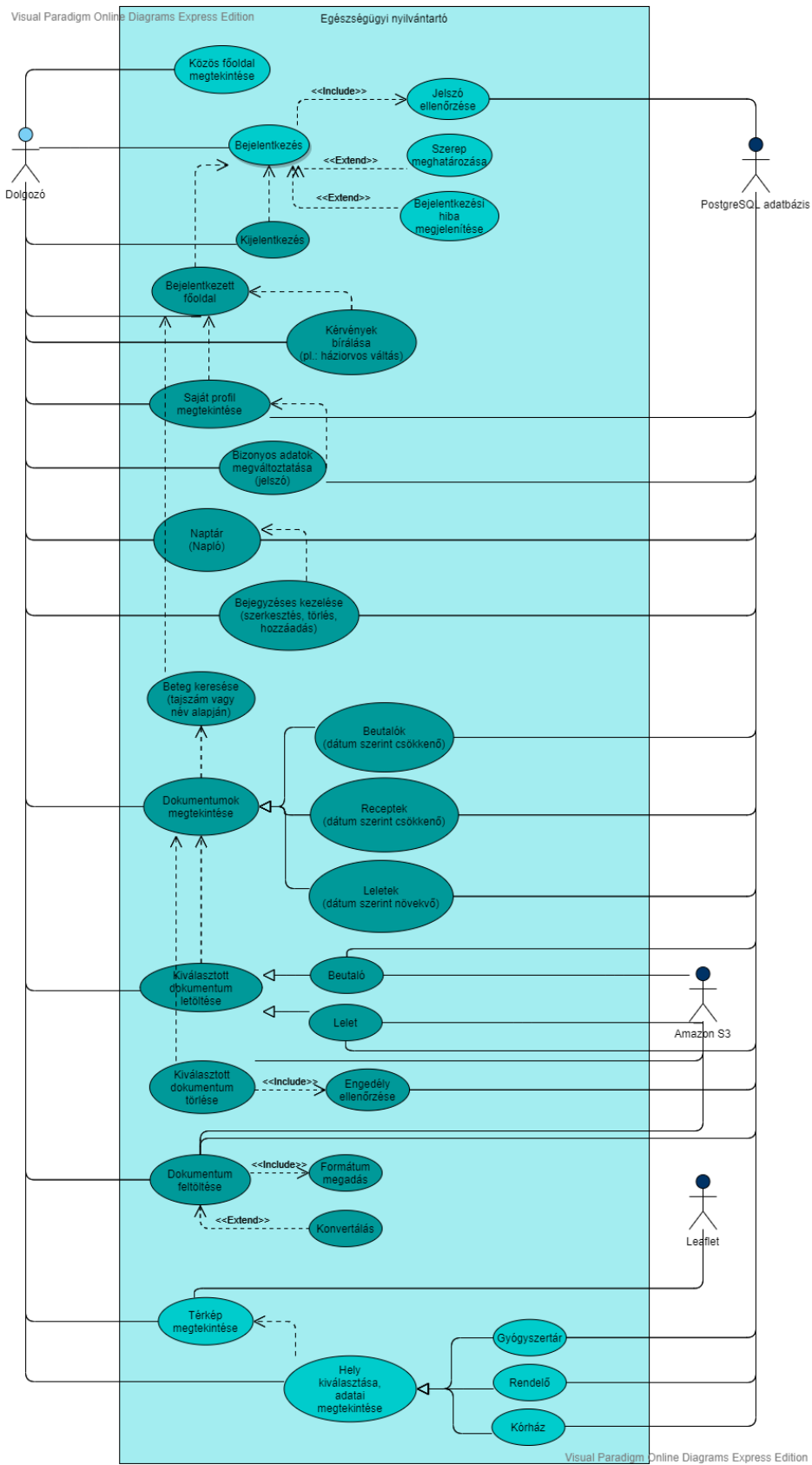
Admin:

- Bejelentkezett kezdőlap
- Saját profil
- Regisztrációs oldalak
- Intézmények adatai

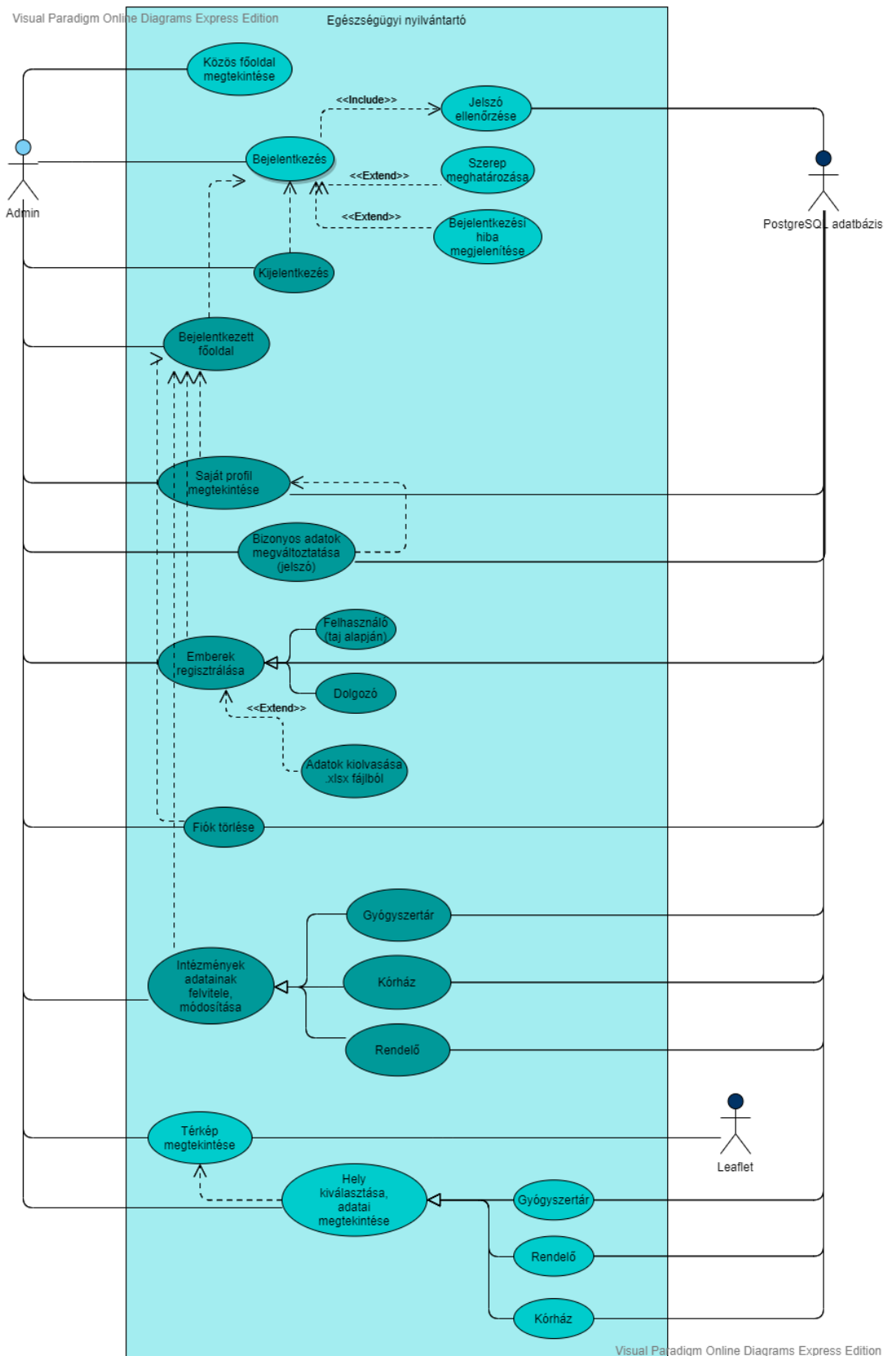
3.1.2. Use Case diagramok



22. ábra Felhasználóhoz tartozó use case diagram



23. ábra Dolgozóhoz tartozó use case diagram



24. ábra Adminhoz tartozó use case diagram

3.1.3. Nem funkcionális követelmények

Hatékonyág

- Rendszerkövetelmények: (a működő, de nem feltétlenül teljes funkcionalitású programhoz)
- Javascript futtatásra képes böngésző
- RAM: ≥ 1 GB
- Szabad hely: ≥ 1 GB
- Minden műveletre igaz legyen, hogy nem fog több időbe telni az elvégzése, mint 10 ms.
- Biztosítani kell az akadásmentes képvilágot a felhasználó számára.

Megbízhatóság

- Ha egy gép teljesíti a rendszerkövetelményben foglaltakat, akkor mindenképpen képes kell legyen a program futtatására. (Tehát a program futtathatósága a rendszerkövetelmények teljesítésének függvénye.)

Biztonság

- Amennyiben az eredeti példányát használja a felhasználó, úgy semmilyen kéréstlen program fertőzés nem származhat a használatából.
- A program nem igényel különösebb biztonsági intézkedéseket a felhasználók részéről.
- A program illetéktelen hozzáféréseket, kéréseket megtagad (például ha más betegek dokumentumait próbáljuk letölteni felhasználóként, vagy ha dolgozóként olyan dokumentumot törölnénk, amit nem mi töltöttünk fel).

Hordozhatóság

- A programot nem kell telepíteni a felhasználó részéről.
- A program minden olyan rendszeren elérhető, ahol böngésző fel van telepítve.
- Operációs rendszertől teljesen független.

Felhasználhatóság

- A program használata egyaránt könnyen elsajátítható a különféle háttérismeretekkel rendelkező felhasználóknak (intuitív).

3.1.3.1. Menedzselési követelmények

Működési:

- A program páciens szerepköri használatához nem szükséges semmilyen szaktudás.
- A program dolgozói szerepkörben történő használatához csak egészségügyi szaktudás kell.
- A programot egyéni preferencia szerint bármikor igénybe veheti bárki, bizonyos funkciókat pedig csak a regisztráltak, szerepkörhöz kötötten.

Fejlesztési:

- A programot könnyen kell tudnunk módosítani.
- A program könnyen konfigurálható legyen.
- Plusz funkciók igénye esetén biztosítani kell a minél könnyebb bővíthetőséget.

3.1.3.2. Külső követelmények

Etikai:

- Nincs korhoz kötve a program használata, mivel nem tartalmaz etikailag kérdéses tartalmat.

Adatvédelmi és biztonsági:

- A program tárolhat személyes adatnak minősülő adatokat (mint például lakcímet, orvosi leleteket, stb), de ezekhez csak az ahhoz jogosultak tudnak hozzáférni (maga a felhasználó, és bizonyos adatokhoz az orvosa).
- Az adatokat adatbázisban illetve fájlserveren tárolja a program, melyek védelméről a szolgáltató gondoskodik.
- A program használ külső céget, de nem szolgáltat feléjük adatot az egyezmény szerint.
- Egyébiránt nem sért adatvédelmi elveket.

3.1.4. User Story-k

Mindenki számára elérhető funkciók

Funkció megnevezése	Közös főoldal megtekintése	
1	Given	Nem áll az oldalon a felhasználó
	When	Beüti a weblap címét a böngésző címsorába
	Then	A weboldal közös főoldala jelenik meg
2	Given	Be van lépve, és meg szeretné tekinteni a főoldalt
	When	Kijelentkezik a kijelentkezés gombbal
	Then	Bejön a közös főoldal
3	Given	Az oldalon áll, és nincs bejelentkezve
	When	Megkeresi a menüpontok között a „Főoldal”-t és rákattint
	Then	Bejön a közös főoldal

Funkció megnevezése	Bejelentkezés	
1	Given	Megtalálta a bejelentkező felületet, valamint tudja a jelszavát és a felhasználónevét
	When	Beírja a jelszót és a felhasználónevet a felhasználó
	Then	Bejelentkezik a rendszer, és a szerepét is meghatározza a felhasználónak
2	Given	Megtalálta a bejelentkező felületet, de nem tudja a szükséges adatait
	When	Beír rossz jelszót és vagy felhasználónevet
	Then	Hibát ír az oldal, és újrapróbálhatja a belépést a felhasználó
3	Given	Be van jelentkezve a felhasználó, és másik fiókkal szeretne belépni
	When	A kilépő gombra kattint, majd a bejelentkező felületen helyes adatokat ír be

	Then	Bejelentkezik a rendszer a második fiókba
--	-------------	---

Funkció megnevezése	Kijelentkezés	
1	Given	Be van jelentkezve
	When	Megkeresi a kilépés gombot, és rákattint
	Then	Kijelentkezik a rendszer, és a főoldalra visz
2	Given	Nincs bejelentkezve
	When	-
	Then	-

Funkció megnevezése	Bejelentkezett menü	
1	Given	Nincs belépve
	When	Belépő felületen bejelentkezik
	Then	Bejelentkezett menü megjelenik
2	Given	Nincs belépve
	When	Nem lép be
	Then	Nem jelenik meg a menü főoldal, csak a közös főoldal
3	Given	Be van jelentkezve, és valamelyik másik oldalon áll
	When	Megkeresi a menü menüpontot, és rákattint
	Then	Megjelenik a bejelentkezett menü

Funkció megnevezése	Saját profil	
1	Given	Bejelentkezett főoldalon áll
	When	Rákattint a „Profil” menüpontra
	Then	Megjelenik a profil, a felhasználó adataival

Funkció megnevezése	Saját profil, adatok megváltoztatás	
1	Given	Profilon áll
	When	Rákattint a megváltoztatni kívánt adat melletti gombra, beírja az adatot, rányom a „Mentés” gombra
	Then	Az adatbázisban megváltozik az adat, és az új jelenik meg a profilon

Funkció megnevezése	Saját profil, jelszó változtatás	
1	Given	Profilon áll
	When	Rákattint a megváltoztatni kívánt adat melletti gombra, beírja helyesen a régi jelszót, és kétszer ugyanazt az új jelszót
	Then	Az adatbázisban megváltozik az jelszó, és így már ennek használatával léphet be a felhasználó
2	Given	Profilon áll
	When	Rákattint a megváltoztatni kívánt adat melletti gombra, beírja helytelenül a régi jelszót, és kétszer ugyanazt az új jelszót
	Then	A jelszó nem változik meg, és hibaüzenetet ír, miszerint rossz az eredeti jelszó
3	Given	Profilon áll
	When	Rákattint a megváltoztatni kívánt adat melletti gombra, beírja helyesen a régi jelszót, és kétszer két különböző új jelszót
	Then	A jelszó nem változik meg, és hibaüzenetet ír, miszerint az új jelszó két mezőjébe írt adat nem egyezik meg

Funkció megnevezése	Térkép megtekintése	
1	Given	Nincs bejelentkezve
	When	Rákattint a „Térkép” menüpontra
	Then	Megjelenik a térkép, és rajta a megjelölt helyek (kórházak, rendelők, gyógyszertárak)
2	Given	Be van jelentkezve, szerepköre páciens
	When	Rákattint a „Térkép” menüpontra
	Then	Megjelenik a térkép, és a rajta megjelölt helyek (kórházak, rendelők, gyógyszertárak)
3	Given	Egy intézmény adatlapját nézi
	When	Vissza gombra kattint
	Then	Megjelenik a térkép, és a rajta megjelölt helyek (kórházak, rendelők, gyógyszertárak)

Funkció megnevezése	Térkép, hely megtekintése	
1	Given	Térképen áll
	When	Rákattint egy megjelölt helyre
	Then	Megjelennek a hely adatai (rendelési idő, nyitvatartás, elérhetőség, ...)
2	Given	Térképen áll
	When	Rákattint a térképen egy helyre, ahol nincs megjelölt intézmény
	Then	-

Csak felhasználói szerepkörrel

Funkció megnevezése	Saját profil, adatok felvitele
------------------------	--------------------------------

1	Given	Profilon áll, páciensi szerepkörben
	When	Rákattint egy megváltoztatható adatra, beírja az adatot, rányom a megjelenő „Mentés” gombra
	Then	Az adatbázisban megváltozik az adat, és az új jelenik meg a profilon

Funkció megnevezése	Profil, háziorvos váltása	
1	Given	Profilon áll páciensi szerepkörben, már letelt 3 hónap után utoljára háziorvost váltott
	When	Rákattint a háziorvos nevére, majd egy felnyíló listában (ahol az ő körzetében szereplő háziorvosok vannak feltüntetve) kiválasztja az új háziorvosát
	Then	Az új háziorvos kérvényt kap a páciensről, de addig nem változik meg a páciens profilján a háziorvos az újra, amíg az el nem fogadta a kérvényt
2	Given	Profilon áll páciensi szerepkörben, még nem telt le a 3 hónap után utoljára háziorvost váltott
	When	Rákattint a háziorvos nevére
	Then	Kiírja a rendszer, hogy még nem válthat háziorvost

Funkció megnevezése	Dokumentumok (lelet, recept, beutaló)	
1	Given	Bejelentkezett főoldalon áll, páciensi szerepkörben, és van megjeleníthető dokumentuma
	When	Rákattint a „Leletek”, „Receptek”, vagy „Beutalók” menüpontra
	Then	Dokumentumok táblázatosan megjelennek dátum szerint rendezve

2	Given	Bejelentkezett főoldalon áll, páciensi szerepkörben, és nincs megjeleníthető dokumentuma
	When	Rákattint a „Leletek”, „Receptek”, vagy „Beutalók” menüpontra
	Then	A rendszer kiírja, hogy nincs megjelenítendő dokumentum

Funkció megnevezése	Dokumentum (lelet, beutaló) letöltése	
1	Given	A „Leletek” vagy a „Beutalók” oldalon áll
	When	Rákattint a felsorolt dokumentumok egyikére
	Then	A dokumentum letöltődik a „Letöltések” vagy a „Downloads” mappába

Csak dolgozó szerepkörrel

Funkció megnevezése	Kérvény bírálása	
1	Given	A bejelentkezett főoldalon áll, dolgozó szerepkörben
	When	Rákattint a „Kérvények” menüpontra, kiválasztja az adott kérvényt, és rányom az „Elfogadás” gombra
	Then	A kérvény ez alapján elfogadott lesz, és amennyiben a páciens rendelkezik email címmel, értesítjük a változásról. A páciens háziorvos megváltozik és az utolsó háziorvos váltása a jelen nap lesz
2	Given	A bejelentkezett főoldalon áll, dolgozó szerepkörben
	When	Rákattint a „Kérvények” menüpontra, kiválasztja az adott kérvényt, és rányom az „Elutasítás” gombra
	Then	A kérvény ez alapján elutasított lesz, a páciens oldalán nem változik semmi, és újra kérvényezhet háziorvos váltást.

Funkció megnevezése	Naptár	
1	Given	A bejelentkezett főoldalon áll, dolgozó szerepkörben
	When	Rákattint a „Naptár” menüpontra
	Then	A Naptár megjelenik, adott hónappal, és kijelöli az adott napot. Megjeleníti az enapi eseményeket (amennyiben van), és az esemény felviteli gombot.

Funkció megnevezése	Naptár, Napi teendő megtekintése	
1	Given	Dolgozó szerepkörben, a Naptár oldalon áll
	When	A naptáron kiválaszt egy napot.
	Then	A nap bejegyzései megjelennek időrendi sorrendben (amennyiben van aznap esemény).

Funkció megnevezése	Naptár, Napi teendő felvitele	
1	Given	Dolgozó szerepkörben, egy nap eseményeinél áll
	When	Rákattint a hozzáadás gombra, és a felugró ablakban kitölti a mezőket, majd rányom a „Mentés”-re
	Then	Bezáródik a felugró ablak, a nap bejegyzései közé elmentődik az esemény, és meg is jelenik az események sorában

Funkció megnevezése	Naptár, Napi teendő törlése	
1	Given	Dolgozó szerepkörben, egy nap eseményeinél áll
	When	Ráviszi a mutatót az eseményre amit ki akar törölni, ekkor megjelenik mellette egy „Törlés” gomb, amire rákattint. Majd a megjelenő ablakban jóváhagyja a törlést.

	Then	Az adott esemény törlődik a Napi események listájából
--	-------------	---

Funkció megnevezése	Beteg keresése, profiljának megtekintése	
1	Given	A bejelentkezett főoldalon áll, dolgozó szerepkörben
	When	Rákattint a „Beteg keresése” menüpontra, ahol egy beviteli mezőbe helyes, a rendszerben szereplő TAJ számot ír bele
	Then	Megjelenik a beteg profilja, és megtekinthetők a krónikus betegségei, születési adatai, lakcíme stb. és 3 másik gomb válik elérhetővé, a „Leletek”, „Beutalók”, és a „Receptek”
2	Given	A bejelentkezett főoldalon áll, dolgozó szerepkörben
	When	Rákattint a „Beteg keresése” menüpontra, ahol egy beviteli mezőbe helyes, a rendszerben nem szereplő TAJ számot ír bele
	Then	Megjelenik egy hibaüzenet arról, hogy ilyen felhasználó nincs regisztrálva a rendszerben.
3	Given	A bejelentkezett főoldalon áll, dolgozó szerepkörben
	When	Rákattint a „Beteg keresése” menüpontra, ahol egy beviteli mezőbe olyan szöveget ír be, mely nem felelhet meg egy TAJ számnak sem
	Then	Megjelenik egy hibaüzenet arról, hogy TAJ számot írjon be

Funkció megnevezése	Beteg adatainak módosítása	
1	Given	Dolgozó szerepkörben a „Beteg profilja” oldalon áll
	When	Rákattint a megváltoztatni kívánt adat mellett lévő gombra, és a felnyílt ablakban a beviteli mezőt/mezőket kitölti és elmenti
	Then	Megváltozik a páciens adatlapja.

Funkció megnevezése	Beteg dokumentumai	
1	Given	Dolgozó szerepkörben, egy beteg adatlapján áll
	When	Rákattint a „Leletek”, „Beutalók”, vagy a „Receptek” gombra
	Then	Megjelennek a dokumentumok idő szerinti sorrendben kilistázva, mellettük pedig pár fontosabb adat

Funkció megnevezése	Beteg dokumentumának (Lelet, Beutaló, Recept) felvitele	
1	Given	Dolgozó szerepkörben, egy beteg „Leletek”, vagy a „Beutalók” oldalán áll
	When	Rákattint a hozzáadás gombra, majd kitölti a mezőket, és tallózással kiválasztja a mappáiban a feltölteni kívánt dokumentumot (amennyiben nem Receptről van szó), „Mentés” gombbal ment
	Then	A dokumentum feltöltődik, mely a listában is megjelenik, és a beteg is meg tudja tekinteni
2	Given	Dolgozó szerepkörben, egy beteg „Leletek”, vagy a „Beutalók” oldalán áll
	When	Rákattint a hozzáadás gombra, majd nem tölt ki minden mezőt, vagy nem tallóz be dokumentumot (amennyiben nem Receptről van szó) és rányom a „Mentés” gombra
	Then	A dokumentum nem töltődik fel, és hibaüzenetet dob, hogy legyen szíves töltse ki az összes mezőt

Funkció megnevezése	Beteg dokumentumának (Lelet, Beutaló) letöltése
------------------------	---

1	Given	Dolgozó szerepkörben, egy beteg „Leletek”, vagy a „Beutalók” oldalán áll
	When	A listában egy kiválasztott dokumentum melletti gombra kattint
	Then	A dokumentum letöltődik a „Letöltések” vagy a „Downloads” mappába

Csak admini szerepkörrel

Funkció megnevezése	Emberek regisztrálása oldal	
1	Given	Admin szerepkörben, a bejelentkezett főoldalon áll
	When	A „Regisztrálás” menüpontra kattint
	Then	Betöltődik három lehetőség: „Admin”, „Beteg”, „Dolgozó”

Funkció megnevezése	Emberek regisztrálása (Admin, Dolgozó, Beteg)	
1	Given	Admin szerepkörben, a „Regisztrálás” oldalon áll, egy fiókot szeretne létrehozni
	When	Kiválasztja az „Admin”, „Dolgozó”, vagy a „Beteg” menüpontot, és a megjelenő mezőket kitölti, olyan felhasználónévvel (vagy TAJ számmal), ami még nem lett regisztrálva
	Then	Elmenti az adatokat, és ezzel létrehoz egy új fiókot
2	Given	Admin szerepkörben, a „Regisztrálás” oldalon áll, egy fiókot szeretne létrehozni
	When	Kiválasztja az „Admin”, „Dolgozó”, vagy a „Beteg” menüpontot, és a megjelenő mezőket kitölti, olyan felhasználónévvel (vagy TAJ számmal), ami már regisztrálva volt

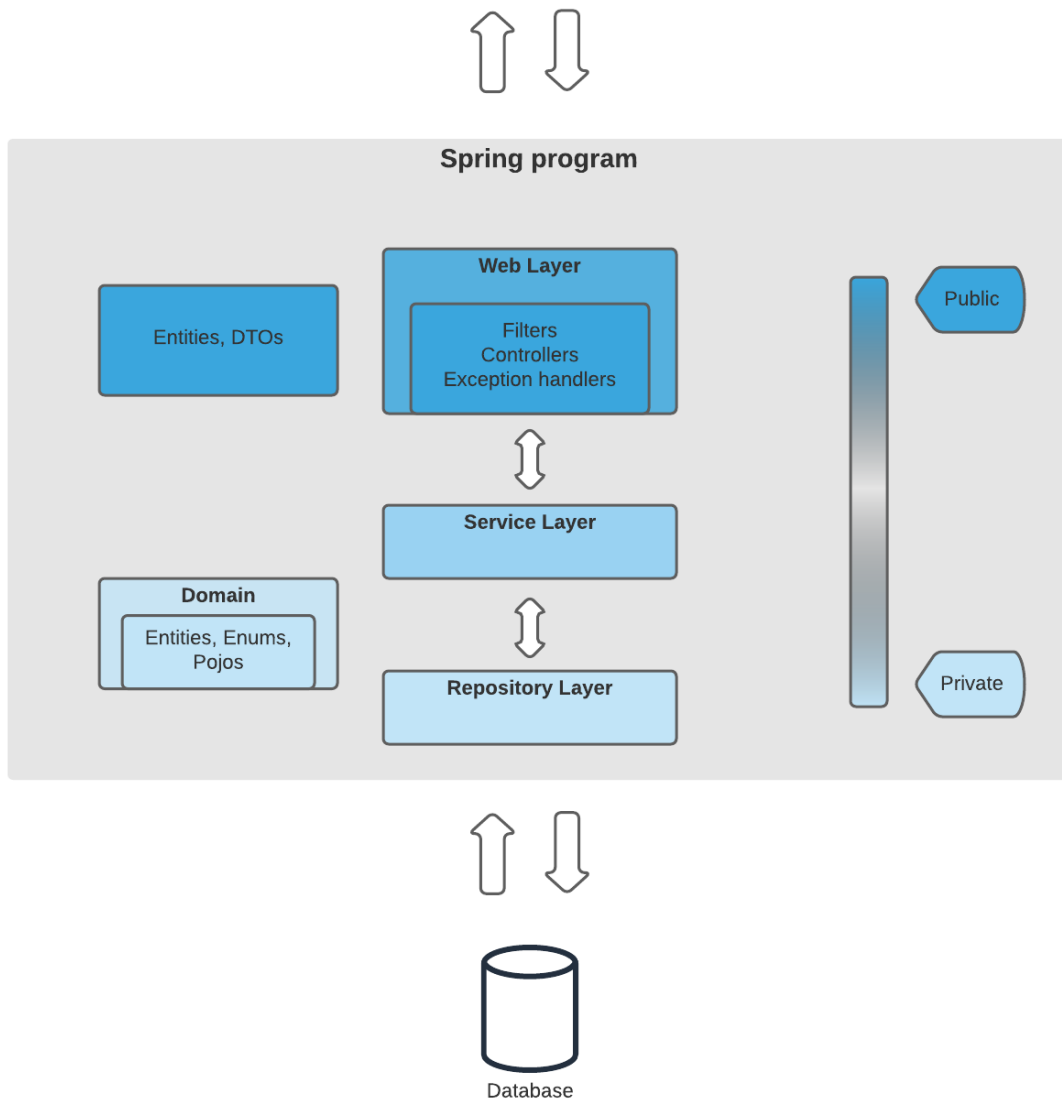
	Then	A rendszer nem menti le az adatokat, és kiírja, hogy ilyen felhasználóval már létezik felhasználó
3	Given	Admin szerepkörben, a „Regisztrálás” oldalon áll, több fiókot szeretne létrehozni
	When	Kiválasztja az „Admin”, „Dolgozó”, vagy a „Beteg” menüpontot, és beimportál egy excel táblázatot, ami helyesen van kitöltve
	Then	Elmenti az adatokat, és ezzel létrehozza az új fiókokat
4	Given	Admin szerepkörben, a „Regisztrálás” oldalon áll, több fiókot szeretne létrehozni
	When	Kiválasztja az „Admin”, „Dolgozó”, vagy a „Beteg” menüpontot, és beimportál egy excel táblázatot, amiben rossz fejlécek (és emellett esetleg rossz adatok is) szerepelnek
	Then	A rendszer nem menti el az adatokat, és szól, hogy a fájlban szereplő fejlécek rosszak, és azt is kiírja, hogy miknek kell lenniük
5	Given	Admin szerepkörben, a „Regisztrálás” oldalon áll, több fiókot szeretne létrehozni
	When	Kiválasztja az „Admin”, „Dolgozó”, vagy a „Beteg” menüpontot, és beimportál egy excel táblázatot, amiben rossz formátumú adat található
	Then	A rendszer egy felhasználót sem ment el a fájlból, és kiírja a hibát, hogy az adott sorban hiba van
6	Given	Admin szerepkörben, a „Regisztrálás” oldalon áll, több fiókot szeretne létrehozni
	When	Kiválasztja az „Admin”, „Dolgozó”, vagy a „Beteg” menüpontot, és beimportál egy excel táblázatot, amiben van olyan felhasználó, aki már regisztrálva lett
	Then	A rendszer a még nem létező fiók adatait elmenti, az első létező után azonban megáll, és kiírja hogy melyik

		sorban szerepel az a felhasználó, aki már benne van az adatbázisban
7	Given	Admin szerepkörben, a „Regisztrálás” oldalon áll, több fiókot szeretne létrehozni
	When	Kiválasztja az „Admin”, „Dolgozó”, vagy a „Beteg” menüpontot, és beimportál egy fájlt, ami nem excel formátumú
	Then	A rendszer hibát ír ki arról, hogy a feltöltött fájl nem excel

Funkció megnevezése	Térkép adatainak felvitele	
1	Given	Admin szerepkörben, a bejelentkezett főoldalon áll, egy új intézmény adatait szeretné rögzíteni
	When	A „Térkép adatai” menüpontra kattint, kitölti a megjelent mezőket, rákattint a „Mentés” gombra
	Then	Elmenti az intézmény adatait
2	Given	Admin szerepkörben, a bejelentkezett főoldalon áll, egy új intézmény adatait szeretné rögzíteni
	When	A „Térkép adatai” menüpontra kattint, nem tölt ki minden szükséges mezőt, rákattint a „Mentés” gombra
	Then	Hibaüzenetet ír, mely felhívja a figyelmet, hogy az összes megjelölt mezőt ki kell tölteni

3.2. Spring Boot program

A szerveroldali alkalmazás egy Spring Boot program, mely elsősorban *RestController*-ekből, *Service*-ekből, *Entity*-kből, és *Repository*-kből épül fel. A program alapvető kommunikációját és működését a lentebbi ábra szemlélteti:



25. ábra Spring program működése⁵

3.2.1. Fejlesztői környezet

A szerveroldali program Spring Tool Suite-ban készült, ami lehet egy plugin vagy egy standalone Eclipse JEE, én az utóbbit használtam.

⁵ EPAM - ELTE_JPA_Spring_Data.pdf

3.2.2. Package-ek

hu.elte.szakdolgozat.egeszsegugyinyilvantarto...

- ❖ config
 - security
 - jwt
 - model
- ❖ controller
- ❖ domain
 - enums
 - documents
 - users
- ❖ exception
- ❖ repo
 - documents
 - users
- ❖ service

3.2.3. Osztályok

3.2.3.1. Entity-k

- **Establishment (EstablishmentType)**

Az intézmények adatait tárolja.

Adatai: *város, utca, házszám, körzetszám, intézmény típusa* mely lehet kórház, rendelő, és gyógyszerár (ezeket az értékeket a EstablishmentType enum határozza meg), továbbá a *telefonszámot, nyitvatartást, egy képet az intézményről és a földrajzi elhelyezkedését* is, mely a térképen való megjelenítéshez szükséges.

- **Event**

A kórházi dolgozók eseményeit tárolja.

Adatai: *esemény leírása, esemény kezdetének ideje, és a dolgozó, akihez az esemény tartozik.*

- **Request (RequestType)**

A háziorvos váltásra irányuló kérvények eltárolására szolgál.

Adatai: A *páciens*, akitől a kérvény érkezett, az *orvos*, akinek a kérvény szól, és az *elbírálás*, mely 3 féle lehet – elutasított, elfogadott, függő.

- **Assignment**

A páciensek beutalóit tárolja. Adatai:

Az a *dátum*, amikor felkerült a rendszerbe, kategóriája, a beutalás *motivációja*, a *felhasználó*, akinek a beutalót írták, és az *orvos*, aki megírta. Továbbá még a *fájl neve* is eltárolódik, ami az Amazon S3-ban azonosítja az oda feltöltött fájlt.

- **DiagnosticReport**

A páciensek leleteit tárolja.

Adatai: Az a *dátum*, amikor felkerült a rendszerbe, *kategóriája*, *megjegyzés*, a *felhasználó*, akinek a beutalót írták, és az *orvos*, aki megírta. Továbbá még a *fájl neve* is eltárolódik, ami az Amazon S3-ban azonosítja az oda feltöltött fájlt.

- **Prescription**

A páciensek receptjeit tárolja.

Adatai: Az a *dátum*, amikor felkerült a rendszerbe, a *lejárat dátuma*, a *gyógyszer neve*, *javasolt szedése*, a *felhasználó*, akinek a receptet írták, és az *orvos* aki megírta.

- **Admin**

Az Admin szerepkörű felhasználók adatait tartalmazza.

Adatai: *Név*, egyedi *felhasználónév*, titkosított *jelszó*.

- **User (Entitlement)**

A felhasználó (=páciens) szerepkörű felhasználók adatait tárolja.

Adatai: *TAJ szám* (ezzel azonosítja magát), titkosított *jelszó*, és a profilra vonatkozó *zárolás*. *Név*, *születési dátum*, *születési hely*, *jogosultság*, melynek 5 fajta lehet (piros, sárga, zöld, barna és kék) és „lámpás” jogszabályra vonatkozik.⁶ *Email címe* ha van, *lakcíme*, *körzetszáma*. *Krónikus betegségeinek* listája, *allergiáinak* és *érzékenységeinek* listája. *Háziorvosa*, és a legutolsó *házi orvos váltásának dátuma*. *A leletei*, *beutalói* és a *receptjei*.

- **Worker**

Az egészségügyi dolgozók adatait tárolja.

Adatai: Egyedi *felhasználónév*, titkosított *jelszó*, és a profilra vonatkozó *zárolás*. *Név*, *rendelés helye*, *körzetszáma*, *telefonszáma*, ha van. *Páciensei*, és felvitt *eseményei*.

3.2.3.2. *RestController-ek*

Négy *RestController* osztályt csináltam, aszerint hogy milyen szerepkör férhet hozzá az *Url*-ekhez. Vannak olyan *url*-ek az osztályok között, melyek formában hasonlítanak, azonban fontos, hogy ezek sem egészen ugyanazt a feladatot látják el. A *kontrollerek* a *HomeController* kivételével, mind használják az *SecurityContextHolder*-t, amiből az autentikált felhasználót lekérve, automatikusan csak azokhoz az adatokhoz nyúl hozzá, mely a felhasználóhoz tartozik. Így például a páciensek véletlenül sem tudják más páciensek dokumentumait lekérni, sem pedig a profiljukban az adatokon változtatni.

- **HomeController**

- */map/town/{town}* - város alapján intézmények lekérése
- */map/establishment/{id}* - id alapján intézmény lekérése
- */map/town/{town}* - város alapján intézmények lekérése
- */map/location* – földrajzi hely alapján intézmények lekérése

⁶ http://www.neak.gov.hu/felso_menu/lakossagnak/ellatas_magyarorszagon/jogosultsag_az_ellatasra/ellatasra_jogosultsag_igazolasa/jogviszony_rendezeresi_eljaras

- `/authenticate (POST)` - felhasználónév, jelszó megadásával autentikáció, JwtResponse-t (mely tokenet tartalmaz) ad vissza

- **WorkerController**

- `/worker/profile (GET)` - dolgozó adatainak lekérése
- `/worker/profile (PUT)` - dolgozó adatainak megváltoztatása
- `/worker/establishments/town/{town} (GET)` – intézmények lekérése város alapján
- `/worker/findUser/{taj} (GET)` – páciens megtalálása taj alapján
- `/worker/findUser/{taj} (PUT)` – páciens adatainak megváltoztatása taj alapján
- `/worker/requests (GET)` – kérvények lekérése
- `/worker/requests/number (GET)` – elbírátlan kérvények számának lekérése
- `/worker/request/judge/{id} (PUT)` – kérvény elbírálása id alapján
- `/worker/assignments/{taj}` – beutaló megtalálása felhasználó TAJ száma alapján
- `/worker/assignment/{id} (GET)` – beutaló megtalálása id alapján
- `/worker/assignment/download/{id}` – beutaló letöltése id alapján
- `/worker/assignment/{taj} (POST)` – beutaló feltöltése felhasználó TAJ száma alapján
- `/worker/assignment/file/{id} (PUT)` – beutalóhoz tartozó fájlt feltöltése id alapján
- `/worker/assignment/delete/{id}` – beutaló törlése id alapján
- `/worker/diagnosticreports/{taj}` – lelet megtalálása felhasználó TAJ száma alapján
- `/worker/diagnosticreport/{id} (GET)` – lelet megtalálása id alapján
- `/worker/diagnosticreport/download/{id}` – lelet letöltése id alapján
- `/worker/diagnosticreport/{taj} (POST)` – lelet feltöltése felhasználó TAJ száma alapján
- `/worker/diagnosticreport/file/{id} (PUT)` – lelethez tartozó fájlt feltöltése id alapján

- `/worker/diagnosticreport/delete/{id}` – lelet törlése id alapján
- `/worker/prescriptions/{taj}` – recept megtalálása felhasználó TAJ száma alapján
- `/worker/prescription/{id} (GET)` – recept megtalálása id alapján
- `/worker/prescription/{id} (POST)` – recept feltöltése id alapján
- `/worker/prescription/delete/{id}` – recept törlése id alapján
- `/worker/calendar/day/{day}` – napra és dolgozóra vonatkozó események lekérése
- `/worker/calendar/event/{id}` – esemény lekérése id alapján
- `/worker/calendar/event (POST)` – esemény felvitele
- `/worker/calendar/event (PUT)` – esemény megváltoztatása
- `/worker/calendar/event/delete/{id}` – esemény törlése id alapján

- **UserController**

- `/user/profile (GET)` – saját profil adatok lekérése
- `/user/profile (PUT)` – saját profil adatok módosítása
- `/user/worker/region` – saját körzetben rendelő dolgozók lekérése
- `/user/canChange` – tud-e háziorvost váltani (letelt-e a jogszabályzatban előírt idő, azóta hogy utoljára cserélt)
- `/user/worker/request/{id} (POST)` – háziorvos váltási kérelem felvitele háziorvos id alapján
- `/user/assignments` – saját beutalók lekérése
- `/user/assignment/{id}` – saját beutaló adatok id alapján
- `/user/assignment/download/{id}` – saját beutaló letöltése id alapján
- `/user/diagnosticreports` – saját leletek
- `/user/diagnosticreport/{id}` – saját lelet adatok id alapján
- `/user/diagnosticreport/download/{id}` – saját lelet letöltése id alapján
- `/user/prescriptions` – saját receptek
- `/user/prescription/{id}` – saját recept adatok id alapján

- **AdminController**

- `/admin/profile (GET)` – admin adatainak lekérése
- `/admin/profile (PUT)` – admin adatainak megváltoztatása
- `/admin/user (POST)` – páciens feltöltése
- `/admin/worker (POST)` – dolgozó feltöltése
- `/admin/admin (POST)` – admin feltöltése
- `/admin/establishment (POST)` – intézmény feltöltése
- `/admin/establishmentPic/{id} (PUT)` – intézményhez kép feltöltése
intézmény id alapján
- `/admin/user/upload (POST)` – pácienseket tartalmazó excel fájl feltöltése
- `/admin/worker/upload (POST)` – dolgozókat tartalmazó excel fájl feltöltése
- `/admin/disable_profile/{username} (PUT)` – profil státuszának (aktív, zárolt) megváltoztatása felhasználónév alapján
- `/admin/change_entitlement/{taj} (PUT)` – páciens jogosultságának megváltoztatása TAJ szám alapján

3.2.3.3. *Service-ek*

Mivel körülbelül ugyanolyan formában szeretném a frontenden megkapni az adatokat, mint ahogy az adatbázisban tárolom, azért nem írtam külön *Data Transfer Object*-eket erre a feladatra, hanem a már meglévő *Entity*-ket használom az adat átküldéséhez. A megírt *service*-k emiatt legfőképp arra szolgálnak, hogy kiszűrjék a megkért adatokat, hogy ne küldjünk át a hálózaton illetéktelen vagy felesleges információt. De vannak olyan *service*-k is, melyek ettől specifikusabb feladatot látnak el, például a *MailService*, az *AmazonClient*, és az *ExcelService*. Ezekről lejjebb írok bővebben.

- **AssignmentService**
- **DiagnosticReportService**
- **PrescriptionService**
- **AdminService**
- **UserService**
- **WorkerService**

- **AmazonClient**
- **ExcelService**
- **MailService**
- **InitDBService**

ExcelService

Excel fájlt feldolgozó osztály. Az adatok könnyebb felvitele miatt raktam bele a programba egy olyan funkciót, hogy excel fájlból ki lehessen olvasni a felhasználókat, illetve dolgozókat, majd ezeket le lehessen menteni). A poi-ooxml függőséget használja.

Ez az osztály 4 metódussal rendelkezik:

- *hasExcelFormat(MultipartFile file)* – megvizsgálja hogy a megkapott fájl ténylegesen excel formátumú-e
- *hasRightHeaders(Row header, String[] rightHeader)* – megvizsgálja hogy a fájlban megfelelők-e az oszlopai, egyezik-e a fejléc azzal, ami alapján be tudjuk olvasni helyesen a fájlt. 2 lehetséges fejléc van:
 - `USER_HEADERS = { "TAJ", "JELSZÓ", "NÉV", "SZÜLETÉSI DÁTUM", "SZÜLETÉSI HELY", "JOGOSULTSÁG", "LAKHELY", "KÖRZET" }`
 - `WORKER_HEADERS = { "FELHASZNÁLÓNÉV", "JELSZÓ", "NÉV" }`
- *excelToUsers(MultipartFile file)* – a fájlból, amennyiben excel, és rendben van a fejléce is, soronként kiolvassa az adatokat, User objektumokat gyárt belőle, és egy User-ekből álló listát ad vissza.
- *excelToWorkers(MultipartFile file)* – a fájlból, amennyiben excel, és rendben van a fejléce is, soronként kiolvassa az adatokat, Worker objektumokat gyárt belőle, és egy Worker-ökből álló listát ad vissza.

Az osztály, amennyiben valamilyen hibát észlel, *ApiException*-t dob, a hibát leíró üzenettel, mint például: „A fájlban nem helyes a fejléc! Az elvárt formátum: TAJ, JELSZÓ, NÉV, SZÜLETÉSI DÁTUM, SZÜLETÉSI HELY, JOGOSULTSÁG, LAKHELY, KÖRZET” , vagy „Hiba történt a fájl beolvasása közben, helytelen formátum; ellenőrizze a " + rowId + " .

sort!”, stb. Ezt a kivételt az *ApiExceptionHandler* kapja el, és *ExceptionPojo* formában küldi el a kliensnek.

MailService

Ennek az osztálynak csak egyetlen metódusa van, és arra a célra hoztam létre, hogy e-mailt tudjon küldeni a szerver egy páciensnek (amennyiben van megadva email címe) arról, ha elfogadták a háziorvos váltási kérelmét. A *spring-boot-starter-mail* függőséget használja.

AmazonClient

Az *AmazonClient* az AWS S3-hoz csatlakozik, ami egy fájlserver. A dokumentumok feltöltését, leöltését, és törlését hivatott megvalósítani ez az osztály. A *aws-java-sdk* függőséget használja. Rendelkezik egy *AmazonS3* objektummal, és 7 metódusa van:

- *initializeAmazon()* – a hitelesítő adatok segítségével létrehozza a kapcsolatot az Amazon s3-al
- *convertMultiPartToFile(MultipartFile mfile)* – multipart fájlból fájl csinál
- *uploadFileToS3bucket(String fileName, File file)* – feltölti az S3-ba a fájlt a megadott fájlnev szerint
- *uploadFile(MultipartFile multipartFile, String fileName)* – aszinkron metódus, mely meghívja a *uploadFileToS3bucket*-t, és elkapja a kivételt amennyiben keletkezik
- *deleteFileFromS3Bucket(String fileName)* – törli az adott fájlnevvvel rendelkező fájlt S3-ból
- *downloadFile(String fileName)* – aszinkron metódus, mely meghívja a *deleteFileFromS3Bucket*-ot, és elkapja a kivételt amennyiben keletkezik
- *generateUrl(String picName)* – 1 órára aktív publikus URL-t generál arra a fájlra (képekre) amiknek a nevét megkapja paraméterként

3.2.3.4. Security

Illetéktelen hozzáférést hivatott megakadályozni. A felhasználóknak azonosítaniuk kell magukat, ezután egy tokent kapnak, ami felhasználásával hozzáférhetnek a nekik szánt információkhoz.

- **AdminDetailsImpl (UserDetails)**
- **WorkerDetailsImpl (UserDetails)**
- **UserDetailsImpl (UserDetails)**
- **SecurityConfig (WebSecurityConfigurerAdapter)**
- **JwtAuthenticationEntryPoint (AuthenticationEntryPoint)**
- **JwtRequestFilter (OncePerRequestFilter)**
- **JwtTokenUtil**
- **JwtRequest (DTO)**
- **JwtResponse (DTO)**
- **(UserService (UserDetailsService))**

Az *AdminDetailsImpl*, a *WorkerDetailsImpl* és a *UserDetailsImpl* a *UserDetails*-t implementálják, mely a felhasználók autentikációs adatait és a szerepköreit tárolja, miután lekértük ezeket az adatbázisból.

A *UserService* a *UserDetailsService*-t implementálja, és tartalmazza annak *loadUserByUsername* metódusát. Ez keresi ki a felhasználót (aki lehet páciens, dolgozó, vagy admin) felhasználónév alapján az adatbázisból, és hozza létre a hozzá tartozó *UserDetails* objektumot.

Az autentikáció a *HomeController*-nél kezdődik, pontosabban annak a *createAuthenticationToken* metódusánál, ami egy *JwtRequest* objektumot kap (mely a felhasználónevet és a jelszót tartalmazza). Itt az *AuthenticationManager*-nél megpróbáljuk autentikálni a felhasználót, majd ha ez sikerült, lekérjük a hozzá tartozó *UserDetails* objektumot a *UserService*-től, majd generálunk ahhoz egy *json web token*t a *jwtTokenUtil* segítségével, és azt belerakjuk a *JwtResponse*-ba, amit elküldünk a kliensnek. Végezetül ez az objektum megérkezik a kliens oldalra, és ezt használhatja a felhasználó (lejártáig) arra, hogy azonosítsa magát.

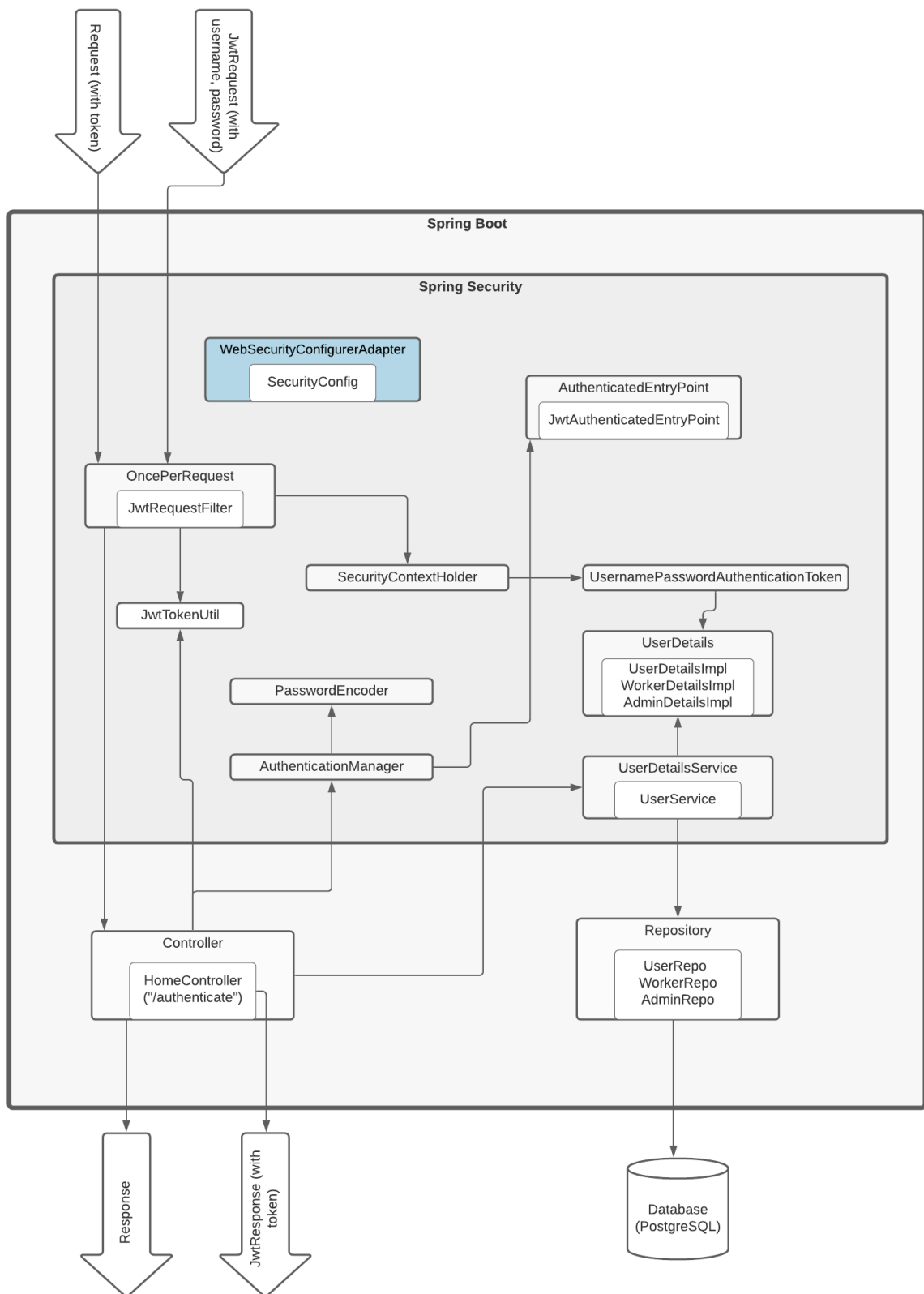
A további lekéréseknél már a *JwtRequestFilter* felügyeli a jwt token meglétét, és helyességét (a *JwtTokenUtil* segítségével), és állítja be a *SecurityContext*-ben a felhasználót, akit az adott jwt azonosít.

A *SecurityConfig* Beanjei:

- *BCryptPasswordEncoder passwordEncoder()*
- *WebMvcConfigurer corsConfigurer()* – itt beállítottam a megengedett származási helyét a kéréseknek, a megjelenített Header-öket a válaszokban, és a megengedett http metódusokat
- *AuthenticationManager authenticationManagerBean()*

A SecurityConfig további metódusai:

- *configureGlobal(AuthenticationManagerBuilder auth)* – beállítottam melyik userDetailsService-t, illetve hogy melyik jelszó kódolót használja a program
- *configure(HttpSecurity httpSec)* – beállítottam az autentikációra vonatkozó kivétel kezelést, melyet ezután a *JwtAuthenticationEntryPoint* végez, továbbá az url-ek hozzáférhetőségét, a *sessionManagement*-et, és hozzáadtam a filterek listájához a *jwtRequestFilter*-t.



26. ábra Spring programban a Security működése⁷

⁷ <https://youtu.be/CsgtYvIR7xk> videóban szereplő ábrán alapul

3.2.4. Kivételek

Az alkalmazáshoz írtam 3 új kivételt. Az *ApiExceptionHandler*, mely a *ExceptionHandlerExceptionResolver*-t terjeszti ki, és kapja el a controllerekhez beérkező és az után keletkezett kivételeket, és továbbítja őket a kliens felé. (A filterekben keletkezőket nem tudja elkapni.)

- **ApiException (RuntimeException)**
- **ForbiddenException (RuntimeException)**
- **JwtException (RuntimeException)**
- **ExceptionPojo**

3 adatot tárol: a hiba üzenetet, a kivételt, és a http státuszt.

- **ApiExceptionHandler (ExceptionHandlerExceptionResolver)**

Elkapja a kivételeket melyek adataiból *ExceptionPojo*-t csinál, és amit DTO-nek használ. Ezt küldi át a hálózaton a kliensnek.

3.2.5. Függőségek

- **spring-boot-starter-web**

Webes alkalmazások készítéséhez (beleértve a RESTful alkalmazást is) kezdő csomag mely Spring MVC-t használ. Tomcat-at használ alapértelmezett beágyazott tárolónak.⁸

- **spring-boot-starter-data-jpa**

Kezdő csomag a Spring Data Java Persistent API (JPA) Hibernate-tel való használatához.⁹

A Spring Data egy magas szintű Spring Source projekt. A Spring Data egy plusz réteget ad a JPA-hoz, melynek köszönhetően nem kell implementálni a repository-kat, és ki tudja következtetni a lekérdezéseket a függvények neveiből. Boilerplate kódok generálásával nagyban megkönnyíti a programozó dolgát.¹⁰

⁸ <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-web>

⁹ <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-data-jpa>

¹⁰ <https://www.javatpoint.com/spring-boot-starter-data-jpa>

- **spring-boot-starter-jdbc**

Kezdő csomag a JDBC használatához a HikariCP kapcsolatkészlettel.¹¹

A Java Database Connectivity, röviden JDBC egy API, mely definiálja az adatbázisok lekérdezéséhez és módosításához szükséges osztályokat és metódusokat.¹²

- **postgresql**

PostgreSQL JDBC illesztőprogram (driver)

Postgres egy nyílt forráskódú relációsadatbázis-kezelő rendszer.

- **h2**

(Hypersonic 2) Nyílt forráskódú relációsadatbázis-kezelő rendszer, melyet Javában írtak. Memóriában tárolódik, és minden egyes leállításkor törlődik. Teszteléshez használom.

- **spring-boot-starter-security**

Kezdő csomag a Spring Security használatához.¹³

A Spring Security egy nagymértékben testreszabható hitelesítési és hozzáférés-ellenőrzési keretrendszer, folyamatosan bővülő lehetőségekkel.¹⁴

- **spring-security-test**

A Spring Security bekapcsolódik a Spring Test tesztelési folyamatába a WithSecurityContextTestExecutionListener osztállyal, amely biztosítja, hogy tesztheink a megfelelő felhasználóval fussanak.¹⁵

- **jaxb-api**

¹¹ <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-jdbc>

¹² https://hu.wikipedia.org/wiki/Java_Database_Connectivity

¹³ <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-security>

¹⁴ <https://spring.io/projects/spring-security>

¹⁵ <https://docs.spring.io/spring-security/site/docs/4.2.x/reference/html/test-method.html>

A JAX-B (Java Architecture for XML Binding - Java Architektúra XML kötéshez) lehetővé teszi a Java fejlesztők számára, hogy feltérképezzék a Java osztályt XML reprezentáción keresztül. A JAXB két fő jellemzője: képes a Java objektumokat XML állományokba menteni és onnan visszatölteni.¹⁶

- **jjwt**

JSON Web Token támogatás a JVM-nek.¹⁷

A JWT egy nyílt forráskódú Json web token könyvtár, amely lehetővé teszi bármely java alkalmazás számára a hozzáférés létrehozását és ellenőrzését, valamint a tokenek frissítését.¹⁸

- **poi-ooxml**

Apache POI - Java API Microsoft formátumú fájlok eléréséhez.¹⁹

- **spring-boot-starter-mail**

Kezdő csomag mely tartalmazza a Java Mail-t és a Spring keretrendszer e-mail küldő támogatását.²⁰

A JavaMail egy Java nyelven írt alkalmazásprogramozási felület, amivel e-maileket lehet küldeni és fogadni SMTP, POP3 és IMAP protokollokon keresztül.²¹

- **aws-java-sdk**

A AWS SDK egyszerűbbé teszi az Amazon Web Services használatát a Java-ban azáltal, hogy konzisztens és ismerős könyvtárakat kínál a Java-fejlesztők számára. Támogatást nyújt az API életciklusának mérlegeléséhez, például a hitelesítő adatok kezeléséhez, újrapróbálkozásokhoz, adatmegosztásokhoz és

¹⁶ <https://hu.wikipedia.org/wiki/JAX-B>

¹⁷ <https://mvnrepository.com/artifact/io.jsonwebtoken/jjwt>

¹⁸ <https://java.jsonwebtoken.io>

¹⁹ <https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml>

²⁰ <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-mail>

²¹ <https://hu.wikipedia.org/wiki/JavaMail>

sorosításokhoz. Az AWS SDK for Java a magasabb szintű absztrakciókat is támogatja az egyszerűbb fejlesztés érdekében.²²

- **spring-boot-starter-test**

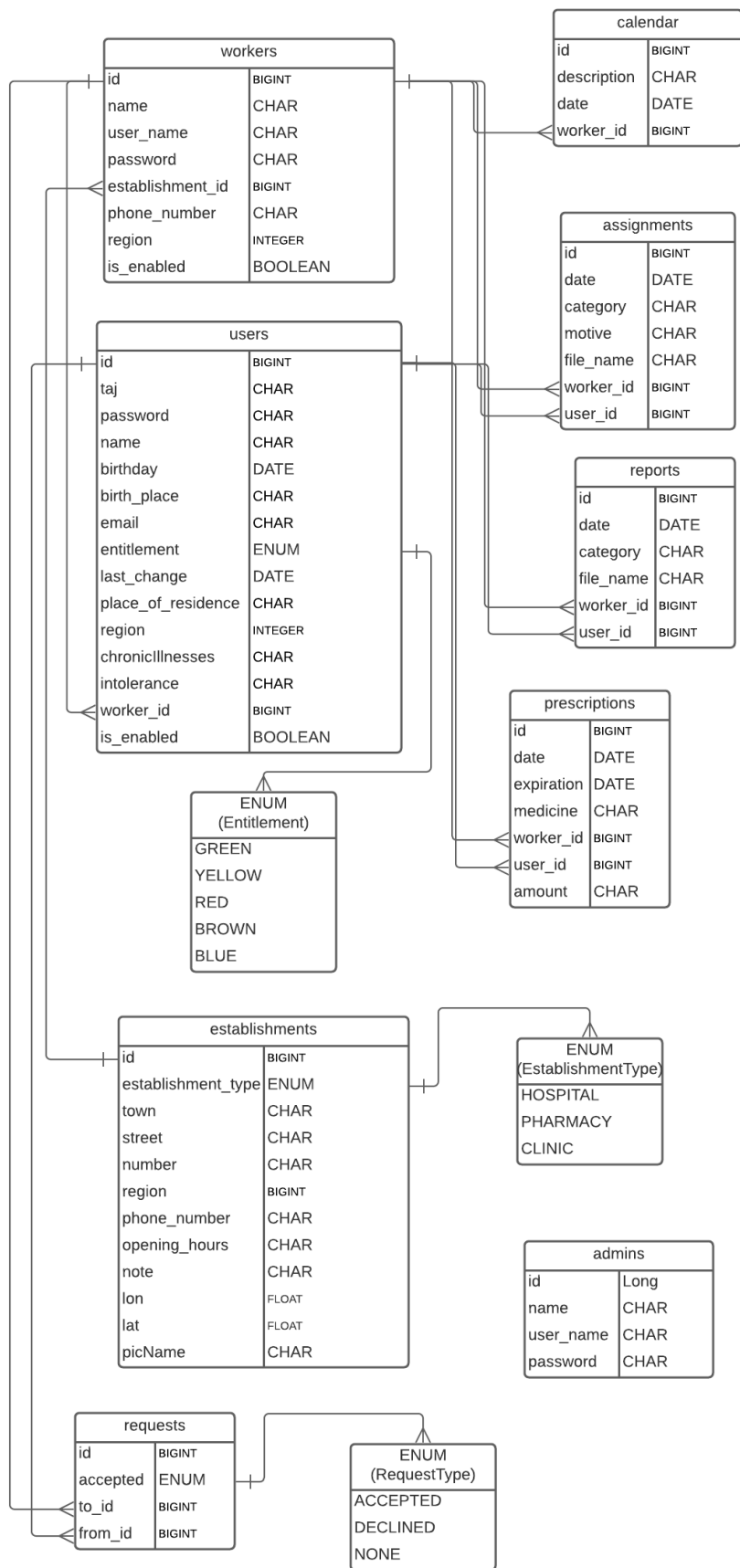
Kezdő csomag Spring Boot alkalmazások teszteléséhez ami olyan könyvtárakat tartalmaz mint a JUnit, a Hamcrest és a Mockito.²³

3.3. Adatbázis

Az használt adatbázis PostgreSQL, mely a Herokun keresztül lett létrehozva, és az Amazon tárolja. A Springnek köszönhetően azonban bármely más adatbázis is lehetne alapja az alkalmazásnak, csak a Drivert, a dialektust, és még pár beállítást kellene kicserélni a programban.

²² <https://aws.amazon.com/sdk-for-java/>

²³ <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-test>



27. ábra Adatbázis diagram

Az adatbázisban a leggyakoribb lekérdezések alapján indexeket hoztam létre.

Az adatbázisban található indexek:

- `idx_assignments_user`: Assignments tábla, `user_id` oszlop
- `idx_reports_user`: Reports tábla, `user_id` oszlop
- `idx_prescriptions_user`: Prescriptions tábla, `user_id` oszlop
- `idx_requests_user`: Requests tábla, `from_id` oszlop
- `idx_workers_region`: Workers tábla, `region` oszlop
- `idx_calendar_worker`: Calendar tábla, `worker_id` oszlop

3.4. Kliensoldal

A kliensoldalhoz Reactet használtam, ami egy nyílt forráskódú Javascript library.

A kliens a szerverrel kommunikál, onnan tölti le a megjelenítendő adatokat. Az oldalakat a `domain` mappában található struktúrák segítségével generálja le. Egy-egy komponenst többször is felhasznál.

Azonosítás, hitelesítés

A kérések headerjébe, az Authentication alá fűzi a `localStorage`-ba elmentett `jwt` token, melyet bejelentkezéskor ment le oda. Ezt a token-t kijelentkezésnél, `jwt` lejártakor, vagy az oldal bezárásakor (beletartozik a böngésző bezárása is) törli. Ez a token felel a felhasználó azonosításáért is.

3.4.1. Fejlesztői környezet

Visual Studio Code-ot használtam, és Prettier extention-t.

3.4.2. Projekt felépítése

A kezdő mappaszerkezetet a React alakítja ki. A `node_modules` mappa tartalmazza a függőségeket. A `public` mappa az `index.html`-t és a többi, közönség számára is elérhető fájlokat (például képeket) tartalmazza.

Az `src`-ben a `js` fájlok találhatóak. A `service` mappában olyan osztályok vannak megírva, melyeket ki lehetett csoportosítani, és valamilyen speciális területre vonatkozó műveleteket hajtanak végre. A `domain` olyan struktúrákat tartalmaz, melyek információt

tárolnak el a szerver oldali adatok felépítéséről, és melyek segítségével valamelyest automatizálható egyes adatok megjelenítése.

A *components* mappa a komponenseket tartalmazza, melyek további mappákra vannak szétbontva az átláthatóság kedvéért.

Mappák:

- ❖ node_modules
- ❖ public
 - assets
- ❖ src
 - components
 - calendar
 - documents
 - form
 - fields&modals
 - map
 - menu
 - profile
 - requests
 - universal
 - user
 - domain
 - services

3.4.3. Domain mappa

- **admin.js** (worker.js és user.js hasonlóképpen)

Admin specifikus adatokat tartalmaz. Admin menüjének adatait, admin jogosultságú kérések url-jeit.

- **classProps.js**

Osztályok leírását tartalmazza az oldalak generálásához épp leghatékonyabb formában (ezért lehet redundáns).

- **formProps.js**

Az űrlapok generálásához szükséges információkat tartalmazza. Minden osztályhoz, aminek van űrlapja, tartozik két js objektum. Egy konstans, ami az űrlap generálásához kell, és egy változó, mely egy üres objektum, az űrlap adatainak tárolásához.

3.4.4. Services mappa

- **auth_service.js**

Belépteti a felhasználót, elmenti az azonosításhoz szükséges adatokat a localStorage-ba, továbbá a felhasználónevet, nevet, szerepkört. Kilépteti a felhasználót, azzal hogy törli a localStorage-ból a lementett adatokat. Létrehozza a lekérésekhez szükséges Header-t a json web token-nel.

- **request_service.js**

Használja az auth_service.js-t, ahonnan lekéri a Header-t. A függvényei a lekérdezések CRUD típusa és Content-Type-ja szerint lettek létrehozva. A függvények adatokat várnak, amiket a body-ba raknak be, és requestOptions-t adnak vissza. A requestOptions egy objektum, és tartalmazza a Header-t, a body-t, és a lekérés metódusát is.

- **date_service.js**

A dátumok és időpontok megjelenítésében segít, és a lekérdezéseknek megfelelően (YYYY-MM-DDTHH:mm) is tudja formázni a dátumokat.

3.4.5. Komponensek

A komponensek gyakran a *domain*-ben található struktúrák segítségével dolgoznak. Azt az információt, hogy épp melyik struktúrára, adatra van szükségük, vagy az url-ből szerzik, vagy a szülő komponens adja át nekik. Ezzel a módszerrel a komponensek általánosíthatók, és újrafelhasználhatók válnak.

Például az összes dokumentumot mind a páciensnél, mind a felhasználónál ugyanaz a két komponens jeleníti meg, a *Document* és a *DocumentList*. Azt az információt hogy melyik dokumentum fajtát kell megjeleníteni, az url-ből szerzik, az

ehhez kapcsolódó struktúrát pedig a *domain* mappában található *classProps.js*-ből, a kérésekhez használt url-eket a *worker.js* és *user.js*-ből, a hozzájuk tartozó űrlap adatokat pedig a *formData.js*-ből kapják.

Az *App* komponens intézi a navigációt a *react-router-dom* komponensei segítségével. A *NavigationBar*-t, és a *Footer*-t mindig megjeleníti, a többi komponens, ami közvetlen alá van sorolva, az mind a tartalmat hivatott megjeleníteni, egy-egy „oldalt” kiteve.

Komponensek felépítése:

- App
 - NavigationBar
 - Login
 - MainPage
 - Map
 - MapModal
 - MapMarker
 - Menu
 - MenuElement
 - Profile
 - ProfileModal
 - PlaceOfResidence
 - ModifyButton
 - WorkerInfo
 - ModifyButton
 - ProfilePassword
 - ModifyButton
 - ModifyButton
 - Form
 - FormForFile
 - DisableProfile
 - FindUser

- DocumentList
 - Document
 - FormModal
- Calendar
 - EventModal
 - ConfirmModal
- Requests
- Footer

3.4.6. Függőségek

A programban több külső függőséget is használ. Ezek listája megtalálható a *package.json* fájlban a „dependencies” alcím alatt.

- "@fortawesome/fontawesome": "^1.1.8",
- "@fortawesome/fontawesome-free-solid": "^5.0.13",
- "@fortawesome/fontawesome-svg-core": "^1.2.30",
- "@fortawesome/free-solid-svg-icons": "^5.14.0",
- "@fortawesome/react-fontawesome": "^0.1.11",
- "@material-ui/core": "^4.11.0",
- "@testing-library/jest-dom": "^4.2.4",
- "@testing-library/react": "^9.5.0",
- "@testing-library/user-event": "^7.2.1",
- "leaflet": "^1.7.1",
- "react": "^16.13.1",
- "react-bootstrap": "^1.3.0",
- "react-calendar": "^3.1.0",
- "react-dom": "^16.13.1",
- "react-leaflet": "^2.7.0",
- "react-router-dom": "^5.2.0",
- "react-scripts": "3.4.3"

Párat kiemelnék közülük:

- **Leaflet**

Egy ingyenes, nyílt forráskódú JavaScript könyvtár, mely egy „csempézett” webes térképet ad hozzá programomhoz.²⁴

- **React-Calendar²⁵**

Nyílt forráskódú naptár React applikációkhoz.

- **Bootstrap**

Twitter által kezdeményezett nyílt forráskódú css könyvtár, mely reszponzív és modern külsőt kölcsönöz a webalkalmazásoknak.

- **Fontawesome**

A Font Awesome egy betűtípus és ikon eszköztár, amely CSS-en és Less-en alapul.²⁶

- **React Router Dom**

Az router lehetővé teszi az alkalmazás számára, hogy különböző komponensek között navigáljon, megváltoztassa a böngésző URL-jét, módosítsa a böngésző előzményeit és szinkronban tartsa a felhasználói felület állapotát.²⁷

²⁴ [https://en.wikipedia.org/wiki/Leaflet_\(software\)](https://en.wikipedia.org/wiki/Leaflet_(software))

²⁵ <https://www.npmjs.com/package/react-calendar>

²⁶ https://en.wikipedia.org/wiki/Font_Awesome

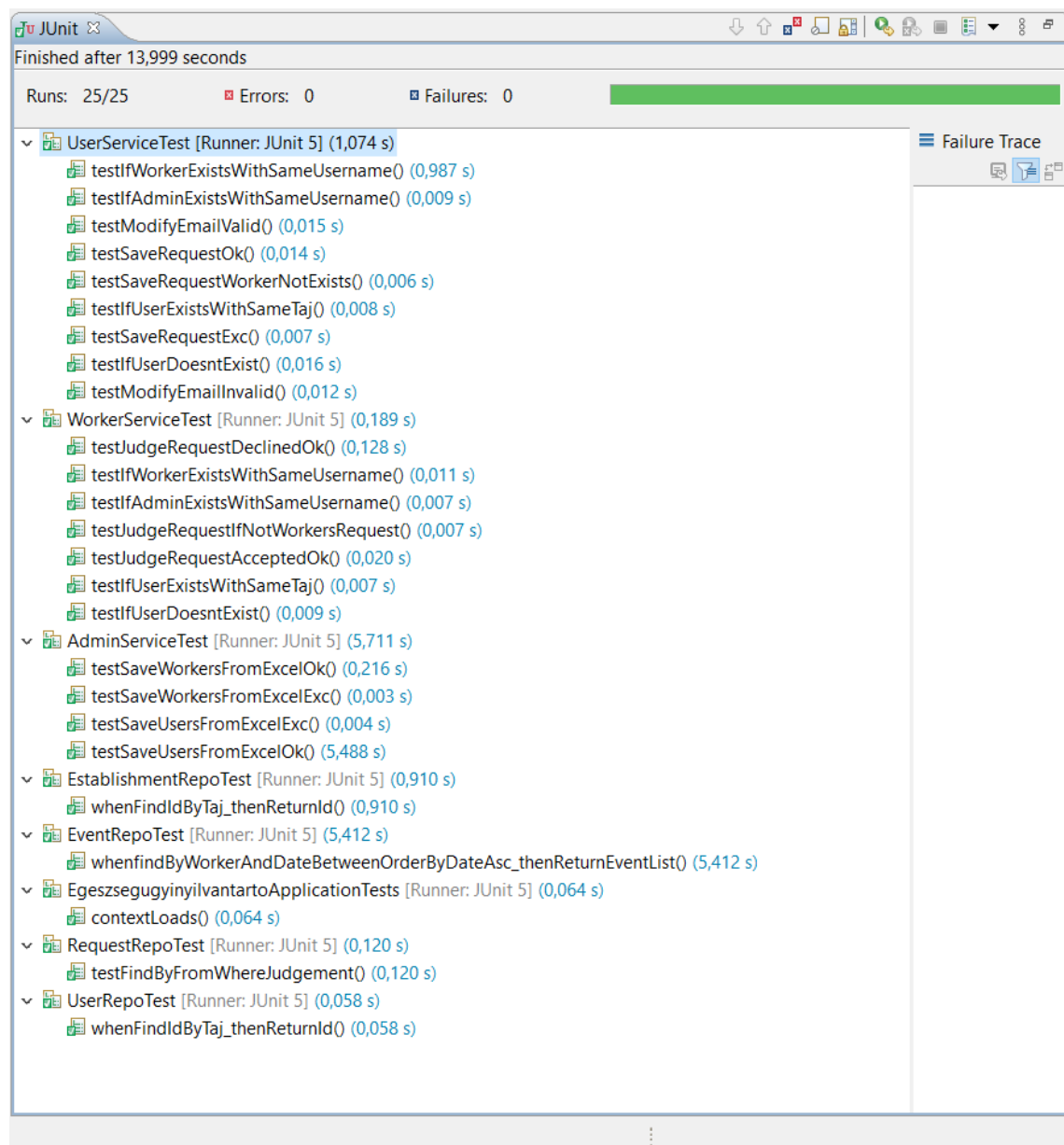
²⁷ <https://blog.logrocket.com/react-router-dom-set-up-essential-components-parameterized-routes-505dc93642f1/>

4. Tesztelés

4.1. Junit tesztelés

Az Repository és a Service osztályok kritikusabb és összetettebb metódusait teszteltem le unit tesztekkel. A Repository tesztekhez *DataJpaTest*-et használta, míg a Service-khez *Mockito*-t.

A tesztelést STS 4-ben futtattam le. Itt láthatók az eredmények:



28. ábra Unit tesztelés, STS képernyőkép

Letesztelt metódusok:

RequestRepo:

- findByFromAndJudgement – elfogadva, elutasítva

UserRepo:

- findByIdByTaj

EstablishmentRepo:

- findByLocation

EventRepo:

- findByWorkerAndDateBetweenOrderByDateAsc

AdminService:

- saveUsersFromExcel(MultipartFile)
 - van már ilyen felhasználónevű felhasználó a rendszerben
 - nem jó a fájl formátuma
 - minden rendben van

UserService:

- saveUser(User)
 - van már ilyen felhasználónevű felhasználó a rendszerben
 - nincs még ilyen felhasználónév a rendszerben
- saveUser(String, String, String, LocalDateTime, String, Entitlement, String, int)
 - van már ilyen felhasználónevű felhasználó a rendszerben
 - nincs még ilyen felhasználónév a rendszerben
- saveRequest(String, Long)
 - már küldött el kérvényt, amit még nem bíráltak el
 - nem létezik dolgozó ilyen id-vel
 - elküldhető a kérvény

- `modifyProfile(Long, String, String)`
 - invalid a megadott email cím
 - valid a megadott email cím

WorkerService:

- `saveWorker(Worker)`
 - van már ilyen felhasználónevű felhasználó a rendszerben
 - nincs még ilyen felhasználónév a rendszerben
- `saveWorker(String, String, String)`
 - van már ilyen felhasználónevű felhasználó a rendszerben
 - nincs még ilyen felhasználónév a rendszerben
- `judgeRequest(String, Long, RequestType)`
 - nem tudja elbírálni ha nem az ő kérvénye
 - elutasítás esetén nem változik meg a páciens legutolsó háziorvos váltásának dátuma, és a háziorvosa sem
 - elfogadás esetén megváltozik a páciens legutolsó háziorvos váltásának dátuma, és a háziorvosa a kérvény elbírálója (és birtokosa) lesz

4.2. Manuális tesztelés

Manuálisan teszteltem a UI-t és egypár olyan összetettebb funkciót, amit így volt hatékonyabb tesztelni.

Excel feltöltés

Az `ExcelService.class`-t és az őt használó `FormForFile.js` működésének tesztelése. (Csak a felhasználó feltöltését teszteltem, mivel a dolgozó feltöltése ugyanazon technikával működik.) Mindegyik tesztelésben megegyezik az elvárt a megtörténttel.

1. Teszt eset:

Nem excel típusú fájl feltöltése. (Jelen esetben word dokumentum.) Ekkor nem tölt fel a program semmit az adatbázisba.

Excel fájl feltöltés

A feltöltött fájl nem .xlsx formátumú, kérem adjon meg excel fájlt!

Excel fájl

Browse

Fájl feltöltése

29. ábra Tesztelés - "Nem xlsx kiterjesztésű a fájl" hibaüzenet

2. Teszteset:

Olyan excel fájl feltöltése, melynek fejléce nem egyezik meg az elvárttal. (Jelen esetben hiányzik a „NÉV” oszlop, és fejléc elem.)

	A	B	C	D	E	F	G	H
1	TAJ	JELSZÓ	SZÜLETÉSI DÁTUM	SZÜLETÉSI HELY	JOGOSULTSÁG	LAKHELY	KÖRZET	
2	32464667	pass	1996.03.02	Kiskunhalas	GREEN	Baja, Varsó utca, 12.	3	
3	12375678	pass	1975.10.24	Pécs	GREEN	Baja, Rókus utca, 1.	2	
4	54386263	pass	1980.07.04	Baja	YELLOW	Baja, Hunyadi János utca, 3.	1	
5	53214572	pass	n	Baja	GREEN	Baja, Szarka utca, 5.	1	
6	27859623	pass	1984.11.05	Kiskunhalas	GREEN	Baja, Róka utca, 10.	1	
7	98368245	pass	1969.05.14	Szeged	RED	Baja, Határ utca, 21.	2	
8	93677231	pass	1970.08.09	Kecskemét	GREEN	Baja, Budai Nagy Antal utca, 2.	3	
9								

30. ábra Teszt - Hibás fejlécű excel fájl

Feltöltés után meg is kapjuk a hibaüzenetet, benne az elvárt fejléccel. Ekkor nem tölt fel a program semmit az adatbázisba.

Excel fájl feltöltés

A fájlban nem helyes a fejléc! Az elvárt formátum:
TAJ, JELSZÓ, NÉV, SZÜLETÉSI DÁTUM, SZÜLETÉSI
HELY, JOGOSULTSÁG, LAKHELY, KÖRZET

31. ábra Teszt - "Nem helyes a fejléc" hibaüzenet

3. Teszteset:

Olyan excel fájl feltöltése, melyben szerepel olyan adat, ami nem az elvárt formátumban van. (Jelen esetben dátum helyett szöveg szerepel a „SZÜLETÉSI DÁTUM” oszlopában, az 5. sorban.)

	A	B	C	D	E	F	G	H
1	TAJ	JELSZÓ	NÉV	SZÜLETÉSI DÁTUM	SZÜLETÉSI HELY	JOGOSULTSÁG	LAKHELY	KÖRZET
2	32464667	pass	Kiss Anikó	1996.03.02	Kiskunhalas	GREEN	Baja, Var	3
3	12375678	pass	Nagy Julianna	1975.10.24	Pécs	GREEN	Baja, Ró	2
4	54386263	pass	Fekete Gábor	1980.07.04	Baja	YELLOW	Baja, Hu	1
5	53214572	pass	Horváth Zoltán	n	Baja	GREEN	Baja, Sza	1
6	27859623	pass	Gál Judit	1984.11.05	Kiskunhalas	GREEN	Baja, Ró	1
7	98368245	pass	Horváth Henriett	1969.05.14	Szeged	RED	Baja, Ha	2
8	93677231	pass	Tóth Gabriella	1970.08.09	Kecskemét	GREEN	Baja, Buc	3

32. ábra Teszt - Rossz formátumú cella az excelben

Ekkor a hibaüzenetben megjelenik a hibát kiváltó adat sorszáma, és a program nem ment le egy felhasználót sem az adatbázisba.

Excel fájl feltöltés

Hiba történt a fájl beolvasása közben, helytelen formátum; ellenőrizze a 5. sort!

33. ábra Teszt - "Rossz formátum az x. sorban" hibaüzenet

4. Teszt eset:

Olyan excel fájl feltöltése, melyben szerepel olyan felhasználót azonosító adat (felhasználónév, vagy TAJ szám), ami már szerepel az adatbázisban. (Jelen esetben ilyen a „32464667” mivel az excel beolvasása közben az első sorban szereplő felhasználó lementődik, és mire a harmadik felhasználóra kerülne a sor, azt fogja találni a program, hogy már létezik ilyen felhasználó.)

	A	B	C	D	E	F	G	H
1	TAJ	JELSZÓ	NÉV	SZÜLETÉSI DÁTUM	SZÜLETÉSI HELY	JOGOSULTSÁG	LAKHELY	KÖRZET
2	32464667	pass	Kiss Anikó	1996.03.02	Kiskunhalas	GREEN	Baja, Varsó utca, 12.	3
3	12375678	pass	Nagy Julianna	1975.10.24	Pécs	GREEN	Baja, Rókus utca, 1.	2
4	32464667	pass	Fekete Gábor	1980.07.04	Baja	YELLOW	Baja, Hunyadi János utca, 3.	1
5	53214572	pass	Horváth Zoltán	1984.03.12	Baja	GREEN	Baja, Szarka utca, 5.	1
6	27859623	pass	Gál Judit	1984.11.05	Kiskunhalas	GREEN	Baja, Róka utca, 10.	1
7	98368245	pass	Horváth Henriett	1969.05.14	Szeged	RED	Baja, Határ utca, 21.	2
8	93677231	pass	Tóth Gabriella	1970.08.09	Kecskemét	GREEN	Baja, Budai Nagy Antal utca, 2.	3

34. ábra Teszt - létező felhasználónév az excelben

Hibaüzenet jelenik meg a már létező, és még egyszer felvinni kívánt felhasználó nevével.

Excel fájl feltöltés

Már létezik felhasználó ilyen felhasználónévvel!
(32464667)

Excel fájl

Browse

Fájl feltöltése

35. ábra Teszt - "Létező felhasználónév" hibaüzenet

Az adatbázisba csak a második felhasználóig mentődnek le az adatok. (Jelen esetben csak a 32464667, és a 12375678 TAJ számú felhasználó mentődik le, az előző 3 már a beolvasás előtt szerepelt az adatbázisban.)

HANGE	NAME	PASSWORD	PLACE_OF_RESIDENCE	REGION	TAJ	WORKER_ID
	Horváth Andrea	\$2a\$10\$FFYUU2AhkmNzTV1WhwNhZ.7RdVzac5A6yJelSMLEKyAGkXsl6iFG	Baja, Kossuth utca 10.	2	12345678	1
	Horváth Zoltán	\$2a\$10\$8CRAGAbi3juqLP9I4djOVe14.ZZZTWE5bn6UX/tNxNgr7wLJEdFuy	Baja, Csurfusz utca 25.	2	12345679	1
	Kiss Pista	\$2a\$10\$nzhuJ/Rrls66i.wxxwCJwmOoTacPXVXYI5ioMrstN78YAnomcWTGq.	Baja, Varsó utca 8.	4	11111111	null
	Kiss Anikó	\$2a\$10\$qu11oBzDgz7b6pf94ZxSOu393JGyrw/.6EgKHrEN5tmdEZdVGInq	Baja, Varsó utca, 12.	3	32464667	null
	Nagy Julianna	\$2a\$10\$9fibaBiwDezQRk8o2Gq9u8.2wunKwTMBclvbhkdM/NTjaxz460DG	Baja, Rókus utca, 1.	2	12375678	null

36. ábra Teszt - Létező felhasználónév esetén az adatbázis

5. Teszteset:

Helyes excel fájl feltöltése.

	A	B	C	D	E	F	G	H
1	TAJ	JELSZÓ	NÉV	SZÜLETÉSI DÁTUM	SZÜLETÉSI HELY	JOGOSULTSÁG	LAKHELY	KÖRZET
2	87654321	pass	Nagy Anikó	1996.03.02	Kiskunhalas	GREEN	Baja, Varsó utca, 12.	3
3	32464667	pass	Halas Iván	1975.10.24	Pécs	GREEN	Baja, Rókus utca, 1.	2
4	54386263	pass	Barna Feren	1959.12.27	Baja	YELLOW	Baja, Hunyadi János utca, 3.	1
5	53214572	pass	Kiss József	1980.07.04	Baja	GREEN	Baja, Szarka utca, 5.	1
6	27859623	pass	Török Gábo	1984.11.05	Kiskunhalas	GREEN	Baja, Róka utca, 10.	1
7	98368245	pass	Gál Abigél	1969.05.14	Szeged	RED	Baja, Határ utca, 21.	2
8	93677231	pass	Fodor Berta	1970.08.09	Kecskemét	GREEN	Baja, Budai Nagy Antal utca, 2.	3

37. ábra Teszt - helyesen kitöltött excel fájl



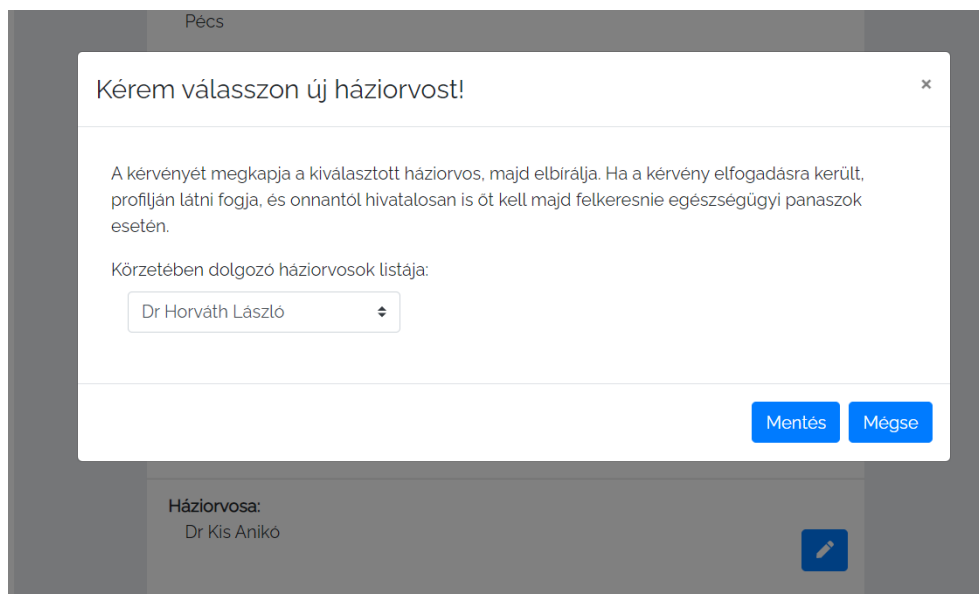
38. ábra Teszt - "Sikeres fájlfeltöltés"

NAME	PASSWORD	PLACE_OF_RESIDENCE	REGION	TAJ	WORK
Horváth Andrea	\$2a\$10\$Ey3Gbm1BbVo24h45wBK1puq6a80dHEn4uZgJNeFWGpP3W5xPv3fcC	Baja, Kossuth utca 10.	2	12345678	1
Horváth Zoltán	\$2a\$10\$Fuogc9uEn3.9WLFfKFmIA.nGa8VhvHMjDsp/vd/I3O1nHBgGQkYuu	Baja, Círfusz utca 25.	2	12345679	1
Kiss Pista	\$2a\$10\$MMKx5xRTbUt2HTg.U95L0On.nKTvJVsyUvTlhq9dfFMEFgMbP7kiO	Baja, Varsó utca 8.	4	11111111	null
Nagy Anikó	\$2a\$10\$WFGTjDhOawrfZ2iQd6Ahx.q2r6U2KedX9t3RUU2AkX1GsePk6CRuW	Baja, Varsó utca, 12.	3	87654321	null
Halas Iván	\$2a\$10\$CGh33RnBW14TeF5xBbSgrOs2dnQvVKy7yT9cAV1WQoqHYc0ge1MSO	Baja, Rókus utca, 1.	2	32464667	null
Barna Ferenc	\$2a\$10\$scI.qIXVmyT9w7wWqi6xkej5TAH3PVg/BmH56POejmu6vBE5o2Wdy	Baja, Hunyadi János utca, 3.	1	54386263	null
Kiss József	\$2a\$10\$uzQ9KhzfJfbk/gsCoo9qTekqZTnAgwnwVlxubrvnpv0S3Wc74d5Da	Baja, Szarka utca, 5.	1	53214572	null
Török Gábor	\$2a\$10\$I.bB1v9T62pE.GLVHWTQZeMUm3.Bcf8dZeuLzb524/E5OA/RwD.JG	Baja, Róka utca, 10.	1	27859623	null
Gál Abigél	\$2a\$10\$4TKIAvvrXp58iUsQm.mo8OvQ78sBMxa4VadPLTKN3L5T1UwtpdUNi	Baja, Határ utca, 21.	2	98368245	null
Fodor Bertalan	\$2a\$10\$SfpvUA16YtDKTT9/grjbPua4I3a3JFzL0xXeibxFjw5butIn0frV6	Baja, Budai Nagy Antal utca, 2.	3	93677231	null

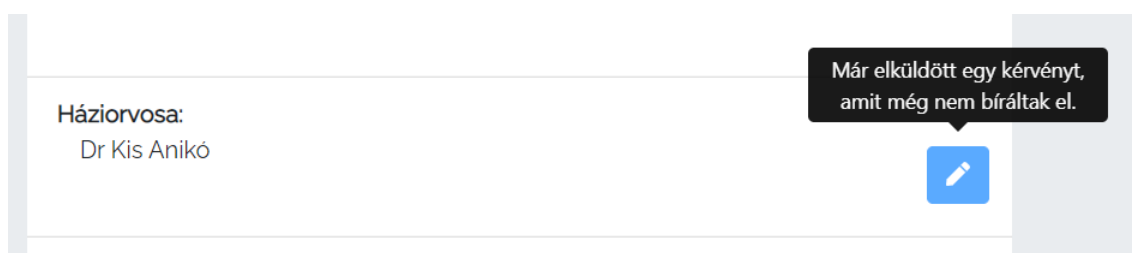
39. ábra Teszt - Adatbázis sikeres feltöltés után

Háziorvos váltás

Először a páciens profiljába lépünk be, ahol kiválasztunk egy háziorvost a listából (ahol az aktuális nem szerepel). Ezután átlépünk a háziorvos profiljába, ahol elfogadjuk a kérvényt. Ekkor a páciens email címére megérkezik a változásról tájékoztató email, és megváltozik a profiljában a háziorvos neve, és az utolsó váltás dátuma.



40. ábra Teszt - Új háziorvos választása



41. ábra Teszt - "Már elküldött egy kérvényt"



42. ábra Teszt - Új elbírátlan kérvény

Kérvények		
Háziorvos váltási kérelem	Kiss Pista	<div>✓</div> <div>✗</div>
Elbíralt kérvények		
Háziorvos váltási kérelem	Horváth Zoltán	elutasítva
Háziorvos váltási kérelem	Horváth Andrea	elfogadva

43. ábra Teszt - Kérvény elfogadva



44. ábra Teszt - Email a nyilvántartótól

Háziorvosa:

Dr Horváth László

Utolsó házi orvos váltás dátuma:

2020.11.26.

Még nem válthat házi orvost, mert még nem telt le a 1 év a legutóbbi váltása óta.

45. ábra Teszt - Háziorvos megváltozása

5. Továbbfejlesztési lehetőségek

Ez a program csak egy kis szelete egy sokkal nagyobb programnak. Örülök hogy ennyit sikerült megvalósítanom belőle, de még sok lehetőség van benne, melyek egy kisebb részét eredetileg is bele szerettem volna rakni, másik jóval nagyobb része viszont a program írása közben jutott eszembe. Az okok amik ebben megakadályoztak: nem találtam ingyenes eszközt hozzá, és/vagy az időm nem engedte meg.

Az ötleteim közül itt meg is neveznék párat, mely sokat javítana a program használhatóságán:

- Felhasználóbarát telepítőt készíteni a programhoz a könnyebb üzembe helyezés érdekében.
- A weboldal reszponzívvá tétele, hogy más felbontású eszközökön is megjeleníthető legyen.
- A dokumentumok kórképek szerinti csoportosítása, megjelenítése. Ezzel könnyebben követhetők lennének a páciensek bizonyos betegségei és azok kezelése.
- Vértételek eredményeinek adatbázisban való tárolása. Ezáltal diagramon lehetne követni a páciensek vérképének változását, ami átláthatóbbá tenné az adatokat, és könnyebben észrevehetővé a kiemelkedően eltérő értékeket.
- Egészségügyi dolgozók megkülönböztetése (fogorvos, általános orvos, stb), speciális funkciók bevezetése hivatásuk alapján.
- Dokumentum konvertáló: Mivel annak nem sok értelme van, hogy a páciensek például excel vagy word formátumban tudják letölteni a feltöltött leleteiket vagy beutalóikat a rendszerből, és mivel az egészségügyi dolgozóknak sok munkájuk van amellet, hogy minden egyes dokumentumot maguk konvertáljanak pdf-fé, melyeket fel szeretnének tölteni, így érdemes lenne a rendszerbe beleépíteni egy dokumentum konvertálót, mely ezt a műveletet automatikusan elvégzi. Ingyenes könyvtárat sajnos egyelőre nem találtam hozzá.
- Földrajzi koordináta megállapítása, kiszámolása: Próbáltam lehetőséget találni arra, hogy pontos cím alapján a program saját maga be tudja állítani az intézmények földrajzi koordinátáit, de erre sajnos még nincs ingyenes,

megbízható lehetőség. Elég kellemetlen ugyanis kikeresgélni minden egyes felvitelhez a koordinátákat manuálisan, és természetesen ez nem várható el az adminoktól.

- Szerveroldali cache-elés bevezetésével, például brute force elleni védelmet is lehetne biztosítani, mely megakadályozná a jelszavak feltörését, és az azzal járó illetéktelen profil használatot.
- A program biztonsági szintjének emelése, több lépcsős belépés bevezetése.
- Elegendő adat összegyűjtése után orvosi statisztikák kiállítására is lehetne használni a programot, megelőző felkészítés után.

6. Összegzés

A szakdolgozat megírása előtt még úgy terveztem, hogy beszélek majd egészségügyi dolgozókkal, akiket meghallgatva használható és hasznos programot készíthetek majd. Azonban az idei járvány megakadályozott eme törekvésemben, és saját elképzeléseim alapján kellett megírnom a szakdolgozatomat, melyek sajnos egészségügyi szempontból kissé hiányosak voltak. Ennek ellenére büszke vagyok a munkámra, ha nem is egészségügyi szempontból, de bizonyos implementációk miatt, és habár nem is vagyok mindennel teljesen elégedett, azt mondhatom, hogy nagyon sokat fejlődtem mialatt elkészítettem ezt a projektet. Rengeteg új dolgot tanultam, sok új technológiát megismertem - még ha nem is professzionális mélységben - és ezt a tudást szeretném gyarapítani a jövőben is.

Köszönetet szeretnék mondani a konzulensemnek, Szalai-Gindl Jánosnak, aki nagyon sokat segített nekem a szakmai precizitásával és türelmével, és aki nélkül ez a szakdolgozatot koránt sem tudtam volna ilyen minőségűre elkészíteni. Köszönöm a barátaimnak is a lelki támogatást a bosszús hibakeresős időkben, és ezt a pár gyönyörű együtt töltött évet!

7. Ábrajegyzék

1. ábra Bejelentkezés felület.....	12
2. ábra Térkép	13
3. ábra Intézmény adatai	13
4. ábra Felhasználó menüjének részlete	14
5. ábra Felhasználó "Receptek" oldala	15
6. ábra Felhasználó "Beutalók" oldala.....	16
7. ábra Felhasználó "Leletek" oldala	16
8. ábra Felhasználó "Profil" oldalának részlete	16
9. ábra Jelszó változtatás hibaüzenettel.....	17
10. ábra Új házi orvos választása	18
11. ábra Inaktív módosítás, házi orvos váltás után	18
12. ábra Dolgozó menüjének részlete	19
13. ábra "Kérvények" oldal	19
14. ábra "Naptár" oldal.....	20
15. ábra Esemény felvitele.....	21
16. ábra "Páciens keresése" oldal, hibaüzenettel.....	21
17. ábra Plusz gombok a páciens adatlapján	22
18. ábra Új dokumentum.....	22
19. ábra Admin menüjének részlete	23
20. ábra Admin űrlapjai	24
21. ábra Felhasználói profil zárolása	25
22. ábra Felhasználóhoz tartozó use case diagram	29
23. ábra Dolgozóhoz tartozó use case diagram	30
24. ábra Adminhoz tartozó use case diagram	31
25. ábra Spring program működése.....	46
26. ábra Spring programban a Security működése	57
27. ábra Adatbázis diagram	62
28. ábra Unit tesztelés, STS képernyőkép	69
29. ábra Tesztelés - "Nem xlsx kiterjesztésű a fájl" hibaüzenet	72
30. ábra Teszt - Hibás fejlécű excel fájl.....	72
31. ábra Teszt - "Nem helyes a fejléc" hibaüzenet	73

32. ábra Teszt - Rossz formátumú cella az excelben.....	73
33. ábra Teszt - "Rossz formátum az x. sorban" hibaüzenet	74
34. ábra Teszt - létező felhasználónév az excelben	74
35. ábra Teszt - "Létező felhasználónév" hibaüzenet	75
36. ábra Teszt - Létező felhasználónév esetén az adatbázis	75
37. ábra Teszt - helyesen kitöltött excel fájl.....	75
38. ábra Teszt - "Sikeres fájlfeltöltés"	76
39. ábra Teszt - Adatbázis sikeres feltöltés után	76
40. ábra Teszt - Új házi orvos választása	77
41. ábra Teszt - "Már elküldött egy kérvényt"	77
42. ábra Teszt - Új elbírálatlan kérvény.....	77
43. ábra Teszt - Kérvény elfogadva	78
44. ábra Teszt - Email a nyilvántartótól.....	78
45. ábra Teszt - Házi orvos megváltozása.....	78

8. Források

- [1] <https://nodejs.org/en/download/>
Elérés dátuma: 2020.11.26.
- [2] <https://www.baeldung.com/spring-boot-war-tomcat-deploy>
Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)
- [3] <http://mirror.23media.de/apache/tomcat/tomcat-8/v8.5.30/bin/>
Elérés dátuma: 2020.11.26.
- [4] <https://dev.to/crishanks/deploy-host-your-react-app-with-cpanel-in-under-5-minutes-4mf6>
Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)
- [5] EPAM - ELTE_JPA_Spring_Data.pdf
Elérés dátuma: 2020.11.26. (referencia ábra)
- [6] http://www.neak.gov.hu/felso_menu/lakossagnak/ellatas_magyarorszagon/jogosultsag_az_ellatasra/ellatasra_jogosultsag_igazolasa/jogviszony_rendezeesi_eljaras
Elérés dátuma: 2020.11.26.
- [7] <https://youtu.be/CsgtYvIR7xk>
Elérés dátuma: 2020.11.26. (videóban szereplő ábrán alapul)
- [8] <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-web>
Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)
- [9] <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-data-jpa>
Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)
- [10] <https://www.javatpoint.com/spring-boot-starter-data-jpa>
Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)

- [11] <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-jdbc>
Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)
- [12] https://hu.wikipedia.org/wiki/Java_Database_Connectivity
Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)
- [13] <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-security>
Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)
- [14] <https://spring.io/projects/spring-security>
Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)
- [15] <https://docs.spring.io/spring-security/site/docs/4.2.x/reference/html/test-method.html>
Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)
- [16] <https://hu.wikipedia.org/wiki/JAX-B>
Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)
- [17] <https://mvnrepository.com/artifact/io.jsonwebtoken/jjwt>
Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)
- [18] <https://java.jsonwebtoken.io>
Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)
- [19] <https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml>
Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)
- [20] <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-mail>
Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)
- [21] <https://hu.wikipedia.org/wiki/JavaMail>
Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)
- [22] <https://aws.amazon.com/sdk-for-java/>

Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)

- [23] <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-test>

Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)

- [24] [https://en.wikipedia.org/wiki/Leaflet_\(software\)](https://en.wikipedia.org/wiki/Leaflet_(software))

Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)

- [25] <https://www.npmjs.com/package/react-calendar>

Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)

- [26] https://en.wikipedia.org/wiki/Font_Awesome

Elérés dátuma: 2020.11.26. (angol nyelvről lefordítva)

- [27] <https://blog.logrocket.com/react-router-dom-set-up-essential-components-parameterized-routes-505dc93642f1/>

Elérés dátuma: 2020.11.28. (angol nyelvről lefordítva)

A diagramokat online diagram tervezőkben csináltam:

- [1] <https://app.lucidchart.com/>

Elérés dátuma: 2020.12.02. (adatbázis diagram, spring app felépítés, security árba)

- [2] <https://online.visual-paradigm.com/>

Elérés dátuma: 2020.12.02. (use case diagram)

8.1. Felhasznált képek

A programomban ingyenesen felhasznált képek címei:

- [1] <https://pixabay.com>

Elérés dátuma: 2020.10.20.

(Pixabay License: Kereskedelmi felhasználásra ingyenes, nem szükséges feltüntetni a tulajdonságokat)

9. Mellékletek

- Témabejelentő
(*Habzda_Fruzsina_XNUHTE_Témabejelentő.pdf*)
- Eredetiségnyilatkozat
(*Habzda_Fruzsina_XNUHTE_Eredetiségi_nyilatkozat.pdf*)
- Edit űrlap
(*Habzda_Fruzsina_XNUHTE_EDIT_Hallgatói_dolgozat_űrlap.pdf*)
- Edit nyilatkozat
(*Habzda_Fruzsina_XNUHTE_EDIT_nyilatkozat_hallgatoidolgozat.pdf*)
- A projektet tartalmazó mappa
(*egeszsegugyinyilvantarto*)