

EÖTVÖS LORÁND TUDOMÁNYEGYETEM

Informatikai Kar

Információs Rendszerek Tanszék

Munkaidő és feladat nyilvántartó rendszer

Témavezető:

Dr. Nikovits Tibor

mesteroktató, matematikus

Készítette:

Kotroczó Roland

programtervező informatikus BSc

Budapest, 2020

SZAKDOLGOZAT

EREDETISÉG NYILATKOZAT

Alulírott **Kotroczó Roland** Neptun-kód: **EA2EEB**

ezennel kijelentem és aláírással megerősítem, hogy az Eötvös Loránd Tudományegyetem Informatikai Karának, **Információs Rendszerek** Tanszékén írt,

Munkaidő és feladat nyilvántartó rendszer című szakdolgozatom/diplomamunkám saját, önálló szellemi termékem; az abban hivatkozott szakirodalom felhasználása a szerzői jogok általános szabályainak megfelelően történt.

Tudomásul veszem, hogy szakdolgozat/diplomamunka esetén plágiumnak számít:

- szószerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

Budapest, 2020. 05. 29.



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

SZAKDOLGOZAT-TÉMA BEJELENTŐ

Név: **Kotroczó Roland**

Neptun kód: **EA2EEB**

Tagozat: **nappali**

Szak: **programtervező informatikus BSc**

Témavezető neve: **Dr. Nikovits Tibor**

munkahelyének neve és címe: **ELTE IK Információs Rendszerek Tanszék
1117 Budapest, Pázmány Péter sétány 1/C.**

beosztása és iskolai végzettsége: **mesteroktató, matematikus**

A dolgozat címe: **Munkaidő és feladat nyilvántartó rendszer**

A dolgozat témája:

A szakdolgozatom témája egy vállalatok számára használható munkaidő-nyilvántartó rendszer, amely kezeli az alkalmazottak személyes adatait, munkaidő adatait, a program továbbá tartalmaz egy plusz modult, ami egy feladatkezelő.

A program 2 fajta bejelentkezési formával fog rendelkezni. Az első esetén egy felhasználónév és jelszó segítségével lehet használni a programot. A második esetben az alkalmazott kiválaszt egy tevékenységet, megadja az azonosítóját, és a rendszer bejelentkeztetés nélkül regisztrálja a tevékenységet. Ilyenkor a felhasználó számára más funkció nem elérhető.

A felhasználónévvel bejelentkezett személyek tetszőleges tevékenység rögzítését elvégezhetik a rendszerben. A tevékenységek kategóriákba rendezhetők. Egy tevékenység esetén meg kell adni a ráfordított időt és opcionálisan megjegyzés is megadható hozzá.

A rendszer rendelkezni fog egy szabadság és időbeosztás tervezővel, ahol a megfelelő jogosultsággal rendelkező felhasználó egyedileg megtervezheti a hozzárendelt személyek beosztásait. A programnak részét képezi egy feladat ütemező modul is, amelyben a felhasználók saját maguk vagy mások számára feladatokat vagy feljegyzéseket adhatnak meg.

A rendszerben szereplő adatok különböző szempontok szerint lekérdezhetők, kereshetők lesznek, valamint az eredményeket excel formátumban lehet majd exportálni.

A rendszer megvalósítására webes technológiát fogok használni. A szerveroldali programok elkészítése Java Spring Boot technológiával és MVC modellben történik, a kliensoldali megvalósítást pedig Angular keretrendszerrel végzem el.

TARTALOMJEGYZÉK

BEVEZETÉS.....	6
Motiváció	7
FELHASZNÁLÓI DOKUMENTÁCIÓ	8
A program telepítése	8
A program felépítése.....	8
A program funkcionális leírása.....	9
Bejelentkezés:	9
Munkaidő-nyilvántartó:	9
Szabadság tervező:.....	12
Időbeosztás tervező:	13
Listázás:	15
Felhasználók – Karbantartás:	16
Projektek – Karbantartás:	18
Feladatok – Karbantartás:	20
Jogosultságok – Karbantartás:	23
Feladatkezelő	25
FEJLESZTŐI DOKUMENTÁCIÓ	28
Felhasznált technológiák.....	28
Java.....	28
Spring	28
JPA	28
Lombok.....	28
MySQL	29
Node.js	29
Angular	29
HTML	29
CSS.....	29
JavaScript	30
Typescript.....	30
Bootstrap.....	30
Felhasználói esetek	30
A szerveroldal (Backend) felépítése.....	31
A program felépítése:.....	32
Az excel fájl elkészítése:	38

A kliensoldal (Frontend) felépítése	38
A program felépítése.....	38
Az adatbázis felépítése.....	40
Adatbázis terv:	40
Tesztelés.....	41
Manuális tesztelés:.....	42
ÖSSZEFOGLALÁS.....	47
ÁBRAJEGYZÉK	48
HIVATKOZÁSOK	49

BEVEZETÉS

A mai digitalizált világunkban megoldást kell találnunk mindennapjainkat érintő kérdésekre, melyek leegyszerűsítenek bizonyos folyamatokat. Az adminisztrációs feladatok digitálisan történő elvégzése még fontosabb, mert nem vesztegetünk annyi felesleges időt és energiát. Az említett előnyök mellett nem mehetünk el a környezetvédelem mellett sem, mert elektronikus adminisztrációval rengeteg papírt tudunk megspórolni, és ezzel óvjuk környezetünket.

A dolgozatom elkészítése folyamán egy olyan program tervezésén és megalkotásán dolgoztam, ami kezeli egy vállalkozás alkalmazottainak munkaidő és szabadság nyilvántartását. Az említett funkciókon kívül más szolgáltatásokat is tartalmaz, ilyen például a feladatkezelő és időbeosztás tervező modul. Az említett funkciók sok segítséget nyújthatnak egy vezetőnek abban, hogy nyomon kövesse beosztottjainak előrehaladását egy adott munkafolyamatnál, vagy esetleg az új feladatok kiosztásánál.

A program rendelkezik egy egyszerűsített nyilvántartó rendszerrel is. Ennek a modulnak a lényege, hogy ha egy felhasználó azonosítani tudja magát egy felhasználói névvel, akkor egy kattintással rögzítheti a saját tevékenységét az adatbázisba. Előfordulhatnak olyan munkaterületek, ahol elegendő pl. a munkavállaló be és kilépését rögzíteni, de tetszőlegesen megadható bármilyen tevékenység. Ilyenkor elegendő ezen funkció használata.

Azok közül, akik csak felhasználói szinten tudják használni a számítógépet, sok olyan ember akad, aki ismeri és sokat használja a Microsoft Office táblázatkezelőjét (Excel). Számukra sok segítséget nyújthat az, hogy az adatbázisba berögzített adatokat könnyedén ki lehet exportálni Excelbe. Ezzel használójának megkönnyítheti életét, ha pl. egyedi statisztikát szeretne készíteni a felhasználók által rögzített adatokból.

Manapság egy programnál elvárás lehet, hogy bárhonnan, bármikor elérhető legyen. Éppen ezért, webes technológiával készült, és ha használója rendelkezik megfelelő web szolgáltatással, akkor könnyedén elérhető és használható, akár okos telefon segítségével is.

Motiváció

Szakedolgozatom témájának egy olyan feladatot szerettem volna, ahol egy olyan programot kell megalkotni, ami sok segítséget nyújthat használóinak. Sokat töprengtem, aztán kitaláltam, hogy egy olyan munkaidő nyilvántartó és feladatkezelő rendszert készítek, ami kis- és középvállalatok számára alkalmas.

Tudjuk, hogy egy vállalat irányítása nem egyszerű, főleg, ha sok emberrel kell együtt dolgozni. Korábbi munkatapasztalataim alapján tudom, hogy egy vezetőnek nagyon fontos, hogy teljes mértékben tudja egy adott munkafolyamatra a ráfordított időt. Ezen ismeretek nagy mértékben befolyásolhatják a vállalat gazdasági döntéseit, de akár szerepet játszhatnak újabb erőforrások bevonásában is.

A már említett tényezők mellett fontosnak tartottam, hogy olyan technológiát használjak szakedolgozatom elkészítése folyamán, ami megfelel a hétköznapi elvárásoknak. Az utóbbi időben a szoftverek fejlesztése elmozdult a webes technológiák irányába. Rohanó világunkban sokszor előfordulhat, hogy folyamatosan változik az a hely, ahonnan dolgoznunk kell. Ennek következtében manapság mindenki rendelkezik okostelefonnal, vagy olyan hordozható eszközzel, ami képes csatlakozni a világhálóra. Ezen tényeket számításba véve úgy döntöttem, hogy sokkal célravezetőbb, ha a programom megfelel ezeknek a követelményeknek. Sokat gondolkoztam azon, hogy milyen programozási nyelvet használjak, és a választásom így esett a szerveroldali Java Spring boot technológiára és a kliensoldali Angular keretrendszerre. Minden program egyik legfontosabb követelménye az, hogy teljes biztonságot nyújtson használóinak az adataik védelmét illetően. A biztonság mellett meg kell említenem azt is, hogy egy programnak letisztultnak és könnyen kezelhetőnek kell lenni. Ezek a technológiák megfelelnek mind a felhasználói élmény maximális elérésére, mind a legnagyobb biztonság kialakítására.

Összegezve a fő célom elkészíteni egy olyan programot, ami megfelel a fentebb említett elvárásoknak és kiszolgálja a használó igényeit.

FELHASZNÁLÓI DOKUMENTÁCIÓ

A program telepítése

A program telepítése előtt a számítógépünkre telepíteni kell a következő programokat:

- Java 8 vagy újabb verziója
- Node.js 12.16.3 vagy újabb verziója
- PuTTY
- XAMPP
- Angular keretrendszer. Ennek a telepítésére a program tartalmaz egy „Angular_install.bat” állományt

Az adatbázis kialakításához készítettem egy „import.sql” scriptet. A XAMPP program első indítása utána a MySQL adatbázis rendszerbe be kell importálni a fentebb említett scriptet.

Miután minden programot telepítettünk a rendszerünkre és elvégeztük az első indításhoz szükséges folyamatokat, nincs más dolgunk, mint a „start.bat” állománnyal indítani az alkalmazást. Ezek után a böngészőnkbe beírjuk a „localhost:4200” címet, és már használhatjuk is a programot. A program először az admin/admin felhasználónév és jelszó párokkal használható.

A program felépítése

A program felépítése 3 fő részre bontatható.

1. Menü sáv, ahol váltani tudunk a funkciók között
2. Az a terület, ahol a funkciók megjelennek. Ez a rész a megjelenítésre szolgáló terület 80%-át teszi ki.
3. Feladatkezelő modul. Az ablak maradék 20%-át tölti ki.

Továbbiakban szemléltetni fogom, és részletesen magyarázni a funkciókat és az ezekkel kapcsolatos tudnivalókat.

A program funkcionális leírása

Bejelentkezés:

A program 2 fajta bejelentkezési formával rendelkezik. Az első esetén egy felhasználónév és jelszó segítségével lehet használni a programot. Ha beírtuk a megfelelő adatokat, akkor a „Bejelentkezés” gombra kattintva elindul a beléptetés a rendszerbe. Ha sikeresen megtörténik a hitelesítés, akkor elérhetővé válnak azok a funkciók, amikhez jogosultságunk van. A munkaidő-nyilvántartóhoz, az időbeosztás tervezőhöz és a feladatkezelőhöz nincs szükség külön jogosultságra, ehhez minden felhasználó hozzáfér. A második típusú egyszerűsített bejelentkezés esetén a felhasználónak meg kell adnia felhasználói nevét, és ki kell választani egy tevékenységet. Ha a felhasználó elvégezte ezeket a lépéseket, akkor a „Rögzítés” gombra kell kattintani. A rendszer ellenőrzi, hogy megfelelő-e a megadott felhasználói név, és ha megtalálható az adatbázisban, akkor rögzíti a tevékenységet. Ebben a bejelentkezési fajtában nincs több teendője a felhasználónak.

The screenshot shows a dark header bar with the text "Munkaidő nyilvántartó rendszer". Below it, there are two login forms side-by-side. The left form is titled "Bejelentkezés" and has two input fields: "Felhasználónév" and "Jelszó", followed by a blue button labeled "Bejelentkezés". The right form is titled "Egyszerűsített bejelentkezés" and has one input field for "Felhasználónév" and a dropdown menu for "Tevékenység" with "Munka kezdése" selected, followed by a blue button labeled "Rögzítés".

1. ábra: Bejelentkezési felület

Bejelentkezés után a program automatikusan átirányít a „Főoldal” menüpontra. Itt kiírásra kerül a felhasználó teljes neve, és alatta megjelennek táblázatos formában a számára kiírt szabadságok.

Munkaidő-nyilvántartó:

Az oldal első fő eleme egy naptár. Ez a naptár dinamikusan generálódik. Betöltés után mindig az aktuális dátumra áll rá, ezzel megkönnyítve a felhasználó tevékenység rögzítési szándékát. Az adott napra kattintva betöltődik a tevékenységi napló, ahol megtekinthetjük a berögzített tevékenységeket. Az évszám mellett jobb és bal oldalt található egy-egy nyíl, amivel képesek vagyunk a hónapok között lépkedni.

2020. Május						
Hétfő	Kedd	Szerda	Csütörtök	Péntek	Szombat	Vasárnap
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

2. ábra: Munkaidő-nyilvántartóban található naptár







A második elem egy űrlap, amelynek kitöltése folyamán választhatunk azon projektek közül, amelyekhez van jogosultságunk rögzíteni. Minden projekthez egyedileg megadható feladatok tartoznak. A megfelelőt kell kiválasztanunk a tevékenységhez a jobb adminisztrálás érdekében. Ezek a feladatok egy lenyíló menüpontban kerülnek listázásra, melyek az egyszerűbb követhetőség miatt, fa struktúrában jelennek meg. A feladat kiválasztása után kötelezően meg kell adni a tevékenységre fordított időt óra és perc formában. Ezeket kötelező beírni, mert hiányukban nem végezhető el a rögzítés. A program előugró ablakban jelzi, ha a felhasználó nem töltött ki olyan mezőt, amelyet kötelező. Az űrlapon az utolsó mező egy megjegyzés rovat, ahol tetszőlegesen hozzáfűzhetünk üzenetet a rögzítendő tevékenységhez, de ez opcionális. Ha minden mezőt jól töltöttünk ki, akkor a „Mentés” gombra kattintva a tevékenységünk rögzítésre kerül az adatbázisba, ellenkező esetben pedig a program figyelmeztet, hogy mely adatokon kell még módosítani, mert nem felelnek meg az elvárásoknak. Bármikor megszakíthatjuk az űrlap kitöltési folyamatát, ilyenkor nincs más dolgunk minthogy a „Törlés” gombra kattintunk, és a mezőket alaphelyzetbe állítja a program.

2020. Május 5.

Projekt	Fejlesztés	Feladat	Tervezés
Óra		Perc	
Megjegyzés			
<div>Mentés</div> <div>Törlés</div>			

3. ábra: Tevékenység rögzítésére szolgáló űrlap

Az űrlap alatt található egy táblázat, amelyben megjelennek az adott napra rögzített tevékenységek. Ebben a részben csak a projekt és a feladat neve jelenik meg. Ha meg szeretnénk tekinteni a ráfordított időt és a hozzá írt megjegyzéseket, akkor rá kell kattintanunk a tevékenység sorára, és a naptár alatt található üres területen listázásra kerül minden adat. A tevékenységek soraiban található egy kék módosításra és egy piros törlésre szolgáló gomb. A módosítás gombra történő kattintás esetén a program rögtön betölti az új tevékenység rögzítésénél használt űrlapba az adatokat. Ha elvégeztük a kívánt módosításokat, akkor a „Mentés” gombra kattintva a program újra ellenőrzi, hogy a beírt adatok megfelelnek-e a kívánt követelményeknek, ha nem, akkor figyelmeztetést küld, hogy mely adatok nem megfelelőek. Abban az esetben, ha minden feltételnek eleget tesznek, akkor a program frissíti az adatokat az adatbázisban. Módosítás közben, ha bármikor meggondoljuk magunkat, akkor a „Törlés” gombra kattintva megszakíthatjuk ezt a folyamatot, és az űrlap alaphelyzetbe áll. Ha tévesen rögzítettünk tevékenységet az alkalmazásban, akkor a törlésre is van lehetőség. A tevékenység mellett található piros gombra kattintva a program figyelmeztet és megerősítésre vár. Ha elfogadjuk és a „Törlés” gombra kattintunk, akkor a tevékenység törlésre kerül az adatbázisból.

Projekt	Feladat	
Fejlesztés	Tervezés	 
Fejlesztés	Frontend implementálás	 
Fejlesztés	UML Diagram készítés	 

4. ábra: Tevékenységek táblázatos megjelenítése

Ezen a felületen tekinthetjük meg azokat az elkészített feladatokat is, amiket a feladatkezelő modulon keresztül fogadtunk el. Bővebben a Feladatkezelő modulnál fogom ismertetni az ezzel kapcsolatos részleteket.

Szabadság tervező:

Az adatbázisban minden felhasználónak berögzítjük, hogy egy év alatt hány nap szabadság kiosztására van lehetőség. Ehhez a modulhoz már megfelelő jogosultsággal kell rendelkezünk, ha használni szeretnénk. Itt van lehetőségünk minden felhasználónak beállítani, hogy adott évben mettől meddig kap szabadságot.

Szabadság felviteléhez egy nagyon egyszerű űrlapot kell kitölteni. Először ki kell választanunk, hogy melyik évre szeretnénk elkészíteni a szabadság tervet. Ha ezt megtettük, akkor egy lenyíló típusú mezőben ki kell választani azt a felhasználót, akinek a szabadságát szeretnénk elkészíteni. Ezek után a program megjeleníti, hogy a kiválasztott felhasználónak mennyi az adatbázisba berögzített összes szabadsága, mennyi a már kiosztott és még kiosztható napok száma. A megjelenő adatok alatt van lehetőségünk megadni a szabadság időtartamának első és utolsó napját. A könnyebb dátum kezelés végett speciális dátum típusú mezőket használ az alkalmazás. Ha a kitöltést sikeresen elvégeztük, akkor a „Mentés” gombra kattintva véglegesíthetjük a szabadság tervet. A program ellenőrzi, hogy megfelelően adtuk meg a dátumot. A rendszer fel van készítve olyan hibalehetőség szűrésére is, hogy nem engedélyezi, hogy a szabadság első napja később legyen, mint az utolsó. Arra is oda kell figyelnünk, hogy a fentebb kiválasztott éven belül van csak lehetőségünk szabadságot kiadni. Ha minden ellenőrzésen átmegy, akkor berögzíti az adatbázisba a tervet, ellenkező esetben hibaüzenettel jelzi a felhasználónak, hogy mely mezőben található adatok nem megfelelőek.

Szabadság tervező

Év	2020	↓	Dolgozó	Kis Pista	↓
----	------	---	---------	-----------	---

Összes szabadság: 20 nap
Kivett szabadságok szám: 13 nap
Kiadható szabadságok száma: 7 nap



Szabadság kiadása

Első nap	2020. 05. 05.	Utolsó nap	2020. 05. 07.
----------	---------------	------------	---------------

Mentés

5. ábra: A szabadságterv rögzítésének űrlapja

A felvitelre szolgáló űrlap mellett lehet megtekinteni a már kiadott szabadságokat. A táblázat megjelenítése egy olyan struktúrára épül, ahol először megjelenik a felhasználó neve és a már kiadott szabadságok napjainak száma. A következő sorokban megtekinthető a felhasználóhoz tartozó szabadságok, hogy mettől meddig tart és a napok száma. A szabadságtervek törlésére is van lehetőség. Minden sorhoz tartozik egy piros színű törlésre szolgáló gomb, amire rákattintva megkezdhetjük a törlést. Minden törlés előtt az alkalmazás figyelmeztet, hogy biztosan szeretnénk-e törölni a tervezetet. A „Törlés” gombra kattintva véglegesen eltávolításra kerül az adatbázisból a tervezet.

Név	Időpont	Nap	
Kis Pista		13	
	2020-03-01 - 2020-03-05	5	
	2020-03-23 - 2020-03-30	8	
Kotroczo Roland		0	

6. ábra: Szabadságok megjelenítésére szolgáló táblázat

Időbeosztás tervező:

Sokak számára nehézséget jelenthet egyben tartani azokat a tevékenységeket, amelyeket egy nap vagy egy hét folyamán el szeretnének végezni. Ez a modul ebben

próbál segítséget nyújtani használóinak, illetve a vezetőknek is lehetőséget nyújt beosztottjaiknak időbeosztás tervezésére.

Az oldal betöltésekor a program mindig az aktuális hétre áll rá. Legfelül megjelenik az aktuális év, és mellette jobb és bal oldalt egy-egy nyíl, amelyek segítségével előre és hátra tudunk lépkedni a hetek között. Azok a felhasználók, akik rendelkeznek jogosultsággal, hogy más felhasználóknak is tudnak terveket kiadni, azok az évszámtól jobbra megtalálják egy „Kiosztott tervek” gombot. A gomb elnevezése elég beszédes, ennek segítségével tudják listázni a kiadott terveket. A táblázat fejlécének első oszlopában található a „Terv hozzáadása” gomb, amely segítségével tudunk új tervet hozzáadni táblázatunkhoz. A további 7 oszlopban az aktuális hét napjai találhatóak. A táblázat első oszlopában az időpontok 0 óra 0 perctől indulva 23 óra 45 percig negyedórás léptékben generálódnak, de a programban igény szerint módosíthatóak ezek az értékek. A program az idő alapján a megfelelő sorokban egymás alatt jeleníti meg az adatbázisba rögzített adatokat.

2020							
Kiosztott tervek							
Terv hozzáadása	Május 4 Hétfő	Május 5 Kedd	Május 6 Szerda	Május 7 Csütörtök	Május 8 Péntek	Május 9 Szombat	Május 10 Vasárnap
00:00							
00:15							
00:30			• Megbeszélés				
00:45							
01:00							

7. ábra: Időbeosztás tervező kinézete

Terv rögzítésekor a következő adatokat kötelezően meg kell adni: dátum, óra, perc és cím. Az említett adatok hiányában a „Mentés” gombra kattintva figyelmeztetést kapunk a rendszertől. Minden tervhez opcionálisan megadható tetszőleges leírás is. Azok a felhasználók, akik rendelkeznek jogosultsággal, hogy más felhasználóknak terveket rögzítsenek, számukra megjelenik egy extra mező, ahol ki lehet választani, hogy melyik felhasználónak az időbeosztás tervébe szeretnék beilleszteni a tervet. Abban az esetben, ha a létrehozó saját magának szeretné elvégezni a rögzítést, és rendelkezik jogosultsággal, hogy másnak is rögzíthet, akkor üresen kell hagyni a felhasználó mezőt.

Dátum	2020. 05. 11.				
Óra	08	↕	Perc	45	↕
Cím	Megbeszélés				
Leírás					
Felhasználó	Kis Pista				
Mentés					

8. ábra: Terv rögzítése

A „Kiosztott tervek” gombra kattintva egy táblázatban megjelennek azok a tervek, amiket más felhasználóknak adott ki a program használója. A program idő szerint csökkenő sorrendben listázza a feladatokat. Ezen a felületen van lehetőségünk a tervek törlésére is. Minden terv sorában található egy szemetes ikonnal jelzett piros gomb, amire rákattintva véglegesen eltávolíthatjuk a tervet az adatbázisból.

Dátum	Cím	Leírás	Kiadott felhasználó
2020. 05. 11. 08:30	Megbeszélés		Kis Pista

9. ábra: Kiosztott tervek listázása

Listázás:

Ebben a menüpontban van lehetőségünk listázni a mások által berögzített tevékenységeket és a felhasználóknak kiadott szabadságokat. A listázás funkcióhoz is megfelelő jogosultsági szinttel kell rendelkezünk.

Az oldal betöltésénél az adatbázisban lévő összes adatot egy táblázatban jeleníti meg. A program lehetőséget ad a megjelenített adatok szűrésére. A szűrők használatakor csak azok az adatok jelennek meg a táblázatban, amelyek megfelelnek a szűrési feltételeknek. A projektek listázásánál a projekt, feladat és felhasználó alapján tudunk szűrni, a szabadságok esetén pedig a felhasználó és az év szerint. Abban az esetben, ha beállítottunk szűrési feltételeket, de ezeket egyszerre törölni szeretnénk, akkor a „Szűrő törlése” gombra kattintva a szűrők alaphelyzetbe állnak, és újra az összes adat megjelenik a táblázatban.

A rendszer lehetőséget ad az adatok exportálására Excel formátumban. A funkció használatához az „Excel” gombra kell kattintani és a program elkészíteni a fájlt,

és letölti a böngésző által beállított alapértelmezett letöltési mappába „export.xlsx” néven. Abban az esetben, ha használtunk szűrési feltételeket, akkor a program annak megfelelően fogja elkészíteni az export állományt.

Projekt	Összes	Feladat	Összes	Felhasználó	Összes	Dátum	Idő	Megjegyzések
Fejlesztés								
Tervezés								
Adatbázis tervezés								
UML Diagram készítés								
				admin		2020-03-08	2 óra 30 perc	Tervezés


10. ábra: Projektek listázása

Felhasználók – Karbantartás:

Az oldal felépítése 2 részre bontható. Az egyik, ahol az új felhasználókat tudjuk rögzíteni, a másik pedig egy táblázat, ahol az adatbázisban levő felhasználók listázásra kerülnek.

Új felhasználó rögzítése esetén a következő adatokat szükséges megadni: vezetéknév, keresztnév, bejelentkezési név, e-mail cím, egy évre szóló összes szabadság és hogy engedélyezett-e számára a bejelentkezés. Minden mező kitöltése kötelező, ennek hiányában a rendszer hibaüzenetet küld a felhasználónak. Minden felhasználónak egyedi bejelentkezési névvel kell rendelkeznie. Mentés előtt a program ellenőrzi a bejelentkezési név egyediségét, és csak abban az esetben fogadja el, ha nem talál egyezőséget az adatbázisban. Új felhasználó regisztrálásánál ügyelni kell a bejelentkezési név és e-mail cím helyes megadására, mert később ezen adatok módosítására nem lesz lehetőség. Az engedélyezett mezőben igen/nem opciókkal tudjuk a rendszer számára jelezni, hogy az adott felhasználó bejelentkezhet-e a programba vagy sem.

Új felhasználó


Vezetéknév	<input type="text"/>
Keresztnév	<input type="text"/>
Bejelentkezési név	<input type="text"/>
E-mail cím	<input type="text"/>
Összes szabadság	<input type="text"/>
Engedélyezett	Igen 
<div>Mentés Törlés</div>	

11. ábra: Új felhasználó rögzítése

Az újonnan beregisztrált felhasználók nem rendelkeznek jelszóval. Első bejelentkezésnél a felhasználónak meg kell adni a bejelentkezési nevét és rá kell nyomni a „Bejelentkezés” gombra. Ilyenkor a rendszer érzékeli, hogy újonnan regisztrált felhasználó szeretne bejelentkezni, és egy felugró ablakban bekéri a jelszavát. A jelszó megadása után a program használhatóvá válik számára a bejelentkezési név és az általa megadott jelszó segítségével.

A már beregisztrált felhasználókat egy táblázatban tekinthetjük meg. A táblázatban nem jelenik meg minden adat a felhasználóról, csak a vezetékeve, keresztnéve, bejelentkezési neve, e-mail címe, utolsó bejelentkezés időpontja és hogy engedélyezett-e számára a bejelentkezés. Egy felhasználó összes adatát úgy tudjuk megtekinteni, ha a táblázatban rákattintunk a sorára. Ilyenkor egy előugró ablakban minden információ kiírásra kerül.

Felhasználó adatai

Vezetéknév: Kotroczo
Keresztnév: Roland
Bejelentkezési név: admin
E-mail cím: k.rol19955@gmail.com
Utolsó bejelentkezés: 2020. 05. 11. 17:51
Engedélyezett: 
Összes szabadság: 20
Jogosultságok:

- Feladat kiosztása
- Szabadságok kiosztása
- Felhasználók kezelése
- Projektek kezelése
- Feladatok kezelése
- Jogosultságok kezelése
- Listázás
- Időbeosztás tervezése

Projektek:

- Fejlesztés: 2020-03-01 -

12. ábra: Felhasználó adatai részletesen.

Az előugró ablaknál, ahol megtekinthetjük a felhasználók részletes adatait lehetőségünk van adatainak módosítására és a felhasználó törlésére. Ha módosítani szeretnénk, akkor a kék gombra kattintva az ablak bezáródik, és a regisztrációnál használt űrlapba betöltődik minden adat. Az adatok módosíthatók a felhasználói név és e-mail cím kivételével. A szerkesztett adatok véglegesítéséhez a „Módosítás” gombra kell kattintani. Az alkalmazás ellenőrzi az újonnan bevitt adatokat, és ha minden feltételnek megfelelnek, akkor a felhasználó adatai véglegesen is módosításra kerülnek az adatbázisban. Abban az esetben, ha a módosítást nem szeretnénk elvégezni, akkor a „Törlés” gombra kell kattintani, ilyenkor az űrlap alaphelyzetbe áll, és a módosítás elvetésre kerül.

Egy felhasználó végleges törlésére úgy van lehetőségünk, hogy megnyitjuk a részletes adatait és az előugró ablaknál a piros törlés gombra kattintunk. Ilyenkor a program figyelmeztet, hogy biztosan törölni szeretnénk-e a felhasználót. Abban az esetben, ha a „Törlés” gombra kattintunk, akkor a felhasználó eltávolításra kerül az adatbázisból.

Projektek – Karbantartás:

Ebben a menüpontban van lehetőségünk új projektek rögzítésére és az adatbázisban szereplő projektek szerkesztésére.

Új projekt létrehozásához meg kell adni egy projekt nevet és arányt. Ügyelnünk kell egyedi név megadására, mert a rendszer két azonos nevű projektet nem engedélyez. Itt van lehetőségünk hozzárendelni a projektekhez a felhasználókat. A felhasználónevek egy lenyíló menüpontban jelennek meg. A felhasználó kiválasztása után meg kell adni, hogy mikortól szeretnénk hozzárendelni a projekthez. Ez a dátum azért szükséges, mert ettől a naptól fogja látni a felhasználó a munkaidő-nyilvántartó modulban a projektet, és így tud tevékenységet rögzíteni hozzá. Utolsó nap megadása nem szükséges, azonban ilyenkor visszavonásig van jogosultsága tevékenység rögzítéséhez. Abban az esetben, ha megadjuk mind a két dátumot, akkor csak az időintervallum között van hozzáférése a felhasználónak. Ha kiválasztottuk a felhasználót, és beállítottuk a hozzáférés dátumait, akkor a felhasználó neve mellett szereplő „+” gombra kell kattintani, ilyenkor kerül összekapcsolásra a projekt és a felhasználó. Ezek után újabb emberek rendelhetők a projekthez. A rendszer nem tud egy felhasználót többször hozzárendelni egy projekthez, ezért nincs lehetőség több időintervallum megadására. Ilyenkor a legbővebb

időintervallumot kell megadni a felhasználónak. Ha megpróbálunk többször hozzáadni egy felhasználót az adott projekthez, akkor a program figyelmeztet, és nem engedélyezi ezt. Egy felhasználót nem csak hozzárendelni lehet, hanem eltávolítani is egy projektből, ilyenkor az adott felhasználó sorában lévő piros törlés gombra kell kattintani. A projekt végleges rögzítéséhez a „Mentés” gombra kell kattintani, és a rendszer elvégzi a szükséges ellenőrzéseket, illetve véglegesíti az adatokat az adatbázisban, vagy hibaüzenetet küld a felhasználónak, hogy mely adatok hibásak, amiket javítani kell. Abban az esetben, ha meggondoltuk magunkat, és mégsem szeretnénk új projektet berögzíteni, akkor a „Törlés” gombra kattintva az űrlapot alaphelyzetbe állíthatjuk.

Új projekt

Név

Fejlesztés

Árány

100

Felhasználó

Kis Pista

+

Első nap

2020. 05. 18.

Utolsó nap

2020. 06. 30.

Mentés

Törlés

Felhasználó	Tól	Ig	
Kotroczo Roland	2020-05-18		
Kis Pista	2020-05-18	2020-06-30	

13. ábra: Új projekt létrehozása

Miután berögzítettük a projekteket, meg is tekinthetjük őket. Erre szolgál a beviteli mezők mellett található táblázat. Itt jelennek meg a projektek. Nem jelenik meg minden információ a táblázatban, csak a projekt neve, aránya, és a hozzárendelt felhasználók és feladatok száma. Ha részletesen meg szeretnénk tekinteni a projekt adatait, akkor a megfelelő projektre kattintva egy előugró ablakban listázásra kerülnek. Részletesen listázásra kerül, hogy melyik felhasználónak milyen időintervallumban van jogosultsága. Minden projekthez rendelhetünk feladatokat, amelyekhez rögzíthetjük a tevékenységeinket a munkaidő-nyilvántartó modulban. Ezen a felületen azokat is megtekinthetjük. A könnyebb olvashatóság érdekében itt is a fa szerkezet alapján kerülnek megjelenítésre.

Projekt adatai

Név: Fejlesztés

Árány: 100

Hozzárendelt felhasználók:

- Kotroczo Roland: 2020-03-01 -

Feladatok:

- Tervezés
- - Adatbázis tervezés
- - - MYSQL
- - UML Diagram készítés
- Fejlesztés
- - Frontend implementálás

14. ábra: Egy projekt részletes adatai

A projektek módosítására is van lehetőség. Az előugró ablaknál a „Projekt adatai” felirat mellett található kék gombra kattintva az ablak bezáródik, és a projekt adatai betöltődnek abba az űrlapba, ahol új projektet tudunk létrehozni. Minden adat változtatható. Módosításnál ugyanazokra a szabályokra kell figyelni, mint új rögzítése esetén. Miután elvégeztük az összes változtatást, a „Módosítás” gombra kell kattintani. Ilyenkor a rendszer újra ellenőrzi az összes beírt értéket, és ha megfelelnek a feltételeknek, akkor elvégzi a módosítást az adatbázisban, ellenkező esetben pedig hibaüzenetet küld a felhasználónak a hibás adatokról. A „Törlés” gombra kattintva elvethetjük a módosításokat, és az űrlap alaphelyzetbe áll.

Előfordulhat, hogy törölni szeretnénk egy projektet az adatbázisból. A projekt részletes adatainál, a fentebb említett kék módosító gomb mellett megtalálható egy piros gomb is, ami a törlésre szolgál. Arra kattintva a program rákérdez, hogy biztosan törölni szeretnénk-e a projektet. A „Törlés” gombra kattintva a projekt véglegesen eltávolításra kerül a rendszerből.

Feladatok – Karbantartás:

Az oldal strukturálisan 3 fő részre osztható. Első az új projektek rögzítése, második a meglevő feladatok listázása és az utolsó pedig a projektek listázása. Ebben a menüpontban tudjuk azokat a feladatokat kezelni, amelyeket hozzá tudunk rendelni a projektekhez, és később a munkaidő-nyilvántartó modulban tevékenységeket tudunk hozzá írni. Itt kell még azokat a feladatokat is megadni, amelyeket az egyszerűsített bejelentkezésnél használunk.

A feladatok felépítése fa szerkezet segítségével működik. Ez azt jelenti, hogy képesek vagyunk egy adott feladatnak alfeladatokat is megadni. Új feladat rögzítése esetén meg kell adni a feladat nevét, szülőjét, bővíthető-e és hogy egyszerűsített bejelentkezésnek számít-e. A név megadásánál ügyeljünk arra, hogy ne adjunk meg azonos nevű feladatot, azonban a fa szerkezet miatt mégis képesek vagyunk erre, hogy ha a feladat nem ugyanarra a szintre kerül. A rendszer mentésnél figyelmeztet, ha ugyanolyan nevű feladatot szeretnénk létrehozni és már létezik a megadott szinten. A szülőnél tudjuk megadni, hogy melyik feladat alá szeretnénk rendelni az új feladatot. Ez az alárendelés csak abban az esetben működik, ha a szülőnél megadtuk, hogy bővíthető-e, ellenkező esetben meg sem jelenik a feladat a szülő mezőben. Az „Egyszerűsített bejelentkezés” mezőnél tudjuk jelezni a rendszer felé, hogy olyan feladatot szeretnénk

létrehozni, amit a felhasználók az egyszerűsített bejelentkezésnél tudnak használni. Ha ezt a mezőt igazra állítjuk, akkor a szülő és bontható mezők inaktívvá válnak. Egyszerűsített bejelentkezési feladatnál nem működik a fa szerkezet. Minden szükséges adat megadása és beállítása után nincs más dolgunk, mint a „Mentés” gombra kattintani. Bármilyen hiba esetén a rendszer jelzi a felhasználó felé a megfelelő hibaüzenet segítségével. Abban az esetben, ha mégsem szeretnénk elvégezni a feladat rögzítését, de elkezdtük az űrlap kitöltését, akkor a „Törlés” gombra kattintva a mezők alaphelyzetbe állnak.








Új feladat

Név	<input type="text"/>
Szülő	- Nincs -
Bővíthető	Igen
Egyszerűsített bejelentkezés	Nem

Mentés Törlés

15. ábra: Új feladat rögzítése

Az oldal második fő részében található az a táblázat, ahol megtekinthetjük az adatbázisban található feladatokat. Itt kerül listázásra a feladat neve, szülője, bontható-e és hogy egyszerűsített bejelentkezésnek számít-e. A feladatok úgy kerülnek listázásra, hogy az összefüggő elemek egymás alatt helyezkednek el. Azok a feladatok, amelyek neve mellett nem található „-” jel, azok számítanak a fa gyökereinek, nekik nincs szülőjük. Azok a feladatok, amelyeknek a neve „-” jellel kezdődik, mind valamelyik feladatnak az alfeladatai. A „-” jelek darabszámai mutatják meg, hogy az adott fában milyen mélyen találhatóak. Minden feladat sorában található egy piros gomb, ami a törlésre szolgál. A gombra kattintva a rendszer figyelmeztet, hogy valóban el szeretnénk-e távolítani az adatbázisból. A „Törlés” gombra nyomva véglegesen eltávolításra kerül. Egy feladat törlésénél ügyelni kell arra, hogy ha egy olyan feladatot távolítunk el az adatbázisból, aminek vannak alfeladatai, akkor azok is törlésre kerülnek. Erre a rendszer is figyelmeztet törlés esetén. Egy feladat módosítására nincs lehetőség a rendszerben. Abban az esetben, ha tévesen mentettük a feladatot, akkor előtte törölni kell és újra rögzíteni.

Név	Szülő	Bontható	Egyszerű. bejel.	
Tervezes	- Nincs -	✓	✗	
- Adatbázis tervezes	Tervezes	✓	✗	
- - MYSQL	Adatbázis tervezes	✗	✗	
- UML Diagram keszites	Tervezes	✗	✗	
Fejlesztés	- Nincs -	✓	✗	
- Frontend implementálás	Fejlesztés	✗	✗	
Munka kezdése	- Nincs -	✗	✓	

16. ábra: Feladatok listázása

Az oldal utolsó részénél, ahol megtekinthetjük a projekteket, van lehetőségünk minden projekthez feladatokat rendelni. A projekt listában kiírásra kerül a neve, aránya, felhasználók- és hozzárendelt feladatok száma. Egy projekt sorára kattintva előugrik egy ablak, ahol listázásra kerülnek a feladatok. Azok a feladatok, amelyek nevei mellett egy „✓” szerepel, össze vannak kapcsolva a projekttel. Azok a feladatok pedig, ahol az „✗” szerepel, nincs kapcsolat a feladatok és a projekt között. A „Projekt feladatai” felirat mellett található kék gomb segítségével tudjuk szerkeszteni, hogy melyik feladatokat szeretnénk hozzáadni a kiválasztott projekthez. A gomb lenyomása után a feladatok sorai szerkeszthetővé válnak. Ha az adott feladat nem volt még hozzárendelve a projekthez, és rákattintunk a sorára, akkor megjelenik a „✓”, ellenkező esetben pedig az „✗”. A feladatok fa struktúrája miatt a rendszer figyeli azt, hogy ha egy alfeladatot választunk ki, akkor a szülő feladatait is ki kell választani, vagy ha esetleg egy gyökér feladatot kivesszünk a listából, akkor az alfeladatait is ki kell. Ezeket a módosításokat a program automatikusan végzi. A véglegesítéshez a kék gomb helyére egy „Mentés” gomb kerül, amelyre rákattintva a rendszer elvégzi a hozzárendeléseket az adatbázisban.

Projekt feladatai

Név	Szülő	
Tervezes	- Nincs -	✓
- Adatbázis tervezes	Tervezes	✓
- - MYSQL	Adatbázis tervezes	✗
- UML Diagram keszites	Tervezes	✗
Fejlesztés	- Nincs -	✓
- Frontend implementalas	Fejlesztés	✓

17. ábra: Feladatok hozzárendelése egy projekthez

Jogosultságok – Karbantartás:

Ebben a menüpontban tudunk létrehozni jogosultsági csoportokat és hozzárendelni felhasználókat. Minden jogosultsági csoportnak megadható, hogy mely modulokhoz férhetnek hozzá. Értelmszerűen csak azokat a modulokat tudjuk szabályozni, amelyek magasabb jogosultsági szintet követelnek meg, mint az alapértelmezett felhasználói szint.


Új jogosultsági csoport megadása esetén egyetlen egy nevet kell megadni, és arra kell ügyelni, hogy nem lehet két azonos nevű csoport az adatbázisban. Ha megadtuk a nevet, akkor a „Mentés” gombra kell kattintani. A rendszer rögzíti a csoportot az adatbázisba, ha nem talál azonos nevűt. Névegyezés esetén a rendszer hibaüzenetet küld a felhasználónak. A név mező ürítésére a „Törlés” gomb szolgál.

Új jogosultsági csoport

18. ábra: Új jogosultsági csoport rögzítése

A beviteli mező mellett található táblázat, ahol megtekinthetjük a már létrehozott jogosultsági csoportokat. A táblázat tartalmazza a csoport nevét, a hozzárendelt jogosultságok- és a csoporthoz rendelt felhasználók darabszámát. Egy sorra kattintva megtekinthetjük, hogy mely jogosultságok és felhasználók vannak hozzárendelve a csoporthoz. Ezen a felületen van lehetőségünk szerkeszteni az összekapcsolásokat. A csoport neve mellett található kék gombra kattintva módosíthatóvá válnak a jogosultság és felhasználó mezők. Azok az elemek, amelyek a „✓” szimbólummal jelennek meg,

hozzá vannak rendelve a csoporthoz, értelemszerűen a „X”-el jelöltek pedig nem. Amikor a szerkesztés engedélyezett, akkor a sorokra kattintva változtathatók. Minden felhasználó csak egy jogosultsági csoporthoz rendelhető hozzá. Módosítás után az előbbi kék gomb helyén szereplő „Mentés” gombra kattintva a rendszer véglegesíti a szerkesztéseket az adatbázisban.

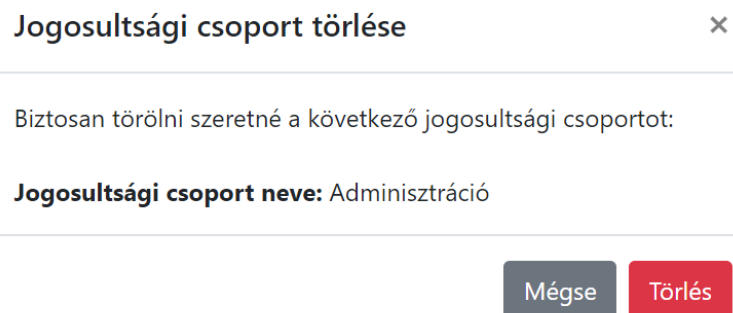
Jogosultsági csoport: Adminisztráció 

Név		Név	Bejelentkezési név	E-mail cím	
Feladat kiosztása	✓	Kotroczo Roland	admin	k.rol9955@gmail.com	✓
Szabadságok kiosztása	✓	Kis Pista	user	k.rol9955@gmail.com	✗
Felhasználók kezelése	✓				
Projektek kezelése	✓				
Feladatok kezelése	✓				
Jogosultságok kezelése	✓				
Listázás	✓				
Időbeosztás tervezése	✓				

19. ábra: Jogosultsági csoportok szerkesztése

Minden jogosultsági csoport esetén van lehetőségünk a nevüket megváltoztatni vagy akár véglegesen törölni. Ahogy korábban már megszokhattuk, a megjelenítési táblázatban, minden sor végén megtalálható egy kék és egy piros gomb, amiket ezekre használhatunk. A kékre kattintva a rendszer betölti a csoport nevét abba a beviteli mezőbe, ahol létrehoztuk. Szerkesztés után nincs más dolgunk, mint a „Módosítás” gombra kattintani. A program ellenőrzi, hogy egyedi csoport nevet adtunk-e meg, ha igen, akkor a változtatások véglegesítésre kerülnek az adatbázisban, ellenkező esetben pedig hibaüzenetet küld a felhasználónak. Ha megkezdtük egy csoport nevének a szerkesztését, de nem szeretnénk elmenteni, akkor a „Törlés” gombra kattintva a módosítások elvetésre kerülnek. A fentebb említett piros gomb nyújt segítséget abban, hogy véglegesen eltávolítsunk egy csoportot az adatbázisból. A gombra kattintva a rendszer még figyelmeztet, és csak a „Törlés” gomb lenyomása után kerül eltávolításra a rendszerből.

Jogosultsági csoport törlése esetén ügyelnünk kell arra, hogy azok a felhasználók, amelyek az adott csoportban szerepelnek, visszakerülnek az alapértelmezett felhasználói szintre. Előfordulhat olyan, hogy a felhasználó azt a csoportot próbálja meg módosítani, amelyikhez hozzá van rendelve, akkor elveszíthet számos jogosultságát, ha áthelyezi magát egy alacsonyabb szintű jogokkal rendelkező csoportba.



20. ábra: Jogosultsági csoport törlésénél figyelmeztetés

Feladatkezelő

Az oldal felépítésnél már említettem, hogy ez a modul teszi ki a program 20%-át vizuális megjelenítésnél. Itt lesz lehetőségünk magunknak vagy más felhasználóknak feladatokat kiosztani, és később ezeket összekötni a munkaidő-nyilvántartásunkkal. Ez a modul mindig aktívan látszik, nem számít, hogy az alkalmazás melyik oldalán tartózkodunk. A weboldal jobb oldalán található sávban helyezkedik el.

A feladatkezelő a következő módon épül fel. A modul legtetjén szereplő „Feladat rögzítése” gomb elég beszédes, annak segítségével tudunk új feladatokat bevinni az adatbázisba. Közvetlenül alatta található a „Kiosztott feladatok” gomb. Ez csak azoknak jelenik meg, akiknek van jogosultságuk más felhasználóknak feladatokat kiosztani. A gombok alatt helyezkednek el az általunk és mások által berögzített feladatok. A rendszer időrendi sorrendben csak azokat jelenít meg, amelyeket még nem teljesítettünk. A könnyebb követhetőség miatt színkódolással jelennek meg a feladatok. Sárga színű az a feladat, aminek a határidőjéből kicsúszott a felhasználó. Kék lesz az, amelyiknek az aktuális dátuma megegyezik a határidővel és fehér lesz az, amelynek a határideje később van, mint az aktuális dátum. Minden feladatnál megtalálható egy „Kész” gomb, melyre kattintva elvégzetté tehetjük a feladatot. Ilyenkor a feladat eltűnik a feladatkezelőből, és bekerül a munkaidő-nyilvántartó modulba, arra a napra, amikor rányomtunk a gombra. A „Törlés” gomb csak azoknál a tevékenységeknél jelenik meg, amelyeket saját magunknak hoztunk létre. Azokat a feladatokat, amelyeket már teljesítettünk, és saját magunknak hoztunk létre, utólag törölhetjük az adatbázisból. Ilyenkor a munkaidő-nyilvántartó modulban meg kell keresni az adott feladatot, és ott van lehetőségünk törölni. Ennek a folyamata megegyezik a hagyományos tevékenység eltávolításával.

Feladat rögzítése
Kiosztott feladatok

Feladatok
Határidő: 2020-05-20 Leírás: Tervezes <div>Kész</div>
Határidő: 2020-05-21 Leírás: Meeting <div>Kész</div> <div>Törlés</div>
Határidő: 2020-05-22 Leírás: Megbeszeles <div>Kész</div> <div>Törlés</div>
Határidő: 2020-05-25 Leírás: Konferencia <div>Kész</div>

21. ábra: Feladatkezelő

Új feladat rögzítése esetén meg kell adnunk a feladat határidejét és leírását. Ide tetszőlegesen bármi megadható. Érdeemes olyan leírást megadni, ami egyértelművé teszi a felhasználó számára a feladatot. Azok a felhasználók, akik rendelkeznek jogosultsággal, hogy mások számára is kioszthassanak feladatokat, számukra megjelenik még egy „Felhasználó” mező is, ahol kiválaszthatják, hogy kinek szeretnék rögzíteni a feladatot. Ha üresen marad ez a mező, akkor automatikusan saját maguknak rögzítik a tevékenységet.

Határidő	2020. 05. 21.
Megjegyzés	Megbeszélés
Felhasználó	Kis Pista
Mentés	

22. ábra: Feladat rögzítése

A másoknak kiosztott feladatokat a fentebb említett gombra kattintva megtekinthetjük. Listázásra kerül a feladat határideje, leírása, a felhasználó neve, akinek

kiadtuk a feladatot, és hogy teljesítette-e már. Ha igen, akkor a teljesítés dátuma is megjelenik. Azok a feladatok törölhetőek, amelyeket még nem teljesített a felhasználó.

Határidő	Leírás	Felhasználó	Teljesített	Teljesítés dátuma
2020-05-25	Konferencia	Kis Pista	✓	2020-05-25
2020-05-20	Tervezés	Kis Pista	✗	

23. ábra: Kiosztott feladatok

FEJLESZTŐI DOKUMENTÁCIÓ

A program megvalósítása folyamán próbáltam törekedni arra, hogy könnyen átlátható és más fejlesztők számára értelmezhető forráskódot hozzak létre. Azokat az algoritmusokat vagy folyamatokat, melyeket több helyen is használok, próbáltam egységesíteni, ezzel elkerülve azt, hogy redundancia jöjjön létre. Több programozási nyelvvel dolgoztam. Próbáltam követni minden nyelv sajátos írott és íratlan konvencióját.

Felhasznált technológiák

Java

A java egy objektumorientált programozási nyelv, aminek a szintaxisa a C nyelvre épül. A compiler bájtkódra fordítja a forráskódot. A lefordított fájlokat a Java virtuális gép futtatja.

Spring

A Spring egy nyílt forráskódú keretrendszer, amely átfogó programozási és konfigurációs modellel szolgál a modern Java alapú applikációknak. A Spring futtatása nehézkes lenne, ezért az alkalmazásomban Spring Boot szolgáltatást használok az egyszerűbb futtatás érdekében.

JPA

A Java Persistence API röviden JPA a Spring Data családjába tartozó keretrendszer a Java programozási nyelvhez, amely az adatbázis műveletekért felel. A repository-ra épülve megkönnyíti a megvalósítást.

Lombok

A Lombok egy Java eszközkönyvtár, aminek a segítségével olyan kódokat generálunk, amelyek mindig ugyanazok, és nem lényeges részei a programnak. Annotációk segítségével tudjuk használni. Ilyen kódrészletek például:

- Getter

- Setter
- Konstruktor

MySQL

A MySQL egy nyílt forráskódú SQL alapú relációs adatbáziskezelő rendszer.

Node.js

A Node.js egy nyílt forráskódú, platformfüggetlen szoftverrendszer, melyet webszerverek készítésére hoztak létre. A JavaScript futtatókörnyezet segítségével hajtja végre a parancsokat. Napjainkra hatalmas népszerűsége tett szert, mert rengeteg új és korszerű fejlesztési eszközzel rendelkezik, amelyek segítik a fejlesztőket.

Angular

Az Angular egy nyílt forráskódú, TypeScript alapú, web alkalmazás keretrendszer. A Google fejlesztette ki, létrehozásának elsődleges célja a kliensoldali dinamikus megjelenítés megkönnyítése volt.

HTML

A HyperText Markup Language röviden HTML egy leíró nyelv, melyet weboldalak megjelenítésére fejlesztettek ki. Napjainkra már szabvánnyá vált a W3C (World Wide Web Consortium) támogatásával.

CSS

A Cascading Style Sheet röviden CSS egy stílusleíró nyelv, amely HTML típusú dokumentumok megjelenítését írja le. Használható még XML alapú dokumentumok stílusának leírására is.

JavaScript

A JavaScript egy objektumorientált programozási nyelv, melyet a weboldalak fejlesztésénél használnak. Kliensoldalon futó webes programok készítésére használják. Gyengén típusos nyelv.

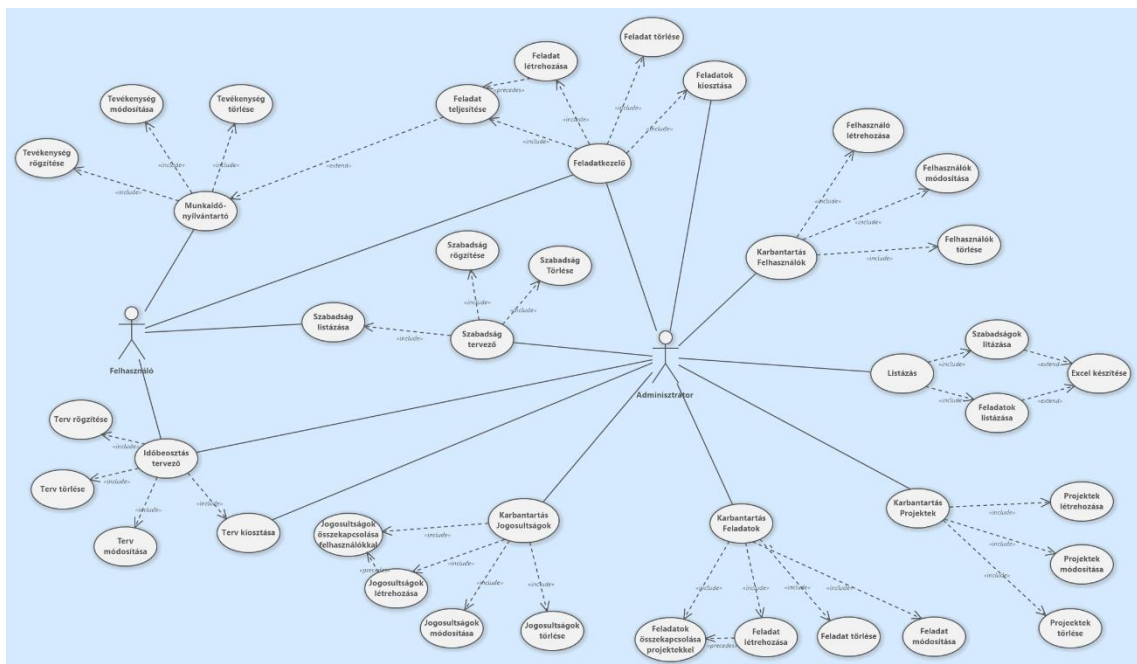
Typescript

A Microsoft által fejlesztett nyílt forráskódú programozási nyelv. A JavaScripthez hasonlóan weboldalak fejlesztéséhez hozták létre, azonban több eszközt tartalmaz, ami elősegíti az alkalmazások fejlesztését. A JavaScript-el ellentétben ez egy erősen típusos nyelv, azonban fordításnál egy JavaScript kóddá alakul át.

Bootstrap

A Bootstrap egy nyílt forráskódú CSS keretrendszer, amelyet reszponzív webalkalmazások fejlesztésének megkönnyítése érdekében készítettek. JavaScript és CSS alapú rendszer.

Felhasználói esetek



24. ábra: Felhasználói esetek diagram

A felhasználói esetek két fő részre bonthatóak, ahogy a 23. ábra szemlélteti. Először az egyszerű felhasználói oldalról közelítsük meg. Az alap felhasználók 3 modullal rendelkeznek, amelyek számos funkciót tartalmaznak. Az adminisztrátorok sokkal bővebb hozzáféréssel rendelkeznek a rendszerhez. A program lehetőséget nyújt, hogy egyénileg megadhassuk, hogy mely adminisztrátori jogot lássa el a program használója.

A szerveroldal (Backend) felépítése

A szerveroldali alkalmazásom megvalósítására a Java alapú Spring keretrendszert választottam. A forráskód egy Maven típusú projektbe lett létrehozva. Minden Spring projekt tartalmaz egy „pom.xml” nevű fájlt, ami tartalmazza azokat az alap információkat, amelyek szükségesek a rendszer indításához. Ezek az adatok határozzák meg, hogy a forráskódunk milyen verziójú Java nyelvet használ. Ez nagyon fontos, mert verziók között lehetnek lényeges különbségek, amik fordítási hibát okozhatnak. A fájl tartalmazza még a verzió számot is, azonban a legfontosabb dolgok, amik szükségesek a fejlesztéshez, a függőségek. Itt soroljuk fel azokat a függőségeket, amelyek elengedhetetlenek a program működéséhez, vagy elősegítik a gyorsabb és dinamikusabb fejlesztést. A következő függőségeket használtam:

- Spring Boot Starter Data JPA
- Spring Boot Starter Web
- Spring Boot Starter Security
- Spring Boot Devtools
- H2
- Lombok
- MySQL Connector Java
- POI-OOXML
- Spring Boot Starter Test

A felsorolt függőségek olyan segéd alkalmazások, amelyeket más fejlesztők készítettek. Ezek a programok nyílt forráskódúak, és más fejlesztők is felhasználhatják a saját alkalmazásuk elkészítésében. Fejlesztői környezetben minden indításnál ellenőrzésre kerül, hogy azok a függőségek, melyek felsorolásra kerültek, létezik-e belőlük újabb verzió. Abban az esetben, ha igen, akkor a Maven típusú projektnek köszönhetően

automatikusan frissítésre kerül nálunk is. Léteznek olyan függőségek is, amelyeknél megadjuk, hogy az alkalmazásunk milyen verziószámút használ, és olyankor nem történik frissítés újabb kiadása esetén.

A program felépítése:

A program 4 fő egységre bontható, amelyek az fájlok elhelyezésében jól látszódnak. Minden osztály és interfész, amit a fejlesztés során használtam, külön csomagban (package) találhatóak meg. Fejlesztés alatt törekedtem arra, hogy a Java konvenciókat betartsam.

Az első egység, amit ismertetni fogok, azok az úgynevezett entitások (Entity). Ezek olyan osztályok, amelyek meghatározzák az alap adatszerkezetünket, amire felépítjük a programunkat. Itt kerülnek definiálásra az osztályon belüli adattagok, függvények, itt határozzuk meg, hogy milyen adatokat szeretnénk adatbázisban tárolni. Minden osztályomat a „BaseEntity.java” absztrakt osztályból származtattam le. Ennek az oka, hogy minden osztályt fel kellett készítenem, hogy exportálható legyen Excel formába. Ennek részleteit később fogom ismertetni. Minden entitásnak egyedi azonosítóval kell rendelkeznie. Az osztályok létrehozásánál nem kell gondoskodnunk az egyedi azonosító generálásról, ez az adatbáziskezelő rendszerre van bízva. Az osztályoknál nincs kézzel megírva getter, setter és konstruktor. A fentebb említett Lombok függőség segít ebben. Ennek segítségével készítettem 2 típusú konstruktort, az egyik az üres, ami nem rendelkezik argumentummal, a másik pedig a minden adattagot argumentumként megkapó konstruktor. Minden osztályhoz tartozik egy úgynevezett @EqualsAndHashCode annotáció, aminek segítségével tudom vizsgálni 2 azonos típusú osztály ekvivalenciáját. Minden adattagnál feltüntetésre került, hogy engedélyezett-e számára, hogy NULL legyen, azaz ne mutasson memória területre az adott változó. A következő osztályokkal rendelkezik az alkalmazásom

- *Activity*: Tevékenységek, amelyeket felhasználók létrehoznak a munkaidő-nyilvántartó modulban. Ezt az osztályt használja a feladatkezelő is.
- *ActivityGroup*: Feladatok
- *ActivityPlan*: Tervek, amiket az időbeosztás tervező modulnál használunk.
- *Holiday*: A felhasználó szabadságát tárolja egy adott intervallumban.
- *Permission*: Jogosultsági csoportok
- *PermissionDetail*: Jogosultsági szintek

- *User*: Felhasználó
- *UserWorkGroup*: Ez az osztály valósítja meg, hogy egy felhasználó melyik projekthez van hozzárendelve.
- *WorkGroup*: Projektek

A következő egység a repository package. Ez a csomag tartalmazza azokat az interfészeket, amelyek segítségével tudunk kommunikálni az adatbázissal. Minden entitáshoz tartozik egy repository. A könnyebb követhetőség miatt, az osztálynevek kiegészítésre kerültek egy „Repository” kulcsszóval. Az interfészek az úgynevezett CrudRepository interfészből kerülnek származtatásra. Ennek segítségével tudjuk a CRUD (Create, Read, Update, Delete) műveleteket használni, azaz létrehozás, olvasás, frissítés és törlés. A repository interfészek tartalmazzák azokat a függvényeket, amelyekkel könnyedén tudunk SQL műveleteket elvégezni. A JPA keretrendszernek köszönhetően nem kell SQL scripteket írni, hanem egy logikát követve kell létrehozni a függvények neveit. Ezekből a függvénynevekből a rendszer elkészíti magának az SQL scriptet. Előfordulhatnak olyan esetek, hogy annyira összetett és egyedi lekérdezést szeretnénk létrehozni, ami nem lehetséges a függvéynévben konkatenált parancs elemekkel. Ezekre az esetekre biztosít számunkra a @Query annotáció megoldást, aminek segítségével megadhatjuk az általunk megírt SQL script-et.

A harmadik egységben találhatóak a vezérlők, amelyek segítségével tudnak más alkalmazások kommunikálni a rendszerünkkel. A vezérlők feladata, hogy fogadják a kéréseket, és az utolsó részben említett service-k által kapott adatokat továbbítják a felhasználóknak. Itt kerülnek megvalósításra azok a végpontok, amelyekre az érkező kérések kiszolgálásra kerülnek. Az alkalmazás 4 fajta kérést szolgál ki.

- GET: Azok a kérések, amelyeket csak listázásra használunk. Nem végeznek semmilyen módosítást az adatbázisban.
- POST: Új adat létrehozása esetén használjuk ezt a kérés típust.
- PUT: Meglévő adatok módosítása esetén használandó.
- DELETE: Olyan kérés, aminek fogadása esetén adatot szeretnénk törölni az adatbázisból.

A végpontokra érkezett kéréseket a szerver JSON szabványnak megfelelő adatként szolgáltatja a küldőnek. A POST és PUT kérésnél a rendszer JSON adatokat vár. A program biztonsági modulja úgy lett elkészítve, hogy minden felhasználóhoz tartoznak jogosultsági szintek. Azok a felhasználók, amelyeknél nincs megadva egyedi jogosultsági

szint, egy alapértelmezett szintet kapnak a rendszertől. A következő jogosultsági szintek léteznek, amelyek egyedi címkékkel vannak ellátva:

- **ROLE_USER:** Alapértelmezett jogosultsági szint.
- **ROLE_ADD_TASK:** Feladat kiosztása
- **ROLE_HOLIDAY_ADMIN:** Szabadságok kiosztása
- **ROLE_USER_ADMIN:** Felhasználók kezelése
- **ROLE_PROJECT_ADMIN:** Projektek kezelése
- **ROLE_ACTIVITY_GROUP_ADMIN:** Feladatok kezelése
- **ROLE_PERMISSION_ADMIN:** Jogosultságok kezelése
- **ROLE_LISTING:** Listázás
- **ROLE_ACTIVITY_PLAN_ADMIN:** Időbeosztás tervezése

Minden felhasználó rendelkezik egy vagy több jogosultsági címkével. A végpontok a címkék segítségével kerülnek levédésre. Ez azt jelenti, hogy miután a rendszer azonosított egy felhasználót, megnézi, hogy van-e jogosultsága kérést küldeni az adott végpontra. Ha igen, akkor kiszolgálja a kérését, ellenkező esetben pedig elutasításra kerül. A program 4 végponttal rendelkezik, amely felhasználó hitelesítés nélkül is elérhető. A következő végpontok kerültek implementálásra:

MainController

Végpont címe	Végpont típusa	Szükséges jogosultsági szint	Rövid leírás
/easy-log-in/activity-group/{id}	POST	-	Egyszerűsített tevékenység rögzítése
/logged-ever/{username}	GET	-	Bejelentkezett-e már valaha a felhasználó
/login	PUT	ROLE_USER	Bejelentkezés
/set-not-logged-user-password	POST	-	Újonnan regisztrált felhasználó jelszavának beállítása

ActivityController

Végpont címe	Végpont típusa	Szükséges jogosultsági szint	Rövid leírás
activity	GET	ROLE_LISTING	Megadja az összes tevékenységet
activity/add/task	POST	ROLE_USER, ROLE_ADD_TASK	Feladat rögzítése a feladatkezelő modulba
activity/delete/{id}	DELETE	ROLE_USER	Tevékenység törlése
activity/delete/task/{id}	DELETE	ROLE_USER	Feladat törlése
activity/edit	PUT	ROLE_USER	Tevékenység módosítása

activity/excel/project/{project}/activity-group/{activity-group}/user/{user}	GET	ROLE_LISTING	Excel fájlt készít a tevékenységekről a projekt, a feladat és a felhasználó szerint.
activity/new	POST	ROLE_USER	Tevékenység rögzítése
activity/task/complete	PUT	ROLE_USER	Feladat teljesített státuszra állítása
activity/task/me	GET	ROLE_USER	Megadja a felhasználó feladatait
activity/task/owned	GET	ROLE_ADD_TASK	Megadja a másoknak kiosztott feladatokat.
activity/year/{year}/month/{month}/day/{day}	GET	ROLE_USER	Megadja a felhasználó feladatait egy adott dátumra
activity-group	GET	ROLE_ACTIVITY_GROUP_ADMIN, ROLE_LISTING	Megadja a projektekhez rendelhető feladatokat
activity-group/{id}	DELETE	ROLE_ACTIVITY_GROUP_ADMIN	Projektekhez rendelhető feladatok törlése
activity-group/add	POST	ROLE_ACTIVITY_GROUP_ADMIN	Projektekhez rendelhető feladatok hozzáadása
activity-group/add/easy-log-in	POST	ROLE_ACTIVITY_GROUP_ADMIN	Egyszerűsített bejelentkezéshez feladat hozzáadása
activity-group/easy-log-in	GET	-	Megadja az egyszerűsített bejelentkezési feladatokat
activity-group/edit/work-group	PUT	ROLE_ACTIVITY_GROUP_ADMIN	Projektekhez rendelhető feladatok módosítása
activity-group/no/easy-log-in	GET	ROLE_ACTIVITY_GROUP_ADMIN	Megadja a nem egyszerűsített bejelentkezési feladatokat

ActivityPlanController

Végpont címe	Végpont típusa	Szükséges jogosultsági szint	Rövid leírás
activity-plan/me/owned	GET	ROLE_ACTIVITY_PLAN_ADMIN	Megadja a kiosztott időbeosztásokat
activity-plan/add	POST	ROLE_USER, ROLE_ACTIVITY_PLAN_ADMIN	Időbeosztás rögzítése/kiosztása
activity-plan/delete/{id}	DELETE	ROLE_USER, ROLE_ACTIVITY_PLAN_ADMIN	Időbeosztás törlése
activity-plan/edit	PUT	ROLE_USER	Időbeosztás módosítása
activity-plan/me	GET	ROLE_USER	A felhasználó időbeosztásait adja meg

HolidayController

Végpont címe	Végpont típusa	Szükséges jogosultsági szint	Rövid leírás
holiday	GET	ROLE_LISTING	Megadja az összes szabadságot
holiday/add/user/{id}	POST	ROLE_HOLIDAY	Szabadság rögzítése egy felhasználónak
holiday/delete/{id}	DELETE	ROLE_HOLIDAY	Szabadság törlése
holiday/excel/user/{user}/year/{year}	GET	ROLE_LISTING	Excel fájlt készít év és felhasználók szűrésével
holiday/me	GET	ROLE_USER	Megadja a felhasználó szabadságait
holiday/year	GET	ROLE_HOLIDAY, ROLE_LISTING	Megadja az évszámokat, amelyekre lett szabadság rögzítve az adatbázisba

PermissionController

Végpont címe	Végpont típusa	Szükséges jogosultsági szint	Rövid leírás
permission	GET	ROLE_PERMISSION_ADMIN	Megadja a jogosultsági csoportokat
permission/{id}/users	GET	ROLE_PERMISSION_ADMIN	Megadja egy jogosultsági csoporthoz rendelt felhasználókat
permission/add	POST	ROLE_PERMISSION_ADMIN	Új jogosultsági csoport rögzítése
permission/delete/{id}	DELETE	ROLE_PERMISSION_ADMIN	Jogosultsági csoport törlése
permission/edit	PUT	ROLE_PERMISSION_ADMIN	Jogosultsági csoport módosítása
permission/edit/{id}/user	PUT	ROLE_PERMISSION_ADMIN	Felhasználók jogosultsági csoporthoz rendelése
permission/edit/permission-detail	PUT	ROLE_PERMISSION_ADMIN	Jogosultsági csoportok összekapcsolása jogosultsági szintekkel

PermissionDetailController

Végpont címe	Végpont típusa	Szükséges jogosultsági szint	Rövid leírás
permission-detail	GET	ROLE_PERMISSION_ADMIN	Jogosultsági szintek listázása

UserController

Végpont címe	Végpont típusa	Szükséges jogosultsági szint	Rövid leírás
user	GET	ROLE_ADD_TASK, ROLE_USER_ADMIN, ROLE_PROJECT_ADMIN, ROLE_PERMISSION_ADMIN, ROLE_LISTING, ROLE_ACTIVITY_PLAN_ADMIN	Felhasználók listázása
user/add	POST	ROLE_USER_ADMIN	Új felhasználó rögzítése
user/delete/{id}	DELETE	ROLE_USER_ADMIN	Felhasználó törlése

user/edit	PUT	ROLE_USER_ADMIN	Felhasználó módosítása
user/holiday	GET	ROLE_HOLIDAY_ADMIN	Megadja egy felhasználó szabadságait

UserWorkGroupController

Végpont címe	Végpont típusa	Szükséges jogosultsági szint	Rövid leírás
user-work-group/user/{id}	GET	ROLE_USER	Megadja, hogy mely projektekhez van hozzárendelve a felhasználó
user-work-group/year/{year}/month/{month}/day/{day}	GET	ROLE_USER	Megadja dátum szerint egy felhasználó projektjeit

WorkGroupController

Végpont címe	Végpont típusa	Szükséges jogosultsági szint	Rövid leírás
work-group	GET	ROLE_PROJECT_ADMIN, ROLE_ACTIVITY_GROUP_ADMIN, ROLE_LISTING	Megadja a projekteket
work-group/{id}	DELETE	ROLE_PROJECT_ADMIN	Projekt törlése
work-group/{id}/users	GET	ROLE_PROJECT_ADMIN	Megadja egy projekt felhasználóit
work-group/add	POST	ROLE_PROJECT_ADMIN	Új projekt rögzítése
work-group/add/{id}/user-work-group	POST	ROLE_PROJECT_ADMIN	Projekt és felhasználók összekapcsolása
work-group/edit	PUT	ROLE_PROJECT_ADMIN	Projekt módosítása
work-group/edit/{id}/user-work-group	PUT	ROLE_PROJECT_ADMIN	Projektek és felhasználók összekapcsolásának módosítása

Az utolsó egység a szolgáltatások (services). Ebben a csomagban található az a szolgáltatás osztályok, amelyek segítségével az adatbázisból lekérdezett adatok feldolgozásra kerülnek. Ezek az osztályok használják a repository-kat, és segítségével kapjuk meg az adatokat. A legtöbb entitáshoz tartozik service osztály. A következő osztályok lettek megvalósítva:

- ActivityGroupService
- ActivityPlanService
- ActivityService
- HolidayService

- MyUserDetailsService
- PermissionDetailService
- PermissionService
- UserService
- UserWorkGroupService
- WorkGroupService

Az excel fájl elkészítése:

Az Excel fájlok generálása miatt volt szükséges egységesíteni az entitásokat. A BaseEntity egy absztrakt osztály, amely egyetlen egy absztrakt függvényt tartalmaz. Minden leszármazott osztályban implementálni kell az absztrakt függvényeket. Ennek a segítségével tudjuk majd lekérni az osztályok adattagjainak értékét. Az osztályokat kiegészítettem 2 statikus változóval. Az első tartalmazza azokat a fejléc címkéket, amelyek az Excel fájl első sorában kerülnek beszúrásra. A második pedig megadja, hogy az adott osztályban hány beszúrható elem szerepel.

Az Excel fájlok az ExcelMaker generikus osztály segítségével készülnek. A generikus osztály a BaseEntity-ből leszármaztatott osztályokat fogad el paraméternek. Az Excel fájlt az Apache POI osztályok segítségével hozza létre a program.

Minden elkészített Excel fájl lementésre kerül a szerveren, amíg a vezérlő nem továbbítja a kliens számára. A továbbítás után törlésre kerülnek.

A kliensoldal (Frontend) felépítése

A program felépítése

A kliensoldal megvalósítására az Angular keretrendszert választottam. A kliens program az NPM csomagkezelő és telepítő rendszerrel működik. Az Angular rendszer konfigurációs beállításai JSON fájlokban vannak tárolva. Az „angular.json” fájl határozza meg az NPM build rendszernek a mappaszerkezetet, és más konfigurációs fájlok helyét. Minden Angular projekt tartalmaz egy „package.json” állományt, amelyben felsorolásra kerültek azok a modulok, melyeket használ a program. A modulokhoz rögzítésre kerülnek a verziószámok. A verziószámok nem csak a rendszernek, hanem a

fejlesztőknek is sok információt szolgáltatnak. Egy csomag verziói között éles különbségek lehetnek, amelyekre a fejlesztés folyamán figyelni kell.

A projekten belül a fejlesztés folyamán több elem is létrehozható, ilyenek a class (osztály), component (komponens), service (szolgáltatás) elemek. Az osztályok a „classes” mappában lettek létrehozva. 9 osztály került megvalósításra. Az osztályokban szereplő adattagok részben megegyeznek azokkal az adattagokkal, amelyek a szerveroldali entitásokban kerültek létrehozásra. Ennek a fő oka, hogy amikor a szerver elküldi a kliens programnak JSON formátumban az adatokat, akkor az Angular automatikusan átalakítja a megfelelő formátumra. Előfordulhat olyan, hogy valamelyik oldalon egy vagy több adattag neve elírásra kerül, ilyenkor az üzenet fogadásakor azok az adattagok nem kapnak értékeket.

A komponensek az „ng generate component <komponens neve>” paranccsal hozhatók létre. Egy komponens generálásakor 3 fájl fog automatikusan létrejönni. Egy .html kiterjesztésű fájl, ami azokat a HTML tag elemeket fogja tartalmazni, amit meg szeretnénk jeleníteni a képernyőn. A második egy .css állomány, ami a html fájl stílus beállításaiért felel. Az utolsó pedig .ts, ami tartalmazni fogja azokat a TypeScript kódokat, amit a program végrehajt működése során. A programom fejlesztése folyamán 18 komponenst hoztam létre, amivel elkészítettem a kliensoldali rendszert.

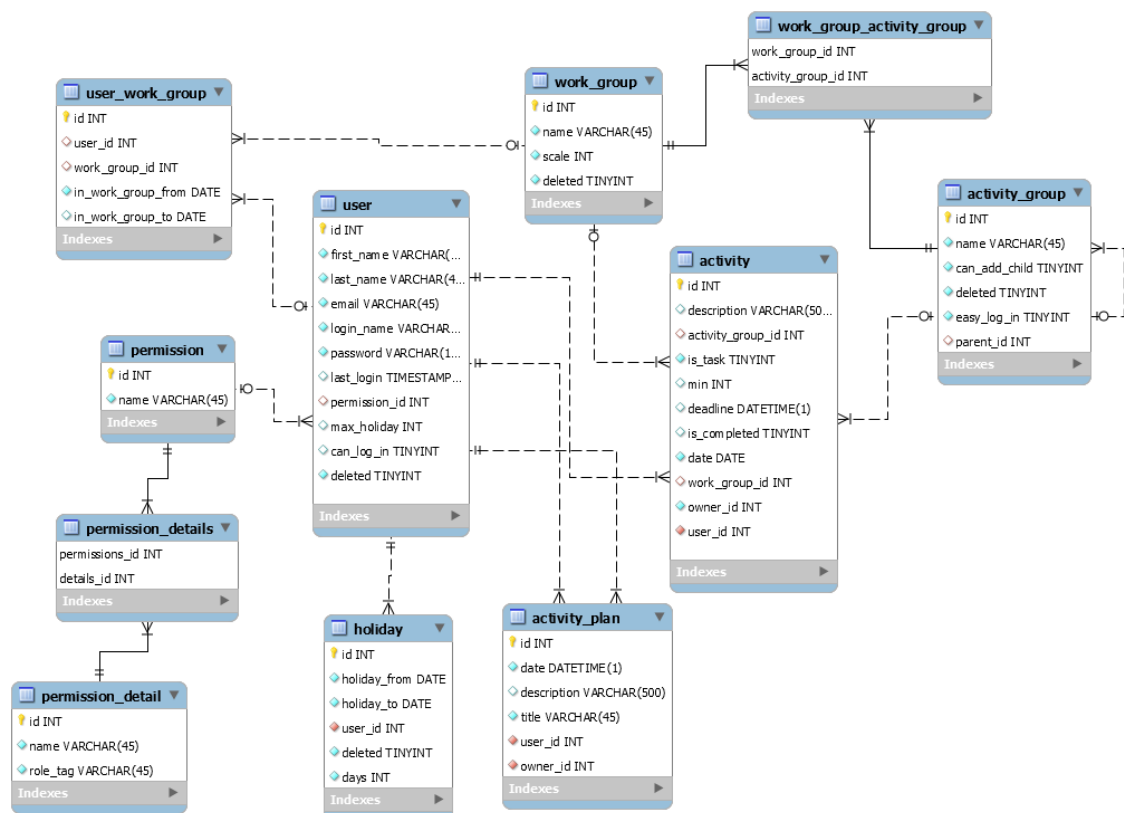
Minden komponenshez tartozik egy selector kifejezés, amely segítségével lehetőségünk van használni. Ezeket a selector kifejezéseket írjuk a HTML fájlokba, és ezek által hívódnak és jelenítődnek meg. A komponensek képesek egymás között kommunikálni.

A projektben szereplő szolgáltatások főként a szerver és a kliensoldal kommunikáció kialakítása érdekében lettek létrehozva. Az egyik legfontosabb szolgáltatás a HttpService, amely kezeli a kliens és szerver közötti kommunikációt. Generikus függvényekként írtam meg a GET, POST, PUT és DELETE kérés típusokat. Ez az állomány tartalmaz még egy „downloadFile” nevű metódust, amely az Excel fájlok letöltéséért felel. A program tartalmaz olyan szolgáltatásokat is, amelyek pl.: a felhasználó hitelesítését végzik, vagy olyanokat, melyek csak függvényeket tartalmaznak. Ezeket a függvényeket a komponensek használják. 13 service került megvalósításra.

Az adatbázis felépítése

A program MySQL relációs adatbázis rendszerrel működik, azonban, ha szeretnénk, kisebb átalakításokkal könnyedén csatlakoztatható más adatbázis rendszerekhez is. A fejlesztés folyamán a H2 relációs adatbázis rendszert használtam. Ez a rendszer rendelkezik egy úgynevezett „In-Memory” móddal. Ebben a módban az adatbázis fájlok fizikálisan nem jönnek létre a számítógépünkön, hanem csak betöltődnek a memória egységbe, és onnan dolgozik a rendszer. A program leállítása után minden információ elvész a memóriából. Az adatbázis tervezésénél, és a szerveroldali program alapjainak lefektetésekor érdemes használni. A program indításakor automatikusan generálódnak le a táblák úgy, hogy minden változtatást tartalmazzanak. Ez megkönnyíti és gyorsítja a fejlesztést, nem kell kézzel megadni minden változtatást. Az adatbázis tervezésre a MySQL Workbench programot használtam. A végleges adatbázis tervet egy .sql kiterjesztésű fájlba exportáltam. Az állományt később beimportáltam a lokálisan futtatott XAMPP MySQL adatbázisának phpmyadmin felületén.

Adatbázis terv:

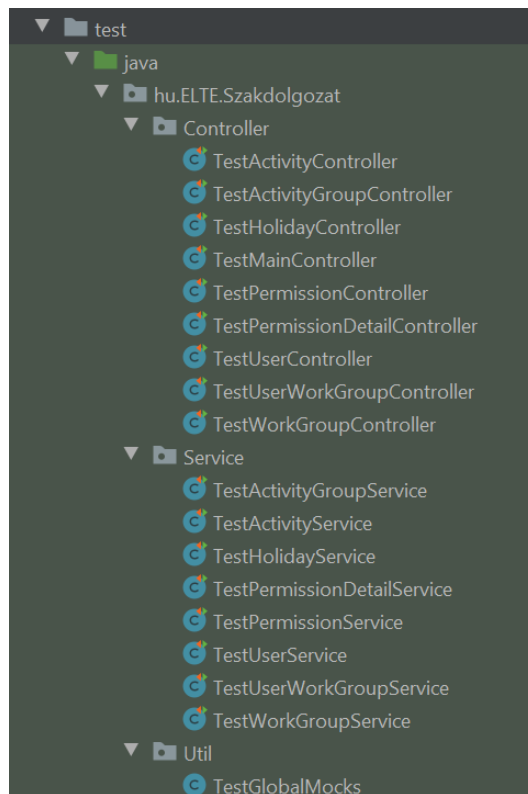


25. ábra: Adatbázis terv

A program elkészítéséhez 11 táblát kellett létrehoznom az adatbázisban. A táblák 1xN és NxN kapcsolatban vannak egymással. A 24. ábrán látható táblák kialakításához elkészítettem 2 SQL script-et. Az egyik felel a táblák létrehozásáért és a köztük lévő kapcsolat kiépítéséért. A második pedig a működéshez szükséges alap információkat tartalmazza. A program alapvető működéséhez fel kell tölteni a „PERMISSION_DETAIL” nevű táblát, amiben megtalálhatóak a jogosultsági szintek. Ezeket manuálisan nem ajánlott módosítani, mert a rendszer az ott rögzített „ROLE_TAG” elemeket használja.

Tesztelés

A program implementálása után elkészítettem 160 tesztet. Ezekben a tesztekben a vezérlők és a szolgáltatások helyes működését vizsgáltam. A vezérlőknél teszteltem, hogy helyesen viselkedik-e a program eltérően attól, hogy melyik felhasználó próbálja elérni az adott végpontokat. Azoknál a végpontoknál, ahol eltérő válaszokat kaphatunk a szervertől pl.: elutasítja a kérést, vagy adatot szolgáltat a kérést küldőnek. Külön teszteltem minden eshetőségre. A szolgáltatások esetében megnéztem, hogy az adatbázisból jövő adatokat helyesen dolgozza fel. Minden tesztfüggvényt próbáltam úgy elnevezni, hogy ha egy tesztrendszer futtatja, és valamiért megbukik a tesztelésen, akkor egyértelmű legyen a fejlesztő számára, hogy mi okozhatja a hibát. Ezek a tesztek a 26. ábrán található struktúrában helyezkednek el.



26. ábra: Gépi tesztfájlok

Manuális tesztelés:

Bejelentkezés:

- Bejelentkezési kísérlet felhasználónév és jelszó megadása nélkül.
Eredmény: sikertelen bejelentkezés
- Bejelentkezési kísérlet hibás felhasználónévvel és jelszóval.
Eredmény: sikertelen bejelentkezés
- Bejelentkezés helyes adatokkal.
Eredmény: sikeres bejelentkezés
- Újonnan regisztrált felhasználó jelszavának beállítása.
Eredmény: sikeres jelszóbeállítás

Egyszerűsített bejelentkezés:

- Bejelentkezés hiányzó felhasználói névvel.
Eredmény: sikertelen bejelentkezés
- Bejelentkezés hibás felhasználónévvel.
Eredmény: sikertelen bejelentkezés

- Bejelentkezés helyes felhasználónévvel.

Eredmény: sikeres bejelentkezés

Munkaidő-nyilvántartó:

- Tevékenység felvitele hiányzó adatokkal.

Eredmény: sikertelen tevékenység rögzítés

- Tevékenység felvitele helyesen kitöltött beviteli mezőkkel.

Eredmény: sikeres tevékenység rögzítés

- Tevékenység felvitelekor az idő mezőbe karakter beírás.

Eredmény: sikertelen tevékenység rögzítés

- Tevékenység módosítása.

Eredmény: a tevékenység módosításra került az adatbázisban

- Tevékenység módosításának visszavonása.

Eredmény: módosítás elvetésre kerül és a mezők alaphelyzetbe állnak

- Tevékenység törlése.

Eredmény: a tevékenység eltávolításra kerül az adatbázisból

- Naptárban a napok közötti váltás.

Eredmény: a kiválasztott napra berögzített tevékenységek listázásra kerültek

Szabadság tervező:

- Engedélyezett mennyiségű szabadság kiosztása.

Eredmény: sikeres szabadság rögzítés

- Az engedélyezetttnél több szabadság kiosztása.

Eredmény: sikertelen szabadság rögzítés

- Szabadság rögzítése hiányzó első és/vagy utolsó nap megadásával.

Eredmény: sikertelen szabadság rögzítés

- Szabadság rögzítése rossz évszám megadásával.

Eredmény: sikertelen szabadság rögzítés

- Szabadság törlése.

Eredmény: a szabadság egy deleted flag-et kap az adatbázisban

Időbeosztás tervező:

- Terv rögzítése hiányzó adatokkal.

Eredmény: sikertelen terv rögzítés

- Terv rögzítése helyes adatokkal.

Eredmény: a terv sikeres rögzítése

- Terv kiosztása más felhasználónak.
Eredmény: a terv sikeres rögzítése
- Terv módosítása.
Eredmény: a terv sikeresen módosításra került az adatbázisban
- Terv törlése.
Eredmény: a terv sikeresen eltávolításra került az adatbázisból
- A táblázatban egy tervre kattintás.
Eredmény: a terv adatainak listázása
- Más felhasználónak kiosztott terv törlése.
Eredmény: a terv eltávolításra került az adatbázisból

Listázás:

- Projektek/szabadságok listázása szűrőfeltételek nélkül.
Eredmény: projektek/szabadságok megjelennek a táblázatban
- Listázás szűrőfeltételekkel.
Eredmény: csak azok a projektek/szabadságok jelennek meg, melyek megfelelnek a szűrőfeltételeknek
- Szűrőfeltételek törlése.
Eredmény: a szűrőfeltételek törlődnek és minden adat megjelenik a táblázatban
- Excel fájl elkészítése szűrőfeltételek nélkül.
Eredmény: a programból letöltött fájlban minden adat megjelenik
- Excel fájl elkészítése szűrőfeltételekkel:
Eredmény: a programból letöltött fájlban csak azok az adatok jelennek meg, amelyek megfelelnek a szűrőfeltételeknek

Karbantartás – Felhasználók:

- Új felhasználó regisztrálása hiányzó adatokkal.
Eredmény: sikertelen felhasználóregisztráció
- Új felhasználó regisztrálása olyan felhasználónévvel, amely már szerepel az adatbázisban.
Eredmény: sikertelen felhasználóregisztráció
- Új felhasználó regisztrálása helyes adatokkal.
Eredmény: sikeres regisztráció
- Felhasználó adatainak módosítása helytelen adatokkal.
Eredmény: sikertelen módosítás

- Felhasználó adatainak módosítása helyes adatokkal.
Eredmény: sikeres módosítás
- Felhasználó módosításának visszavonása.
Eredmény: módosítások elvetésre kerültek, a mezők alaphelyzetbe állnak
- Felhasználó törlése.
Eredmény: a felhasználó deleted flag-et kap az adatbázisban

Karbantartás – Projektek:

- Új projekt rögzítése üres mezőkkel.
Eredmény: sikertelen rögzítés
- Új projekt rögzítése olyan névvel, amely már szerepel az adatbázisban.
Eredmény: sikertelen rögzítés
- Új projekt rögzítése helyes adatokkal.
Eredmény: sikeres rögzítés
- Projekt módosítása.
Eredmény: sikeres módosítás
- Projekt módosításának visszavonása.
Eredmény: a módosítások elvetésre kerültek, a mezők alaphelyzetbe állnak
- Projekt törlése.
Eredmény: a projekt deleted flag-et kap az adatbázisban

Karbantartás – Feladatok:

- Új feladat rögzítése hiányzó adatokkal.
Eredmény: sikertelen rögzítés
- Új feladat rögzítése olyan névvel, amely már szerepel az adatbázisban.
Eredmény: sikertelen rögzítés
- Új feladat rögzítése helyes adatokkal.
Eredmény: sikeres rögzítés
- Új egyszerűsített bejelentkezéshez tartozó feladat rögzítése
Eredmény: sikeres rögzítés
- Feladat törlése.
Eredmény: a feladat deleted flag-et kap az adatbázisban
- Feladat összekapcsolása egy projekttel.
Eredmény: sikeresen létrejön a kapcsolat

Karbantartás – Jogosultságok:

- Új jogosultsági csoport rögzítése név megadása nélkül.
Eredmény: sikertelen rögzítés
- Új jogosultsági csoport rögzítése olyan névvel, amely már szerepel az adatbázisban.
Eredmény: sikertelen rögzítés
- Új jogosultsági csoport rögzítése helyes adatokkal.
Eredmény: sikeres rögzítés
- Jogosultsági csoport nevének módosítása helytelen névvel.
Eredmény: sikertelen módosítás
- Jogosultsági csoport nevének módosítása helyes névvel.
Eredmény: sikeres módosítás
- Jogosultsági csoport törlése.
Eredmény: a csoport eltávolításra került az adatbázisból
- Csoport jogosultsági szintek és felhasználók beállítása.
Eredmény: sikeres összekapcsolás

Feladatkezelő:

- Új feladat rögzítése hiányzó adatokkal.
Eredmény: sikertelen rögzítés
- Új feladat rögzítése helyes adatokkal.
Eredmény: sikeres rögzítés
- Új feladat kiosztása más felhasználóknak.
Eredmény: sikeres feladatkiosztás
- Feladat törlése.
Eredmény: a feladat eltávolításra került az adatbázisból
- Kiosztott feladat törlése.
Eredmény: a feladat eltávolításra került az adatbázisból
- Feladat teljesítésre.
Eredmény: a feladat is_completed flag-et kap az adatbázisban.

ÖSSZEFOGLALÁS

Az alkalmazásom tervezése és implementálása során törekedtem korszerű technológiák bevonására, és egy olyan kód elkészítésére, amely könnyen olvasható fejlesztői szemmel. Minden funkciót úgy alakítottam ki, hogy letisztult és könnyen értelmezhető legyen a felhasználó számára. Célom volt, hogy mindent modulokra bontsak, és ezzel elősegítsem a további fejlesztéseket.

Egy program igazán sohasem készül el, mindig találhatunk benne olyan részeket, amelyeket „még jobbra” elkészíthetünk. A dolgozatom fejlesztése alatt jutottak nekem is eszembe újabb modulok, amelyekkel tovább lehetne javítani az alkalmazást. Az egyszerűsített bejelentkezést össze lehetne kötni egy beléptetőrendszerrel is. Ebben az esetben már nem kellene felhasználói nevet beírni, hanem egy mágneskártya adatait is rögzíthetnénk. Számos ehhez hasonló, érdekesebbnél érdekesebb részekkel lehetne továbbfejleszteni az alkalmazást.

ÁBRAJEGYZÉK

1. ábra: Bejelentkezési felület	9
2. ábra: Munkaidő-nyilvántartóban található naptár	10
3. ábra: Tevékenység rögzítésére szolgáló űrlap	11
4. ábra: Tevékenységek táblázatos megjelenítése.....	11
5. ábra: A szabadságterv rögzítésének űrlapja.....	13
6. ábra: Szabadságok megjelenítésére szolgáló táblázat.....	13
7. ábra: Időbeosztás tervező kinézete	14
8. ábra: Terv rögzítése	15
9. ábra: Kiosztott tervek listázása	15
10. ábra: Projektek listázása.....	16
11. ábra: Új felhasználó rögzítése.....	17
12. ábra: Felhasználó adatai részletesen.	17
13. ábra: Új projekt létrehozása	19
14. ábra: Egy projekt részletes adatai	19
15. ábra: Új feladat rögzítése	21
16. ábra: Feladatok listázása	22
17. ábra: Feladatok hozzárendelése egy projekthez.....	23
18. ábra: Új jogosultsági csoport rögzítése	23
19. ábra: Jogosultsági csoportok szerkesztése	24
20. ábra: Jogosultsági csoport törlésénél figyelmeztetés	25
21. ábra: Feladatkezelő	26
22. ábra: Feladat rögzítése	26
23. ábra: Kiosztott feladatok.....	27
24. ábra: Felhasználói esetek diagram	30
25. ábra: Adatbázis tervek	40
26. ábra: Gépi tesztfájlok	42

HIVATKOZÁSOK

Minden hivatkozás 2020.05.25-én elérhető volt és került a leírásba.

1. Java: <https://www.java.com/en/>
2. Node.js: <https://nodejs.org/en/>
3. Java spring: <https://spring.io/>
4. Angular CLI: <https://cli.angular.io/>
5. Angular Material: <https://material.angular.io/>
6. Bootstrap: <https://getbootstrap.com/>
7. MySQL: <https://www.mysql.com/>