**ZÁRÓDOLGOZAT**

Webes munka nyilvántartó rendszer

Tábori Bence László

2021, Tatabánya

Tartalomjegyzék

[**A rendszer célja** 5](#_Toc68335393)

[**A rendszerrel szemben elvárt követelmények** 5](#_Toc68335394)

[**Funkcionális követelmények** 5](#_Toc68335395)

[**Nem funkcionális követelmények** 5](#_Toc68335396)

[**A rendszer aktorai és jogosultságai** 6](#_Toc68335397)

[**Dolgozó** 8](#_Toc68335398)

[**Munkáltató** 9](#_Toc68335399)

[**Adatbázis** 9](#_Toc68335400)

[**Egyedtípusok** 9](#_Toc68335401)

[**Egyedtípus szerkezeti listák** 9](#_Toc68335402)

[**Users** 9](#_Toc68335403)

[**Users\_roles** 11](#_Toc68335404)

[**Users\_projektek** 11](#_Toc68335405)

[**Ugyfelek** 12](#_Toc68335406)

[**Ugyfelek\_projektek** 12](#_Toc68335407)

[**Projektek** 13](#_Toc68335408)

[**Munkak** 14](#_Toc68335409)

[**Koltsegek** 14](#_Toc68335410)

[**Tipusok** 15](#_Toc68335411)

[**Roles** 15](#_Toc68335412)

[**Kerelmek** 16](#_Toc68335413)

[**Elszamolasok** 16](#_Toc68335414)

[**Egyed-kapcsolat táblázat** 17](#_Toc68335415)

[**Kódjegyzék** 17](#_Toc68335416)

[**Adatszótár** 20](#_Toc68335417)

[**Adattáblák és kapcsolataik** 21](#_Toc68335418)

[**Inputok és outputok** 22](#_Toc68335419)

[**Inputok** 22](#_Toc68335420)

[**Outputok** 23](#_Toc68335421)

[**Alkalmazott fejlesztői eszközök** 24](#_Toc68335422)

[**Back-end** 24](#_Toc68335423)

[**Front-end** 25](#_Toc68335424)

[**Dokumentáció** 25](#_Toc68335425)

[**Szerver követelmények** 25](#_Toc68335426)

[**Szerkezeti felépítés** 25](#_Toc68335427)

[**Adatbázis** 26](#_Toc68335428)

[**Entity** 26](#_Toc68335429)

[**Repository** 26](#_Toc68335430)

[**Service** 26](#_Toc68335431)

[**Controller** 27](#_Toc68335432)

[**View** 27](#_Toc68335433)

[**Konfiguráció** 27](#_Toc68335434)

[**Osztályok** 27](#_Toc68335435)

[**Main osztály** 27](#_Toc68335436)

[**Konfigurációs osztályok** 27](#_Toc68335437)

[**MethodSecurityConfig** 28](#_Toc68335438)

[**WebConfig** 28](#_Toc68335439)

[**Controllerek** 28](#_Toc68335440)

[**ViewController** 28](#_Toc68335441)

[**PostController** 29](#_Toc68335442)

[**ApiController** 30](#_Toc68335443)

[**Entity-k** 31](#_Toc68335444)

[**User** 31](#_Toc68335445)

[**Ugyfel** 32](#_Toc68335446)

[**Projekt** 33](#_Toc68335447)

[**Munka** 34](#_Toc68335448)

[**Repository-k** 35](#_Toc68335449)

[**UserRepository** 35](#_Toc68335450)

[**UgyfelRepository** 36](#_Toc68335451)

[**ProjektRepository** 36](#_Toc68335452)

[**MunkaRepository** 36](#_Toc68335453)

[**Service-ek és ezek implementációi** 37](#_Toc68335454)

[**UserService és UserServiceImplementation** 37](#_Toc68335455)

[**UgyfelService és UgyfelServiceImplementation** 38](#_Toc68335456)

[**ProjektService és ProjektServiceImplementation** 39](#_Toc68335457)

[**MunkaService és MunkaServiceImplementation** 39](#_Toc68335458)

[**Kapcsolatok** 40](#_Toc68335459)

[**A rendszer funkciói részletesen** 42](#_Toc68335460)

[**Bejelentkezés** 42](#_Toc68335461)

[**Jelszó módosítás** 42](#_Toc68335462)

[**Elfelejtett jelszó** 42](#_Toc68335463)

[**Kérelem** 43](#_Toc68335464)

[**Dolgozó funkciói** 43](#_Toc68335465)

[**Munkaidő rögzítése** 43](#_Toc68335466)

[**Munkák megjelenítése** 44](#_Toc68335467)

[**Munka módosítása** 45](#_Toc68335468)

[**Tevékenység-diagram** 45](#_Toc68335469)

[**Munkáltató funkciói** 47](#_Toc68335470)

[**Dolgozó feltöltése** 47](#_Toc68335471)

[**Dolgozók megjelenítése** 48](#_Toc68335472)

[**Dolgozó módosítása** 48](#_Toc68335473)

[**Dolgozó munkáinak megtekintése és elszámolása** 49](#_Toc68335474)

[**Tesztelés** 49](#_Toc68335475)

[**Telepítés szerverre** 51](#_Toc68335476)

[**Adatbázis** 51](#_Toc68335477)

[**Levelezés** 52](#_Toc68335478)

[**Folyamat** 53](#_Toc68335479)

[**Felhasználói útmutató** 53](#_Toc68335480)

[**A program rövid ismertetése** 53](#_Toc68335481)

[**Műszaki követelmények** 53](#_Toc68335482)

[**Használati útmutató Alkalmazottaknak** 53](#_Toc68335483)

[**Jelszó módosítása** 54](#_Toc68335484)

[**Munkaidő rögzítése** 55](#_Toc68335485)

[**Munka módosítása** 57](#_Toc68335486)

[**Használati útmutató Munkáltatóknak** 58](#_Toc68335487)

[**Ügyfél feltöltése** 59](#_Toc68335488)

[**Ügyfél-projekt kötés rögzítése** 60](#_Toc68335489)

[**Ügyfél elszámolása** 61](#_Toc68335490)

[**Továbbfejlesztés lehetőségei** 61](#_Toc68335491)

[**Mellékelt fájlok** 63](#_Toc68335492)

[**Irodalomjegyzék** 66](#_Toc68335493)

# **A rendszer célja**

A rendszer fejlesztésének célja egy kisebb vállalkozás(10-20 fő) által használható web alapú munka nyilvántartó rendszer elkészítése.

# **A rendszerrel szemben elvárt követelmények**

## **Funkcionális követelmények**

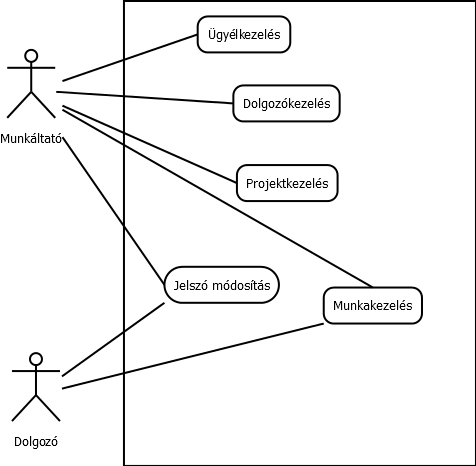
Alkalmazott szempontjából tegye lehetővé új munkák rögzítését, megtekintését, ezek szűrését évre és hónapra, valamint módosítását. Továbbá, jelszó módosítási lehetőséget.

Munkáltató szempontjából tegye lehetővé ügyfelek, dolgozók, projektek, illetve munkák kezelését. Dolgozók, illetve ügyfelek projekthez kötését és ezek kezelését.

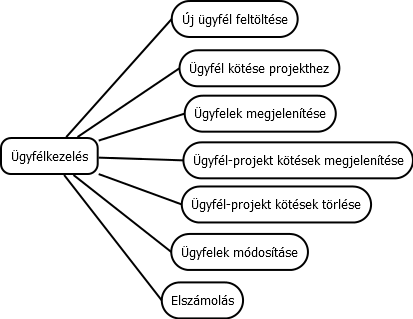
## **Nem funkcionális követelmények**

Biztosítsa, hogy az adatok ne kerülhessenek illetéktelen felhasználó kezébe. Mindenki csak azokat az adatokat láthassa, vagy módosíthassa, ami a feladatköréhez kötődik. Ne tároljon adatvédelmi szempontból nem megengedett adatot. A kezelőfelület használata ne igényeljen informatikai előismereteket.

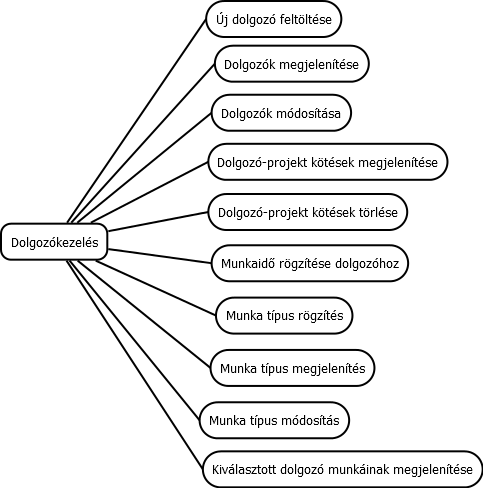
# **A rendszer aktorai és jogosultságai**



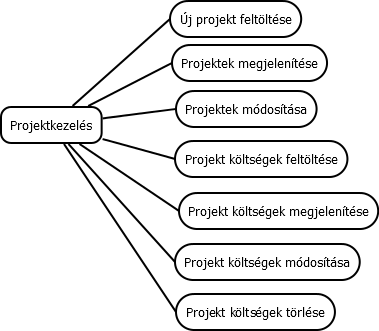
1. Teljes use-case modell



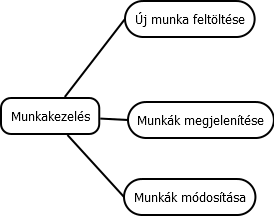
2. Ügyfélkezelés use-case



3. Dolgozókezelés use-case



4. Projektkezelés use-case



5. Munkakezelés use-case

## **Dolgozó**

A dolgozó szerepkörbe tartozó felhasználók az adatbázis csak egy kisebb részéhez férnek hozzá. Képesek munkát rögzíteni, a rögzített munkákat megtekinteni, illetve módosítani pontosan egy napig. Utóbbi a csalások elkerülése végett fontos. Továbbá, tudnak munkákat szűrni évre, hónapra és jelszót módosítani.

## **Munkáltató**

A munkáltató szerepkör a program legnagyobb szerepköre. Az adatbázis teljes olyan tartalmához hozzáfér, amit a programból manipulálni lehet, ilyen például az ügyfélkezelő, dolgozókezelő, projektkezelő és a munkakezelő.

# **Adatbázis**

Ez a fejezet tartalmazza az adatbázis logikai felépítését, beleértve az egyedtípusokat, a köztük lévő kapcsolatok, adatszótárt és kódjegyzéket.

## **Egyedtípusok**

A következő táblázat tartalmazza az összes egyedtípust.

|  |  |  |  |
| --- | --- | --- | --- |
| Egyed azonosító | Egyedtípus neve | Egyedtípus szöveges értelmezése | Terjedelem |
| E1 | users | Felhasználók adatai | ~10000 |
| E2 | users\_roles | Felhasználók szerepei | ~10000 |
| E3 | users\_projektek | Felhasználó-projekt kötések | ~999999 |
| E4 | ugyfelek | Ügyfelek adatai | ~99999 |
| E5 | ugyfelek\_projektek | Ügyfél-projekt kötés | ~999999 |
| E6 | projektek | Projektek adatai | ~99999 |
| E7 | munkak | Munkák adatai | ~999999 |
| E8 | koltsegek | Költségek adatai | ~99999 |
| E9 | tipusok | Munka típusok adatai | ~15 |
| E10 | roles | Szerepkörök | ~2 |
| E11 | kerelmek | Kérelmek adatai | ~1000 |
| E12 | elszamolasok | Elszámolások adatai | ~999999 |

1. Egyedtípus táblázat

## **Egyedtípus szerkezeti listák**

### **Users**

Users egyedtípus részletes leírása.

|  |  |  |  |
| --- | --- | --- | --- |
| E1 | users | | |
| Tulajdonság típus neve | Tulajdonság típus szöveges leírása | Intervallum | Típusa |
| id | Felhasználó azonosítója | 1-10000 | azonosító |
| aktiv | Aktív vagy sem | 1-2 | leíró |
| alaporaber | Felhasználó órabére | 1-10000 | leíró |
| bruttoszorzo | Felhasználóhoz tartozó bruttó szorzó | 0-2 | leíró |
| email | Felhasználó e-mail címe | 1-255 | leíró |
| password | Felhasználó jelszava | 1-255 | leíró |
| teljesnev | Felhasználó teljes neve | 1-255 | leíró |
| token | Feltöltéskor keletkező azonosító, ami kimegy e-mailben | 1-255 | leíró |
| created\_at | Feltöltés dátuma | 1001.01.01 00:00:00-3000.12.31 00:00:00 | leíró |
| updated\_at | Módosítás dátuma | 1001.01.01 00:00:00-3000.12.31 00:00:00 | leíró |

2. users szerkezeti lista

### **Users\_roles**

user\_roles egyedtípus részletes leírás.

|  |  |  |  |
| --- | --- | --- | --- |
| E2 | users\_roles | | |
| Tulajdonság típus neve | Tulajdonság típus szöveges leírása | Intervallum | Típusa |
| users\_id | Felhasználó azonosítója | 1-10000 | kapcsoló |
| roles\_id | Szerepkör azonosítója | 1-10000 | kapcsoló |

3. users\_roles szerkezeti lista

### **Users\_projektek**

users\_projektek egyedtípus részletes leírása

|  |  |  |  |
| --- | --- | --- | --- |
| E3 | users\_projektek | | |
| Tulajdonság típus neve | Tulajdonság típus szöveges leírása | Intervallum | Típusa |
| id | Kötés azonosítója | 1-10000 | azonosító |
| oraber | Kötéshez tartozó órabér | 1-100000 | leíró |
| projekt\_id | Projekt azonosítója | 1-100000 | kapcsoló |
| user\_id | Felhasználó azonosítója | 1-10000 | kapcsoló |

4. users\_projektek szerkezeti lista

### **Ugyfelek**

ugyfelek egyedtípus részletes leírása

|  |  |  |  |
| --- | --- | --- | --- |
| E4 | ugyfelek | | |
| Tulajdonság típus neve | Tulajdonság típus szöveges leírása | Intervallum | Típusa |
| id | Ügyfél azonosítója | 1-100000 | azonosító |
| aktiv | Aktív vagy sem | 1-2 | leíró |
| name | Ügyfél neve | 1-255 | leíró |
| created\_at | Feltöltés dátuma | 1001.01.01 00:00:00-3000.12.31 00:00:00 | leíró |
| updated\_at | Módosítás dátuma | 1001.01.01 00:00:00-3000.12.31 00:00:00 | leíró |

5. ugyfelek szerkezeti lista

### **Ugyfelek\_projektek**

ugyfelek\_projektek egyedtípus részletes leírása

|  |  |  |  |
| --- | --- | --- | --- |
| E5 | ugyfelek\_projektek | | |
| Tulajdonság típus neve | Tulajdonság típus szöveges leírása | Intervallum | Típusa |
| id | Kötés azonosítója | 1-1000000 | azonosító |
| oraber | Kötéshez tartozó órabér | 1-100000 | leíró |
| projekt\_id | Projekt azonosítója | 1-100000 | kapcsoló |
| ugyfel\_id | Ügyfél azonosítója | 1-100000 | kapcsoló |

6. ugyfelek\_projektek szerkezeti lista

### **Projektek**

projektek egyedtípus részletes leírása

|  |  |  |  |
| --- | --- | --- | --- |
| E6 | projektek | | |
| Tulajdonság típus neve | Tulajdonság típus szöveges leírása | Intervallum | Típusa |
| id | Projekt azonosítója | 1-100000 | azonosító |
| name | Projekt neve | 1-255 | leíró |
| leiras | Projekt leírása | 1-255 | leíró |
| status | Projekt státusza | 1-10 | leíró |
| type | Projekt típusa | 1-10 | leíró |
| tervezettoraber | Projekt tervezett órabére | 1-100000 | leíró |
| created\_at | Feltöltés dátuma | 1001.01.01 00:00:00-3000.12.31 00:00:00 | leíró |
| updated\_at | Módosítás dátuma | 1001.01.01 00:00:00-3000.12.31 00:00:00 | leíró |

7. projektek szerkezeti lista

### **Munkak**

munkak egyedtípus részletes leírása

|  |  |  |  |
| --- | --- | --- | --- |
| E7 | munkak | | |
| Tulajdonság típus neve | Tulajdonság típus szöveges leírása | Intervallum | Típusa |
| id | Munka azonosítója | 1-1000000 | azonosító |
| datum | Munka dátuma | 19 | leíró |
| megjegyzes | Munkához tartozó megjegyzés | 1-255 | leíró |
| munkaleiras | Munka leírása | 1-255 | leíró |
| raforditas | Munka ráfordítás | 0-24 | leíró |
| tipus\_id | Típus azonosítója |  | kapcsoló |
| kotes\_id | Kötés azonosítója |  | kapcsoló |
| created\_at | Feltöltés dátuma | 1001.01.01 00:00:00-3000.12.31 00:00:00 | leíró |
| updated\_at | Módosítás dátuma | 1001.01.01 00:00:00-3000.12.31 00:00:00 | leíró |

8. munkak szerkezeti lista

### **Koltsegek**

koltsegel egyedtípus részletes leírása

|  |  |  |  |
| --- | --- | --- | --- |
| E8 | koltsegek | | |
| Tulajdonság típus neve | Tulajdonság típus szöveges leírása | Intervallum | Típusa |
| id | Költség azonosítója | 1-100000 | azonosító |
| koltseg | Költség összege | 1-1000000 | leíró |
| megnevezes | Költség megnevezése | 1-255 | leíró |
| projekt\_id | Projekt azonosítója | 1-100000 | kapcsoló |
| created\_at | Feltöltés dátuma | 1001.01.01 00:00:00-3000.12.31 00:00:00 | leíró |
| updated\_at | Módosítás dátuma | 1001.01.01 00:00:00-3000.12.31 00:00:00 | leíró |

9. koltsegek szerkezeti lista

### **Tipusok**

tipusok egyedtípus részletes leírása

|  |  |  |  |
| --- | --- | --- | --- |
| E9 | tipusok | | |
| Tulajdonság típus neve | Tulajdonság típus szöveges leírása | Intervallum | Típusa |
| id | Típus azonosítója | 1-15 | azonosító |
| megnevezes | Típus megnevezése | 1-255 | leíró |

10. tipusok szerkezeti lista

### **Roles**

roles egyedtípus részletes leírása

|  |  |  |  |
| --- | --- | --- | --- |
| E10 | roles | | |
| Tulajdonság típus neve | Tulajdonság típus szöveges leírása | Intervallum | Típusa |
| id | Szerepkör azonosítója | 1-2 | azonosító |
| name | Szerepkör megnevezése | 1-255 | leíró |
| created\_at | Feltöltés dátuma | 1001.01.01 00:00:00-3000.12.31 00:00:00 | leíró |
| updated\_at | Módosítás dátuma | 1001.01.01 00:00:00-3000.12.31 00:00:00 | leíró |

11.roles szerkezeti lista

### **Kerelmek**

kerelmek egyedtípus részletes leírása

|  |  |  |  |
| --- | --- | --- | --- |
| E11 | kerelmek | | |
| Tulajdonság típus neve | Tulajdonság típus szöveges leírása | Intervallum | Típusa |
| id | Kérelem azonosítója | 1-1000 | azonosító |
| email | Milyen e-mailről jött a kérelem | 1-255 | leíró |
| tipus | Kérelem típusa | 1-255 | leíró |
| token | Kérelemhez tartozó generált azonosító | 1-255 | leíró |
| created\_at | Feltöltés dátuma | 1001.01.01 00:00:00-3000.12.31 00:00:00 | leíró |
| updated\_at | Módosítás dátuma | 1001.01.01 00:00:00-3000.12.31 00:00:00 | leíró |

12. kerelmek szerkezeti lista

### **Elszamolasok**

elszamolasok egyedtípus részletes leírása

|  |  |  |  |
| --- | --- | --- | --- |
| E12 | elszamolasok | | |
| Tulajdonság típus neve | Tulajdonság típus szöveges leírása | Intervallum | Típusa |
| id | Elszámolás azonosítója | 1-1000000 | azonosító |
| osszeg | Elszámolt összeg | 1-1000000 | leíró |
| munka\_id | Munka azonosítója | 1-1000000 | kapcsoló |
| datum | Elszámolás dátuma | 1001.01.01-3000.12.31 | leíró |

## **Egyed-kapcsolat táblázat**

Megfigyelhető, hogy több kapcsolatnál is van egy külön kötés egyedtípus, ami csak azonosítókat tartalmaz. Ezt azért csináltam így, mert például törlésnél nincs semmi kötöttség.

|  |  |
| --- | --- |
| Kapcsolat | Kapcsolat megnevezése |
| ugyfelek-ugyfelek\_projektek | Ügyfél projekthez kötés |
| ugyfelek\_projektek-projektek | Projekt ügyfélhez kötés |
| projektek-users\_projektek | Projekt felhasználóhoz kötés |
| projektek-koltsegek | Költség projekthez kötés |
| users\_projektek-users | Projekt felhasználóhoz kötés |
| roles-users\_roles | Szerepkör felhasználóhoz kötés |
| users\_roles-users | Felhasználó szerepkörhöz kötés |
| users\_projektek-munkak | Felhasználó-Projekt kötés munkához kötés |
| munkak-elszamolasok | Elszámolás munkához kötés |
| munkak-tipusok | Típusok munkához kötés |

13. egyed-kapcsolat táblázat

## **Kódjegyzék**

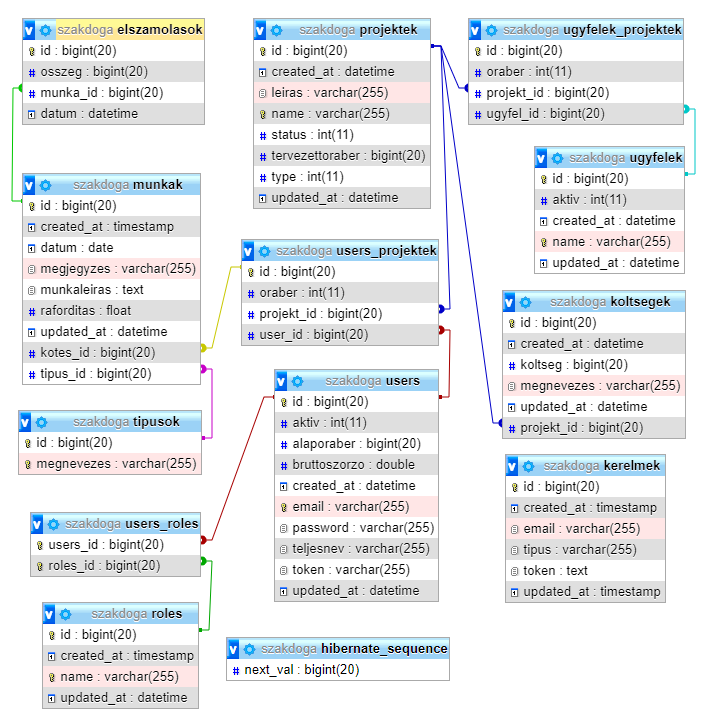
|  |  |  |  |
| --- | --- | --- | --- |
| Tulajdonság-típus neve | Kód szöveges értelmezése | Felvehető érték(ek) | Előfordulás |
| id(user) | A felhasználó azonosítására szolgál | egész számok | users, users\_projektek, users\_roles |
| aktiv | A felhasználó aktivitását tárolja | 0-3 közötti egész szám | users, ugyfelek |
| alaporaber | A felhasználóval megállapodott alap órabért tárolja | egész szám | users |
| bruttoszorzo | A felhasználóhoz tartozó bruttó szorzót tárolja | 0-2 közötti egész szám | users |
| email | E-mail cím-et tárol | létező e-mail cím | users, kerelmek |
| password | A felhasználó jelszavát tárolja | megadott jelszó | users |
| Tulajdonság-tipus neve | Kód szöveges értelmezése | Felvehető érték(ek) | Előfordulás |
| teljesnev | A felhasználó teljes nevét tárolja | A felhasználó teljes neve(Vezetéknév + keresztnév) | users |
| token | Véletlen generált karaktersorozatot tárol | véletlen generált karakterek | users, kerelmek |
| created\_at | A létrehozás dátumát tárolja | 1000-01-01 00:00:00 – 3001-12-31 | users, ugyfelek, projektek, munkak, koltsegek, roles, kerelmek |
| updated\_at | A módosítás dátumát tárolja | 1000-01-01 00:00:00 – 3001-12-31 | users, ugyfelek, projektek, munkak, koltsegek, roles, kerelmek |
| id(users\_projektek) | A User-projekt kötés azonosítására szolgál | egész szám | users\_projektek |
| oraber | Órabér tárolását | egész szám |  |
| id(ugyfelek) | Ügyfél azonosítására szolgál | egész szám | ugyfelek, ugyfelek\_projektek |
| name | Nevek tárolására szolgál | Ügyfél teljes neve(Vezetéknév + keresztnév) | ugyfelek, projektek, roles |
| id(ugyfelek\_projektek) | Ügyfél-projekt kötés azonosítására szolgál | egész szám | ugyfelek\_projektek |
| id(projektek) | Projekt azonosítására szolgál | egész szám | projektek, users\_projektek, ugyfelek\_projektek |
| leiras | Projekt leírását tárolja | szöveg | projektek |
| status | Projekt státuszát tárolja | 0-2 egész szám | projektek |
| type | Projekt típusát tárolja | 1-4 egész szám | projektek |
| Tulajdonság-típus neve | Kód szöveges értelmezése | Felvehető érték(ek) | Előfordulás |
| tervezettoraber | Projekt tervezett órabérét tárolja | egész szám | projektek |
| id(munkak) | Munka azonosítására szolgál | egész szám | munkak, elszamolasok |
| datum | Rögzített dátumot tárol | 1000-01-01 00:00:00 – 3001-12-31 | munkak, elszamolasok |
| megjegyzes | Munkához tartozó megjegyzést tárol | szöveg | munkak |
| munkaleiras | Munka leírását tárolja | szöveg | munkak |
| raforditas | Munkára való ráfordítás-t tárol | 0.25-23.75 | munkak |
| id(koltseg) | Költség azonosítására alkalmas | egész szám | koltsegek |
| koltseg | Költség összegét tárolja | egész szám | koltsegek |
| megnevezes | Megnevezéseket tárol | szöveg | koltsegek, tipusok |
| id(roles) | Szerepkör azonosítására szolgál | egész szám | roles, users\_roles |
| id(kerelmek) | Kérelem azonosítására szolgál | egész szám | kerelmek |
| id(elszamolasok) | Elszámolás azonosítására szolgál | egész szám | elszamolasok |
| osszeg | Összeg forintban | egész szám | elszamolasok |

## **Adatszótár**

|  |  |  |
| --- | --- | --- |
| Tulajdonság-típus | Megnevezés | Előfordulás |
| Név |
| id(user) | A dolgozó azonosítója | E1, E2, E3 |
| aktiv | A felhasználó aktivitása | E1, E4 |
| alaporaber | A dolgozóval megegyezett alap órabér | E1 |
| bruttoszorzo | A dolgozó bruttó szorzója | E1 |
| email | Felhasználó e-mail címe | E1, E11 |
| password | Felhasználó jelszava | E1 |
| teljesnev | Felhasználó teljes neve | E1 |
| token | Új felhasználó létrehozásakor és kérelem elküldésekor létrejövő véletlen generált karaktersorozat | E1, E11 |
| created\_at | Létrehozás dátuma | E1, E4, E6, E7, E8, E10, E11 |
| updated\_at | Módosítás dátuma | E1, E4, E6, E7, E8, E10, E11 |
| id(users\_projektek) | Felhasználó-projekt kötés azonosítója | E3 |
| oraber | Kötéshez tartozó órabér | E5 |
| id(ugyfelek) | Ügyfél azonosítója | E4, E5 |
| name | Név | E4, E6, E10 |
| id(ugyfelek\_projektek) | Ügyfél-projekt kötés azonosítója | E5 |
| id(projektek) | Projekt azonosítója | E8, E3, E5, E6 |
| leiras | Projekt leírása | E6 |
| status | Projekt státusza | E6 |
| type | Projekt típusa | E6 |
| tervezettoraber | Projekt tervezett órabére | E6 |
| id(munkak) | Munka azonosítója | E7 |
| datum | Dátum | E7 |
| megjegyzes | Megjegyzés | E7 |
| munkaleiras | Munka leírása | E7 |
| raforditas | Munkára ráfordított idő | E7 |
| id(koltseg) | Költség azonosítója | E8 |
| koltseg | Költség forintban | E8 |
| megnevezes | Megnevezés | E8, E9 |
| id(roles) | Szerepkör azonosítója | E10 |
| Tulajdonság-típus | Megnevezés | Előfordulás |
| Név |
| id(kerelmek) | Kérelem azonosítója | E11 |
| id(elszamolasok) | Elszámolás azonosítója | E12 |
| osszeg | Elszámolt összeg forintban | E12 |

## **Adattáblák és kapcsolataik**

A következő kép tartalmazza az adattáblák közötti kapcsolatokat



30. Adatbázis és kapcsolatok

# **Inputok és outputok**

Ebben a fejezetben be fogom mutatni a rendszer összes inputját és outputját. Ezeket az adatokat táblázatok segítségével fogom bemutatni.

## **Inputok**

Ez a táblázat tartalmazza a rendszer összes inputját.

|  |  |  |  |
| --- | --- | --- | --- |
| A rendszer-input | | Felhasználói funkció | |
| Azonosító | Név | Azonosító | Név |
| I1 | Bejelentkezés űrlap | IFF1 | Bejelentkezés |
| I2 | Ügyfél feltöltése űrlap | IFF2 | Új ügyfél rögzítése adatbázisba |
| I3 | Ügyfél módosítása űrlap | IFF3 | Már rögzített ügyfél adatainak módosítása |
| I4 | Ügyfél aktivitása jelölőnégyzet | IFF4 | Már rögzített ügyfél aktivitásának állítása |
| I5 | Ügyfél-projekt kötés űrlap | IFF5 | Ügyfél projekthez kötésének rögzítése az adatbázisba |
| I6 | Ügyfél-projekt kötés törlés gomb | IFF6 | Már rögzített ügyfél-projekt kötés törlése |
| I7 | Elszámolás űrlap | IFF7 | Elszámolás táblázatok generálása, bevitt adatok alapján |
| I8 | Dolgozó feltöltése űrlap | IFF8 | Új dolgozó feltöltése az adatbázisba |
| I9 | Dolgozó módosítása űrlap | IFF9 | Már rögzített dolgozó adatainak módosítása |
| I10 | Dolgozó aktivitása jelölőnégyzet | IFF10 | Már rögzített dolgozó aktivitásának állítása |
| I11 | Dolgozó-projekt kötés űrlap | IFF11 | Dolgozó projekthez kötésének rögzítése az adatbázisba |
| I12 | Munkaidő feltöltése űrlap | IFF12 | Új munkaidő feltöltése az adatbázisba. |
| I13 | Munka típus feltöltése űrlap | IFF13 | Új munka típus feltöltése az adatbázisba |
| I14 | Dolgozó-projekt kötés törlése gomb | IFF14 | Már rögzített kötés törlése az adatbázisból |
| I15 | Munka típus módosítás űrlap | IFF15 | Már rögzített munka típus adatainak módosítása |
| I16 | Projekt feltöltése űrlap | IFF16 | Új projekt feltöltése az adatbázisba |
| I17 | Költség feltöltése űrlap | IFF17 | Új költség feltöltése az adatbázisba projekthez kötve |
| I18 | Projekt módosítása űrlap | IFF18 | Már rögzített projekt módosítása |
| I19 | Projekt költség módosítása űrlap | IFF19 | Már rögzített projekt költség módosítása |
| I20 | Projekt költség törlése gomb | IFF20 | Már rögzített projekt költség törlése az adatbázisból |
| I21 | Munkák módosítása űrlap | IFF21 | Már rögzített munka módosítása |
| I22 | Munkák törlése gomb | IFF22 | Már rögzített munkák törlése az adatbázisból |
| I23 | Jegyezz meg gomb | IFF23 | Belépésnél jegyezz meg gomb. Ilyenkor belépve marad a felhasználó |
| I24 | Elfelejtett jelszó űrlap | IFF24 | Belépésnél jelszó módosítása |
| I25 | Jelszó változtatási kérelem gomb | IFF25 | Dolgozó oldalon jelszó változtatási kérelem küldése |
| I26 | Munka szűrés űrlap | IFF26 | Dolgozó munka oldalon és munkáltató munka megtekintés oldalon év-re és hónapra munkák szűrése |

14. A rendszer összes inputja

## **Outputok**

A következő táblázat tartalmazza a rendszer összes outputját.

|  |  |  |  |
| --- | --- | --- | --- |
| A rendszer-input | | Felhasználói funkció | |
| Azonosító | Név | Azonosító | Név |
| O1 | Ügyfél táblázat | OFF1 | Ügyfelek adatainak megjelenítése |
| O2 | Ügyfél-projekt kötés táblázat | OFF2 | Ügyfelek projektekhez kötésének megjelenítése |
| O3 | Dolgozó táblázat | OFF3 | Dolgozók adatainak megjelenítése |
| O4 | Dolgozó-projekt táblázat | OFF4 | Dolgozók projektekhez kötésének megjelenítése |
| O5 | Munka típusok táblázat | OFF5 | Munkák típusainak megjelenítése |
| O6 | Projekt táblázat | OFF6 | Projektek adatainak megjelenítése |
| O7 | Projekt költségei táblázat | OFF7 | Projektek költségeinek megjelenítése |
| O8 | Munkák táblázat | OFF8 | Minden munka adatainak megjelenítése |
| O9 | Dolgozónként munkák táblázat | OFF9 | Dolgozó oldalról munkák megjelenítése bejelentkezett dolgozó alapján, munkáltató oldalról kiválasztott dolgozó munkáinak megjelenítése |
| O10 | Munka összesítő táblázat | OFF10 | Dolgozó és munkáltató oldalon a munkák táblázat alatt összesített adatok megjelenítése |
| O11 | Elszámolás táblázat | OFF11 | Munkák összesítése ügyfélre, projektre, évre és hónapra lebontva táblázatban megjelenítve |

15. A rendszer összes outputja

# **Alkalmazott fejlesztői eszközök**

## **Back-end**

Programozási nyelv: Java 15

Keretrendszer: Spring boot

Integrált fejlesztői környezet: Spring Tool Suite 4.0.9

Csomagkezelő/builder: Maven 3.6.3

Függőségek:

* nz.net.ultraq.thymeleaf
* org.springframework.security
* org.springframework.boot
* org.thymeleaf.extras
* com.h2database
* mysql
* org.postgresql
* com.fasterxml.jackson.core

Adatbázis:

* Típus: MySQL, PostgreSQL
* Szerver:
  + **MySQL** esetén: XAMPP v3.2.4 (Apache, MySQL)
  + **PostgreSQL** esetén: Heroku (https://www.heroku.com/)

## **Front-end**

Programozási nyelv: Javascript, CSS, HTML

Keretrendszerek: Bootstrap 5, JQuery

Futtatási környezet: Firefox, Edge, Chrome böngészők

## **Dokumentáció**

Szövegszerkesztő program: Microsoft Word

Képszerkesztő program: Photoshop, Paint 3D, Képmetsző, Dia

# **Szerver követelmények**

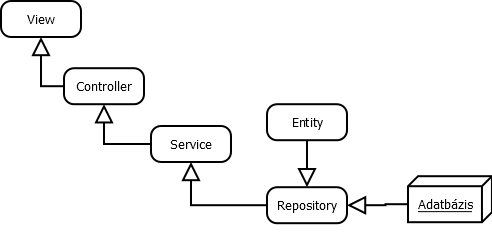
**Minimum:**

* Processzor: Intel Xeon W-2123
* Memória: 2 GB DDR4
* Tárhely: 50 GB SSD
* Operációs-rendszer: Windows Server 2016

**Ajánlott:**

* Processzor: Intel Xeon Scalable Gold vCPU @2,6 GHz
* Memória (RAM): 4 GB DDR4
* Tárhely: 100 GB SSD
* Operációs-rendszer: Windows Server 2016

# **Szerkezeti felépítés**



6. Rétegek felépítése

Azért hoztam létre több réteget, mert így biztonságosabb az adatfolyam, mivel a view-ból érkező adat nem konkrétan lép kapcsolatba a repository-val(így lényegében az adatbázissal), hanem egy service rétegen keresztül. Minden rétegnek létrehoztam egy saját csomagot a main csomag alcsomagjaként.

## **Adatbázis**

MySQL alapú relációs adatbázis. A program tábláiban megtalálható összes adatot az adatbázis tárolja. Ezekre az adatokra hatással lehetnek szerepkörtől függően a felhasználók.

A továbbiakban látható lesz, hogy az adatbázis táblák modelljei különböző osztályokban (entitykben) vannak legyártva. A Spring boot sajátossága az, hogy nincs szükség külső adatbázis fájlra, mivel induláskor (ha úgy állítjuk be), létrehozza az adatbázist az osztályok és a köztük lévő kapcsolatok alapján.

Ez azért előny, mert így nincs szükség az adatbázis külön konfigurálására. Továbbá, ha szükséges, akkor külön SQL utasításokat is kapcsolhatunk az induláshoz.

## **Entity**

Minden egyes entity-nek bejegyzett osztály egy adatbázis tábla modellje, objektum-orientált formában. Erre azért van szükség, hogy később könnyebben kezelhető legyen, bármilyen adatbázisból lekérdezett vagy feltölteni kívánt adat a repository-n keresztül. Minden entity egy @Entity annotációval van ellátva, melynek segítségével a container képes bejegyezni az adott osztályt, mint entity. Az entity-k az **entity** nevezetű alcsomagban vannak.

## **Repository**

A repository-k közvetlen kapcsolatban állnak az adatbázissal. Ezek lényegében interfacek, amik ki vannak terjesztve egy ős által. Egy adott repository azzal az adatbázis táblával áll kapcsolatban, amire beállítjuk egy entity segítségével, ami már csatolva van egy adott táblához. Ehhez szükséges az úgy nevezett CrudRepository interface, ami be van építve a Spring boot-ba. Ezáltal az interface által van kiterjesztve az összes repository. Ebben lényegében alapvető adatbázis utasítások találhatóak, függvények formájában. Minden entity egy @Repository annotációval van ellátva, melynek segítségével a container képes bejegyezni az adott osztályt, mint repository. A repository-k a **repository** nevezetű alcsomagban vannak.

## **Service**

A service réteg az adatok feldolgozásáért felel. Ezt én két részre bontottam. Egy interface-re és egy implementációra. Az inteface implementálva van a hozzá tartozó implementációs osztályba. Ez egyrészt azért hasznos, mert egy service implementációhoz tartozhat többféle interface is, másrészt azért, mert így sokkal rendezettebb és átláthatóbb lesz a kód. Minden service egy @Service annotációval van ellátva, melynek segítségével a container képes bejegyezni az adott osztályt, mint service. A service-ek a **service** nevezetű alcsomagban vannak.

## **Controller**

A programban a controller szerepe csak adattovábbítás és fogadás. Három féle controller-t hoztam létre, amik a következők:

* ViewController
* PostController
* ApiController

Ezek a controllerek a **controller** nevezetű alcsomagban vannak.

## **View**

Ez a réteg felel a megjelenítésért. Ide tartoznak például a különböző html, css, javascript fájlok, illetve a spring boot-ba beépített thymeleaf kiegészítő is. A view-val kapcsolatos file-ok(html,css,javascript) a **resources** mappában megtalálhatóak.

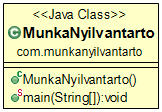
## **Konfiguráció**

A rendszer egyedi konfigurációi az **application.properties** konfigurációs fájlban állíthatóak. Ilyen lehet például az adatbázis elérés, szerver port, logger szint vagy esetleg a programban használt függőségek beállításai.

# **Osztályok**

## **Main osztály**

Itt található a main függvény, ezért ez a program belépési pontja. Minden adat ezen keresztül megy át.



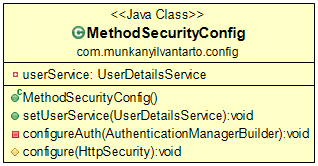
7. A main osztály felépítése

## **Konfigurációs osztályok**

Minden konfigurációs osztály a **config** nevezetű alcsomagban van.

### **MethodSecurityConfig**

Ez az osztály foglalkozik az útvonalak kezelésével. Itt állítottam be azt, hogy egy adott útvonalat milyen szereppel rendelkező személy érhet el. Ennek az osztálynak az őse a Spring boot-ba beépített jogosultság kezelő osztály.



8. Az útvonal kezelésével foglalkozó osztály

### **WebConfig**

Ez az osztály foglalkozik a Spring MVC testreszabásával. Jelen esetben itt a kezdőképernyőt állítottam be a login-ra. Ennek az osztálynak az őse a WebMvcConfigurer osztály, ami egy Spring boot-ba beépített osztály.

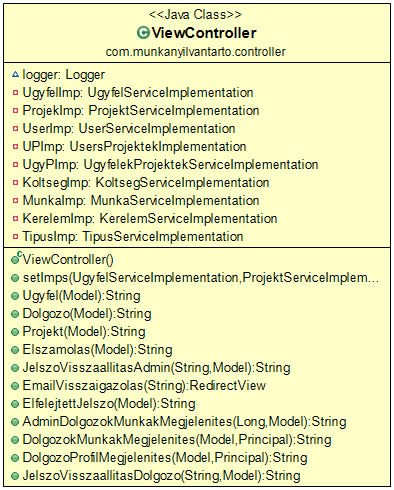


9. A Spring MVC testreszabásával foglalkozó osztály

## **Controllerek**

### **ViewController**

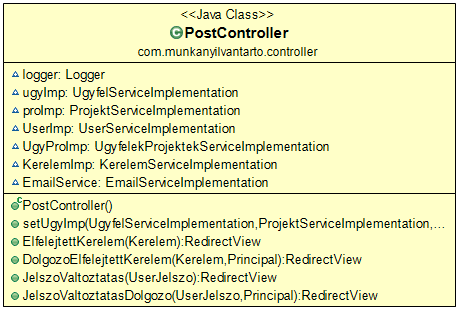
Ez a controller felel a megjelenítésért. Az ebben lévő függvényekhez tartozik egy útvonal és egy html fájl, amit meg kell jelenítenie. Ezen felül, képes adatokat is átadni a view-nak, mint például egy táblázat sorait vagy üres objektumokat amiket fel tud tölteni a felhasználó egy form-ban. Adott függvényhez tartozó útvonalat annotáció segítségével lehet megadni. Üres objektumként adhatunk át entity-ket is, így a PostControllerben már csak át kell adni a service rétegnek adatfeldolgozásra.



10. ViewController felépítése

### **PostController**

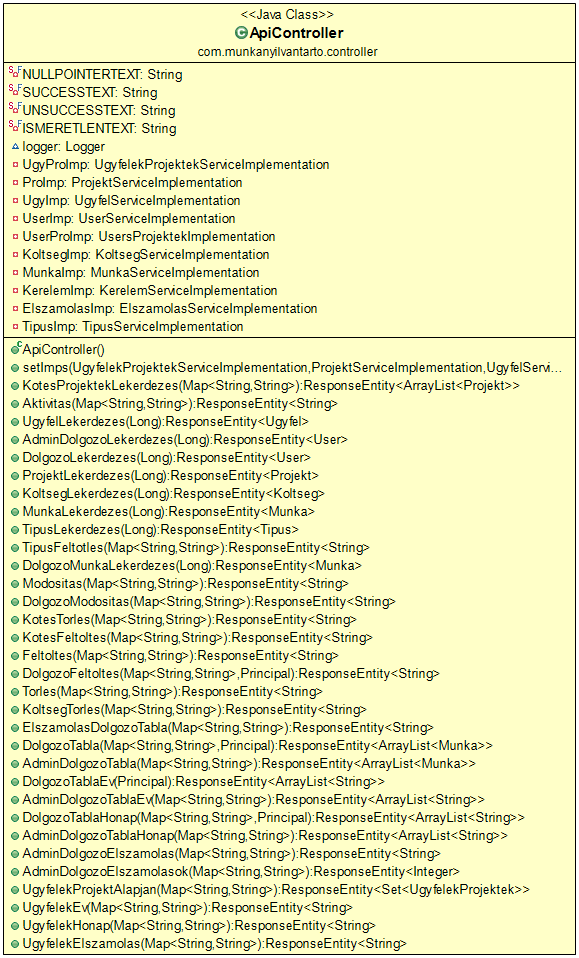
Ez a controller csak a view-ból közvetlenül beérkező adatokkal foglalkozik. Ebben is minden külön függvény egy adott útvonalhoz van csatolva, annotáció segítségével. Ez a controller képes fogadni a ViewController-ben elindított üres objektum, felhasználó által feltöltött verzióját, miután a felhasználó elküldi azt.



11. PostController felépítése

### **ApiController**

Ez a controller tartalmazza az összes web service-t. Ez azért érdekes, mert képes kommunkálni más külső programokkal és adatokat tud átadni nekik és fogadni tőlük. Ezen controller segítségével közvetetten lehet kezelni ugyanazt az adatbázist, így lehetővé teszi több és különböző platformon történő továbbfejlesztést. Illetve, ez a legnagyobb osztály.



12. ApiController felépítése

## **Entity-k**

A következőkben bemutatok néhány fontosabb entity-t. Mindegyik entity-nek van egy egyedi azonosítója. Ezt az azonosítót tárolja az id attribútum. Továbbá, megfigyelhető, hogy minden entity tartalmaz egy üres, publikus konstruktort, amire azért van szükség, hogy a repository el tudja érni a benne lévő adatokat, illetve mindegyik tartalmaz gettereket és settereket, ugyanis például a setterek segítségével képes feltölteni a az objektumot a repository a bejövő adatokkal. Ha egy entity-t egy másikhoz kötünk, akkor egy adott lekérdezésnél a repository fel tudja tölteni a kapcsolt entity-t is a megfelelő adatokkal. Példa erre a Projekt entity-ben látható „koltseg” attribútum, aminek a típusa egy Set, amiben a kötött(Koltseg) entity példányait tárolja, attól függően, hogy melyik van a Projekt entity-hez kötve. Jelen példában tehát ez tárolja az adott projekt összes költségét.

### **User**

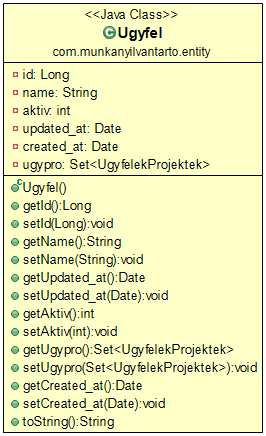
Ez a legnagyobb entity, illetve ez a user adatbázis tábla modellje. Tárolja az adott felhasználó összes adatát. Ilyen például az e-mail cím, jelszó, felhasználónév és szerepkör. A User entity kapcsolatbam van más entitykkel is, mint például a UsersProjektek entity, ami lényegében egy kötés táblának a modellje, ezért ez csak kapcsolatokat tárol néhány leíró adattal. Ennek a kapcsolatnak a típusa 1:N. Továbbá, N:M kapcsolatban áll még a Role entity-vel, ami a szerepkörök adatait tartalmazza.



13. User entity

### **Ugyfel**

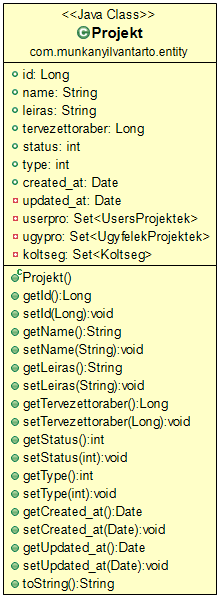
Ez az ugyfel adatbázis tábla modellje. Tárolja az összes adatot egy adott ügyfélről, mint például a neve és az aktivitása. Ez az entity szintén kapcsolatban van egy másik entity-vel az UgyfelekProjektek-kel, ami szintén csak kapcsolatokat tárol. Ebben az esetben a kapcsolat 1:N típusú az Ugyfel entity oldaláról, viszont a túloldalról ez N:1.



14. Ugyfel entity

### **Projekt**

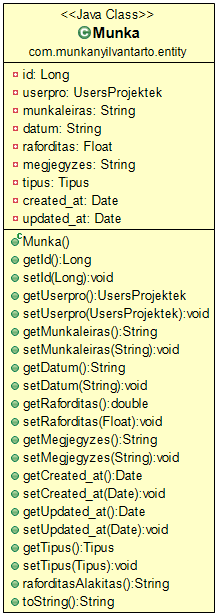
Ez a projekt adatbázis tábla modellje. Tárolja az összes adatot egy adott projektről. Ez a második legnagyobb entity, viszonylag sok kapcsolattal és adattal. Kapcsolatban van például az UgyfelekProjektek-kel, az UsersProjektek-kel és a Koltseg-ekkel is. Mindegyik kapcsolat a Projekt entity oldaláról 1:N, viszont a túloldalról N:1.



15. Projekt entity

### **Munka**

Ez a munka adatbázis tábla modellje. Tárolja egy adott munka minden adatát, illetve kapcsolatait. Kapcsolatban van a UsersProjektek táblával. A kapcsolat típusa N:1.



16. Munka entity

## **Repository-k**

Ebben az alfejezetben bemutatom a fentebb részletezett entity-khez tartozó repository-kat. Ezekben a repository-kban található minden metódus, ami adatbázis utasítások végrehajtására jó. A CrudRepository-nak alapból két paramétert kell megadni. Egy adott entity-t, amit kötni szeretnénk hozzá és az adott entity azonosítójának típusát. Ezt figyelembe véve, minden repository létrehozásánál megadtam ezeket a paramétereket. Ebben az alfejezetben megfigyelhető lesz, hogy egy adott metódus nevében, ugyanazok a kifejezések szerepelnek, mint például a „find” vagy a „By”. Ezek azok a kulcsszavak, amiket a legtöbbször használok egy metódus nevében. Ez egy különlegessége a Spring boot-nak, ugyanis futáskor a repository-ban létrehozott metódus neveket átalakítja sql lekérdezésekké. Az egyszerűbb lekérdezéseket ezzel a módszerrel oldottam meg.

### **UserRepository**

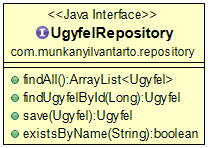
Minden user táblán elvégzendő adatbázis műveletre létrehoztam 1-1 metódust. Ebben a repository-ban csak lekérdezések találhatóak. Ezek a következő ábrán láthatóak:



17. User repository

### **UgyfelRepository**

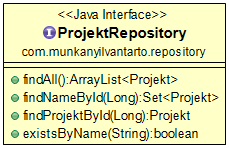
Ez a repository kezeli az összes ugyfel táblán elvégzendő adatbázis műveletet. A lentebbi ábrán látható, hogy itt már nem csak lekérdező függvények vannak, hanem manipuláló és létezés ellenőrző függvények is. Ilyen például a „save”, ami értelemszerűen a mentésre vonatkozik. Ennél a metódusnál paraméterként mindig meg kell adni azt az entity-t, ami a repository-hoz kötve van. Jelen esetben ez az Ugyfel entity. Továbbá ott van az „exists” szóval kezdődő függvény, ami a létezésre utal. Jelen esetben a metódus megvizsgálja, hogy név alapján létezik-e adott sor az adatbázisban. Paraméterként pedig meg kell adnunk, hogy milyen típusú az az adat, ami szerint szeretnénk futtatni a keresést. Jelen esetben a név egy szöveg típusú adat, ezért a paraméter típusa String.



18. UgyfelRepository osztálydiagramja

### **ProjektRepository**

A projekt táblát ez a repository kezeli. Csatlakoztatva van hozzá a Projekt entity. Ebben a repository-ban főként lekérdező metódusok vannak, egy létezést vizsgáló metóduson kívül.



19. Projekt repository

### **MunkaRepository**

Ez a legnagyobb terjedelmű repository. Ebben megtalálhatóak már általam létrehozott lekérdező metódusok, mint például az „elszamTablaDolgozoOsszesito” vagy a „findSumRaforditasByProjektId”. Az utóbbi metódus nevében megfigyelhető néhány, már előbb említett kulcsszó, viszont itt különösebb szerepük nincs. Ez azért van, mert létrehozhatóak saját lekérdezés alapján is metódusok. Ezt a @Query annotációval oldottam meg.



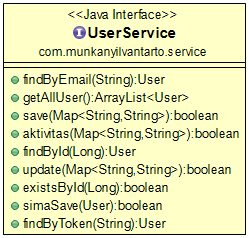
20. MunkaRepository osztálydiagramja

## **Service-ek és ezek implementációi**

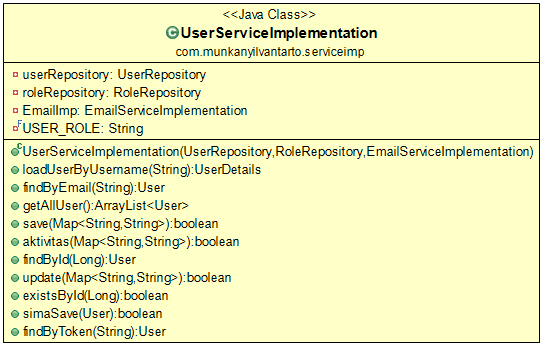
Ebben az alfejezetben be fogom mutatni az előző alfejezetben bemutatott repository-khoz kapcsolódó service-eket. Alapból egy service-et interface-ként hoztam létre. Az interface-ben létre lehet hozni olyan metódusokat, amikhez majd a hozzá tartozó implementációban/implementációkban hozzárendelünk egy repository függvényt. Repository-kat az adott implementációban kétféleképpen hívhatunk meg. Az első megoldás egy konstruktor segítségével valósítható meg, a másik megoldás pedig egy setter segítségvel. Mindkettőben a közös az, hogy el kell látni egy @Autowired annotációval. Erre azért van szükség, hogy a container bejegyezze a kapcsolatot, így kímélve a garbage collector-t. Én a második megoldás alapján oldottam meg a csatlakoztatást minden esetben. Ezen megoldás előnye a könnyebb tesztelhetőség. Általában egy service interface-ben, adott metódust ugyanazon néven hoztam létre, mint a hozzá tartozó repository-ban. Egy adott interface-ben, bármilyen módosítás esetén a hozzá tartozó implementáció/implementációkban is módosítani kell.

### **UserService és UserServiceImplementation**

Maga a UserService egy interface, amihez tartozik a UserServiceImplementation osztály. Ez az osztály el van látva egy @Service annotációval, aminek segítségével a container service osztályként jegyzi be. Az első diagramban megtekinthető a UserService interface felépítése a másodikban pedig ennek az implementációja.



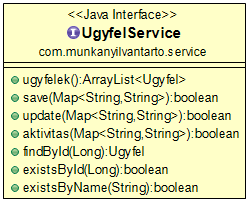
21. UserService interface osztálydiagramja



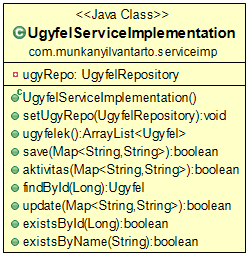
22. UserServiceImplementation osztálydiagramja

Látható, hogy az implementációban megtalálhatóak ugyanazok a metódusok, csak itt már van hozzájuk rendelve művelet. Továbbá az attribútumok között ott vannak azok a repository-k, amiket használ az osztály.

### **UgyfelService és UgyfelServiceImplementation**

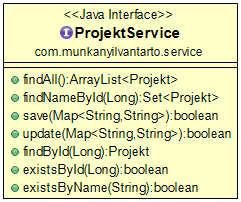


23. UgyfelService interface osztálydiagramja

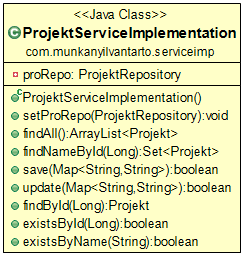


24. UgyfelServiceImplementation osztálydiagramja

### **ProjektService és ProjektServiceImplementation**

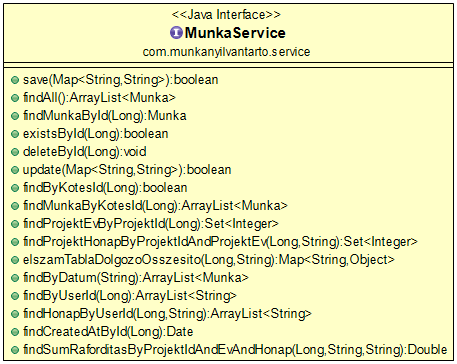


25. ProjektService interface osztálydiagramja

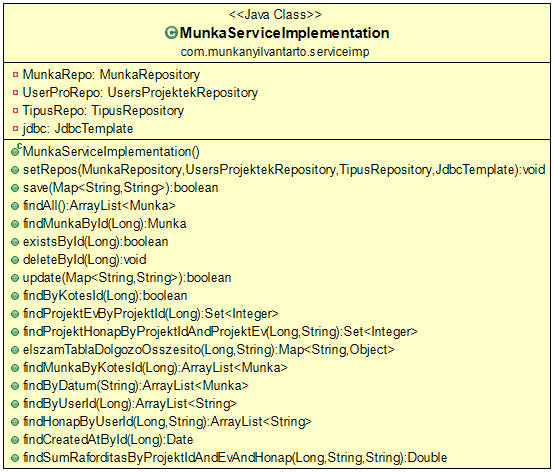


26. ProjektServiceImplementation osztálydiagramja

### **MunkaService és MunkaServiceImplementation**



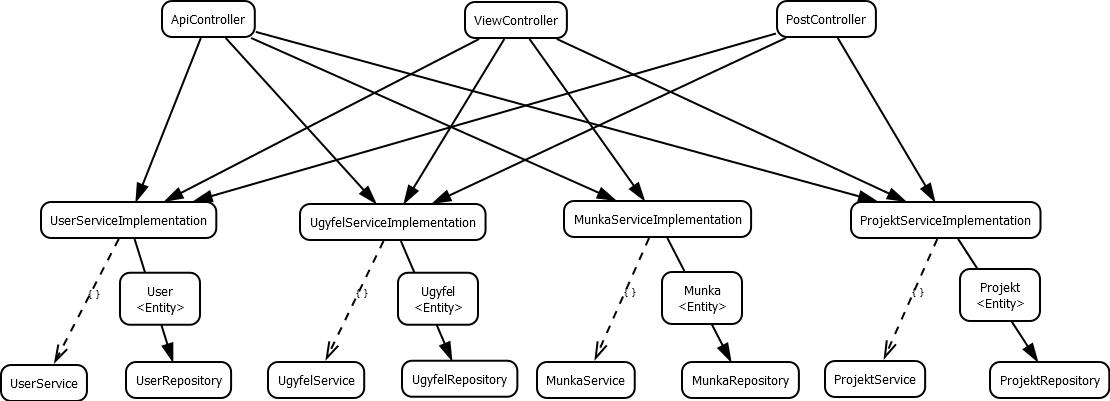
27. MunkaService interface osztálydiagramja



28. MunkaServiceImplementation osztálydiagramja

## **Kapcsolatok**

A következő oldalon megtalálható diagramon láthatóak a rétegek közötti kapcsolatok az előzőekben kifejtett néhány osztály alapján.



29. Rétegek közötti kapcsolat-diagram

# **A rendszer funkciói részletesen**

A következő fejezetben, be fogok mutatni néhány funkciót szerepkörökre lebontva. A kiválasztást gyakoriság szerint végeztem el.

## **Bejelentkezés**



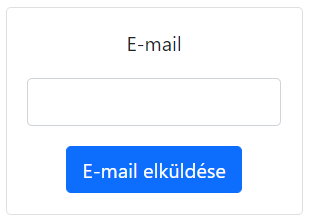
31. Bejelentkezés űrlapja

Minden felhasználónak a felületre való belépéshez szükségük van bejelentkezési adatokra. Ezek az adatok egy e-mail és egy jelszó párosból állnak. A munkáltató belépési adatai a rendszer telepítésekor kerülnek rögzítésre. Utána a munkáltató képes új dolgozókat felvenni és a belépési adataikat létrehozni és kezelni.

## **Jelszó módosítás**

Jelszó módosítás kétféleképpen történhet. Elfelejtett jelszó, illetve kérelem alapján. Utóbbi akkor érhető el, ha az adott felhasználó be van jelentkezve.

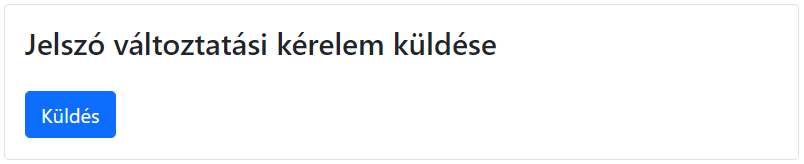
### **Elfelejtett jelszó**



32. Elfelejtett jelszó űrlapja

Az „elfelejtett jelszó” féle jelszó módosítási módszernél még be nem jelentkezett felhasználó is képes jelszót módosítani. Azonban, ezt csak úgy tudja megtenni, ha ismeri az adott felhasználóhoz tartozó e-mail címet, illetve ha hozzá is fér magához az e-mail fiókhoz.

### **Kérelem**



33. Jelszó változtatási kérelem űrlapja

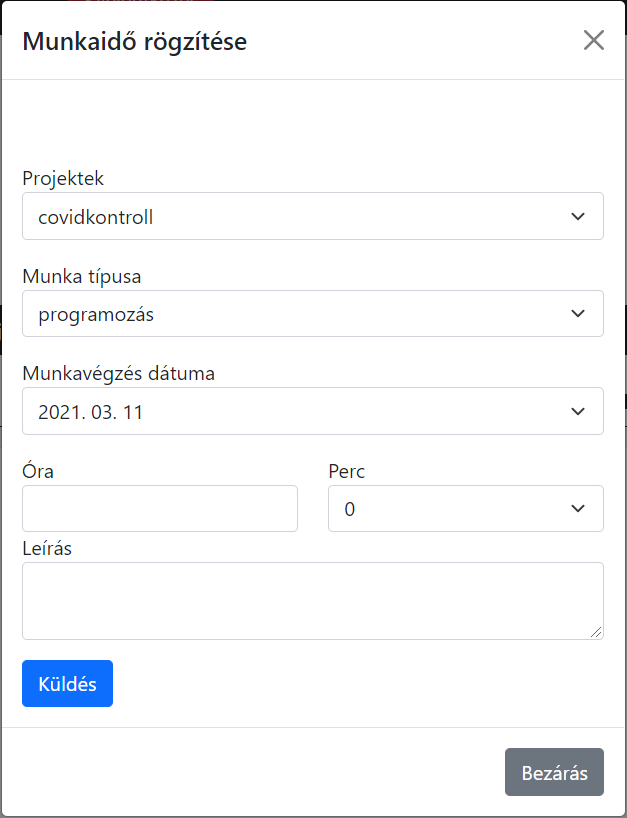
Kérelem alapján történő jelszó módosításhoz már szükség van egy bejelentkezett felhasználóra, ugyanis ennél a módszernél az e-mail cím manuális megadása helyett a felhasználó tárolt adatait felhasználva küldi ki a rendszer a jelszó változtatásról szóló e-mail-t.

## **Dolgozó funkciói**

A dolgozónál a munkákkal kapcsolatos funkciókat fogom bemutatni. A következőkben bemutatott tevékenységek mindegyikéhez, szükség van egy sikeresen azonosított és bejelentkezett dolgozó szerepkörbe tartozó felhasználóra.

### **Munkaidő rögzítése**

Egy új munka rögzítéséhez a munkaidő rögzítése gombra kell klikkelni. Ennek hatására megjelenik a következő felugró ablak, ami tartalmaz egy űrlapot. Ezen űrlap segítségével lehet feltölteni új munkát az adatbázisba.



34. Munkaidő rögzítés űrlapja

**A munkaidő rögzítés tartalmazza:**

* Projektek legördülő menü
  + Ebben a legördülő menüben mindig azok a projektek jelennek meg, amikhez az adott felhasználó kötve van. Ez leegyszerűsíti a kezelést. Mindez Javascript segítségével van megvalósítva.
* Munka típusa legördülő menü
  + Itt jelenik meg minden olyan munka típus, amit létrehozott a munkáltató.
* Munkavégzés dátuma
  + A munkavégzés dátuma legördülő menüben található a mai, illetve a tegnapi nap, mivel egy dolgozó csak egy napra visszamenőleg tud munkát rögzíteni.
* Óra, Perc
  + Az óra mező egy sima szám alapú mező. Ebbe 0-23-ig írhat számokat a dolgozó.
  + A perc egy legördülő menü, amiben az elemek 15 perces osztásokban vannak, tehát 0, 15, 30, 45. Ez a könnyebb számolhatóság, illetve a könnyebb kezelhetőség érdekében van így.
* Leírás
  + Ez egy szöveg beviteli mező, ami a munka magyarázatát kell, hogy tartalmazza. Az ebben bevitt szöveg nem lehet hosszabb, mint 100 karakter.

A bejelentkezett dolgozó azonosítóját egy rejtett mezőben tároltam. Ez azért szükséges, mert az adatbázisba való feltöltés előtt le kell kérdezni azt a kötés azonosítót, aminél a kiválasztott projekt és a dolgozó kötve van. A továbbiakban pedig ez lesz az a kötés azonosító, ami szerint tárolja a munkát az adatbázisban.

Az űrlap kitöltése után a küldés gombra kattintva, AJAX segítségével eljuttatja az adatokat a szerver oldalra, egy előre megadott controller adott metódusához, ami aztán továbbadja a service-nek ami feldolgozza és átadja a repository-nak ami feltölti az adatbázisba.

### **Munkák megjelenítése**

A munkák megjelenítését egy táblázatban oldottam meg. Alapból egy üres táblázat áll rendelkezésre, amiben csak a header elemek szerepelnek. Az oldal betöltésekor indul egy AJAX kérés a szerver megadott web service-e felé. Ez feldolgozza és válaszul küld egy JSON objektumot, ami tartalmazza az összes megjelenítendő munka adatait a kötések adataival együtt.



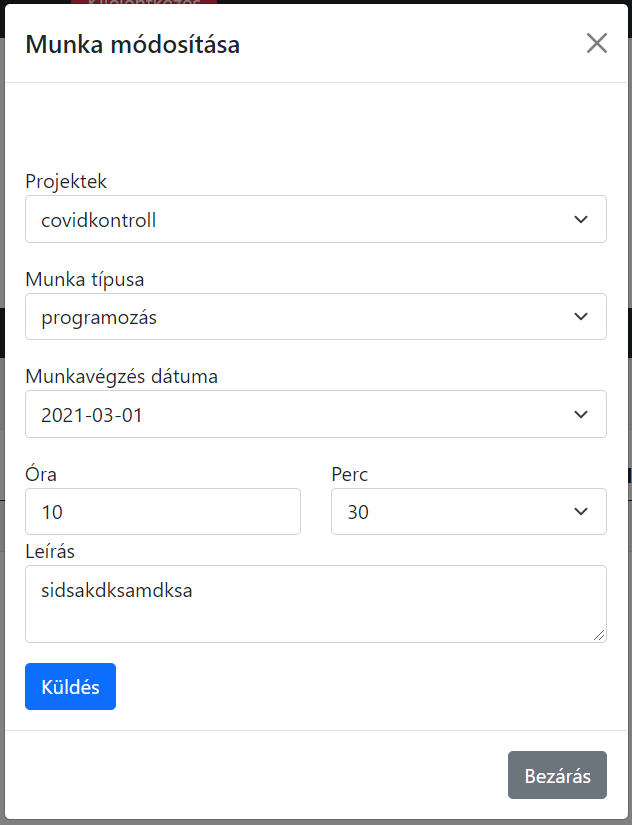
35. Munkák tábla

A megjelenített adatokra szűrést is végezhet a felhasználó év és hónap szerint. A táblázat fölött látható két legördülő menü az évre és a hónapra. A hónap mező függ az évtől. Az év pedig függ a rögzített munkák időpontjától. Ezekben az esetekben is mind a két mezőt javascript segítségével töltöm fel. Az év mezőnél AJAX segítségével elkérem az adott web service-től az összes évet, amihez munka van rögzítve. Fontos, hogy nem az összes rögzített munkánál kell ezt megtenni, csak annál, amit az adott felhasználó rögzített. Ha a felhasználó kiválaszt egy évet, akkor szintén indul egy AJAX kérés, aminél már átadom az év-et is, hogy a hozzá tartozó hónapokat lekérdezzem. Ezek után ha a megjelenítés gombra kattint a felhasználó a tábla sorai törlődnek, majd feltöltődnek a kiválasztott év és hónap szerint.

Látható az ábrán, hogy a táblázat alatt van egy összesítő táblázat is. Ebben javascripttel összesítem, mindig a megjelenített munkákat a **munkanapok**, **ráfordítás**, **átlag óradíj** és **teljes összeg** szerint.

### **Munka módosítása**

A fentebb részletezett táblázat utolsó oszlopában található egy **módosítás** gomb. Ha a felhasználó erre a gombra kattint, akkor a következő űrlap jelenik meg.

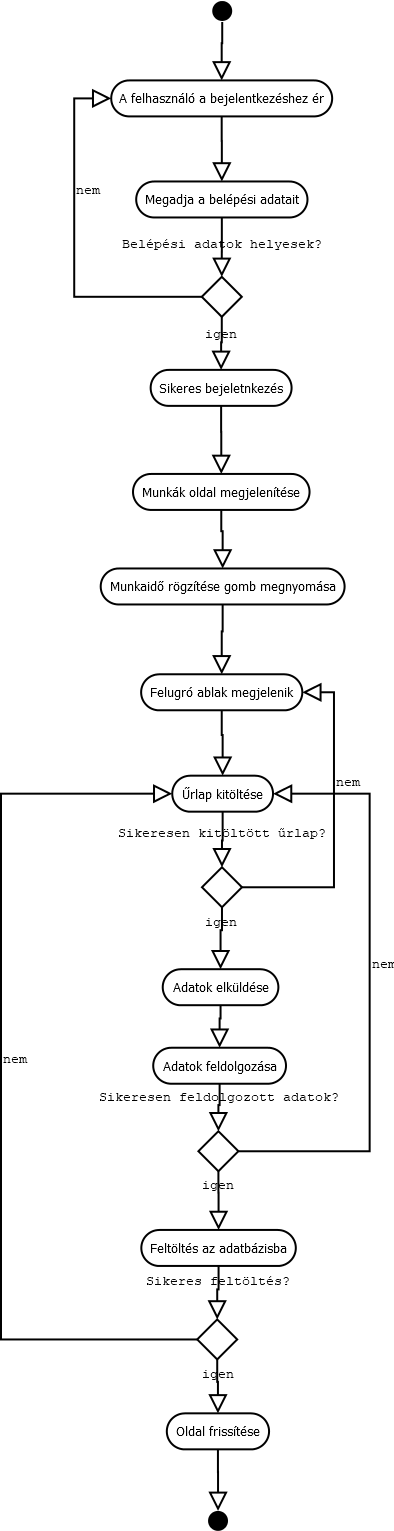


36. Munka módosítása űrlapja

Megfigyelhető, hogy ez az űrlap már előre fel van töltve az adott sor adataival. Mindezt javascript segítségével oldottam meg. Ez az űrlap ugyanazokat a mezőket tartalmazza, mint a feltöltés. Annyi különbséggel, hogy rejtett mezőben tárolom a munka azonosítóját is.

### **Tevékenység-diagram**

Ebben a diagramban be fogom mutatni a munkaidő rögzítés folyamatát a bejelentkezéstől kezdve.

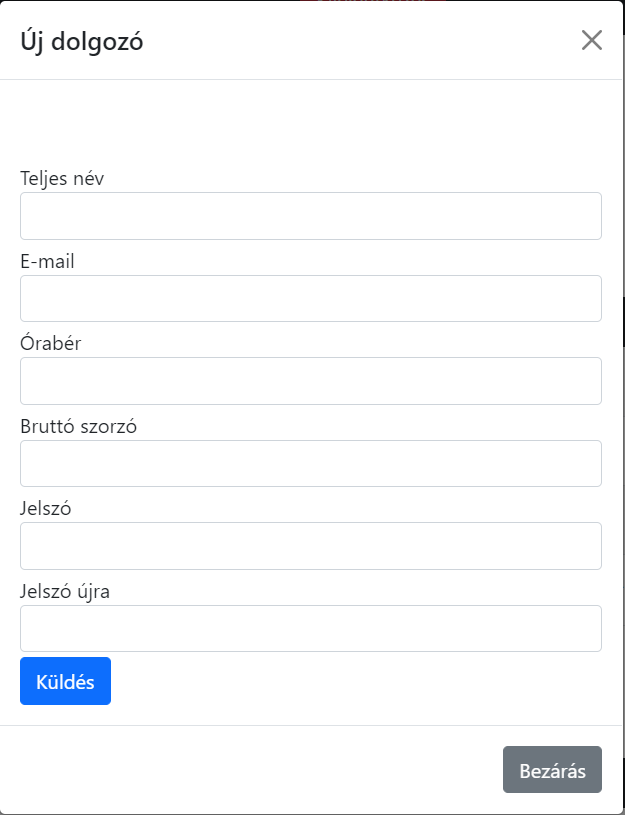


37. Munkaidő rögzítés tevékenység-diagramja

## **Munkáltató funkciói**

### **Dolgozó feltöltése**

Egy új dolgozó rögzítéséhez első körben a dolgozó oldalra kell elnavigálni, utána pedig az „Új dolgozó feltöltés” gombra klikkelve megnyílik az alábbi felugró ablak. Ebben található egy űrlap, amelynek kitöltésével és elküldésével tölthető fel új dolgozó az adatbázisba.

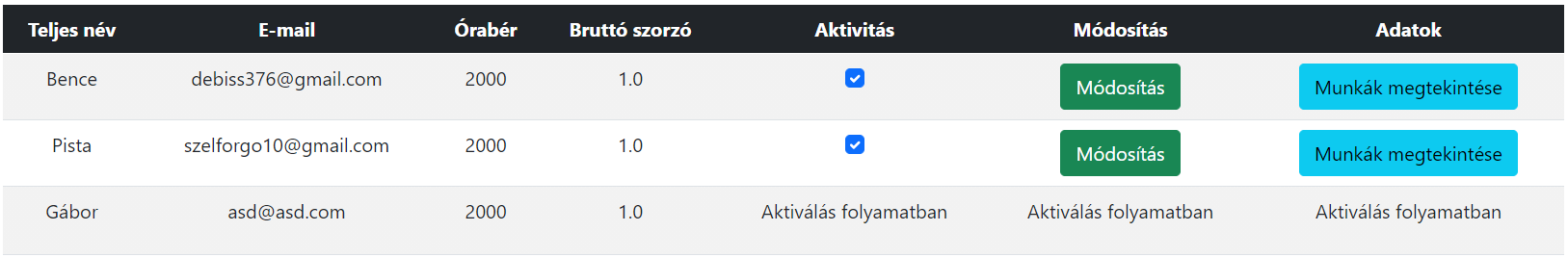


38. Új dolgozó feltöltés űrlapja

**A dolgozó feltöltés tartalmazza:**

* Teljes név
  + Ebben a szöveg típusú beviteli mezőben adható meg a dolgozó teljes nevét.
* E-mail cím
  + Ez az e-mail típusú mező szolgál az e-mail cím megadására.
* Órabér
  + Ez a szám típusú mező szolgál az órabér megadására
* Bruttó szorzó
  + Ebben a mezőben tizedes tört formájában lehet megadni bruttó szorzókat 0 és 1 között (Ebbe beletartozik az 1 is).
* Jelszó és jelszó újra
  + Ez a két mező összefüggésben van egymással, ugyanis nem azonos értékek esetén hibát jelez az elküldésnél. A biztonság érdekében használok két mezőt a jelszó megadására.

### **Dolgozók megjelenítése**



39. Dolgozók megjelenítésére szolgáló tábla

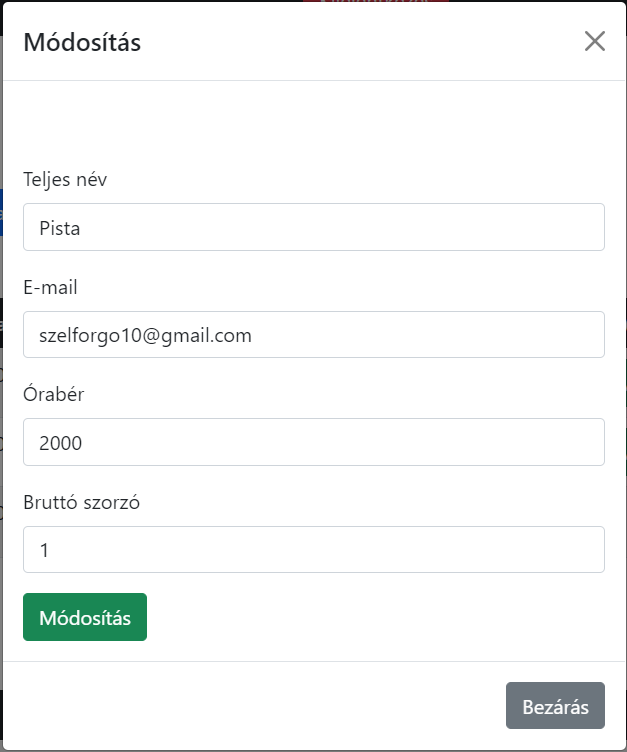
Az táblázatban látható, hogy adott sor egy adott dolgozó adatait tartalmazza. Az adatok biztonsága végett nem minden adatot jelenítettem meg a dolgozókról. Ilyen adat például a jelszó. Ezt a munkáltató oldaláról módosítani, megjeleníteni nem lehet!

A táblázatban 7 sor van megjelenítve. Ezekből az oszlopokból leíró adatok csak az első 4 sorban találhatóak a többi mind funkciókat tartalmaznak, amiket a továbbiakban részletezni fogok.

Továbbá, egy új dolgozó felvétele után a munkáltató még nem tud semmit csinálni az adott dolgozó adataival, mivel egy új dolgozó feltöltésekor a dolgozónak elküldődik egy e-mail, ami tartalmaz egy aktivációs linket. Amíg erre a linkre nem klikkel rá, addig a felhasználó nem aktív és ameddig nem aktív, addig nem lehet manipulálni az adatait.

### **Dolgozó módosítása**

Az előbb bemutatott táblázat 6. sorában található a módosítás gomb. Erre a gombra klikkelve megnyílik a következő űrlap már előre kitöltve az adott sor adatai alapján.



40. Dolgozó módosítás űrlap

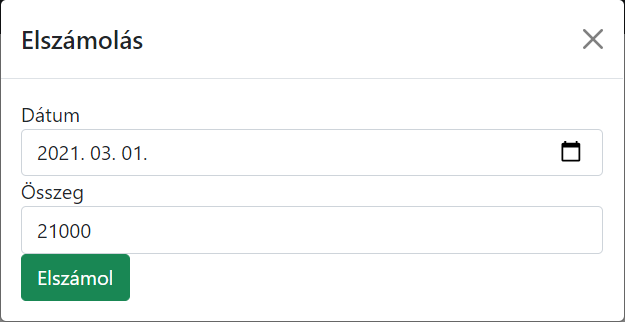
### **Dolgozó munkáinak megtekintése és elszámolása**

A táblázat utolsó oszlopában látható egy „Munkák megtekintése” gomb. Erre a gombra klikkelve az adott dolgozóhoz tartozó munkák oldal jelenik meg, ahol lehetőség van keresésre, szűrésre, munka rögzítésre és elszámolásra is.



41. Pista dolgozóhoz tartozó munkák listája

Az ábrán Pista dolgozóhoz tartozó adatok láthatóak. A „Munkaidő rögzítés” gombra klikkelve a fentebb már bemutatott űrlap jelenik meg azzal a különbséggel, hogy itt bármilyen időpontra lehet munkát rögzíteni. A szűrés, illetve az összesítés ugyanúgy működik, mint a dolgozó oldali munkák felületen. Továbbá, ennél a módosítás gomb helyett egy checkbox található, aminek bepipálása esetén a megjelenő űrlapban meg lehet adni az elszámolni kívánt összeget, illetve az elszámolás dátumát. Ez az űrlap a következő ábrán látható.

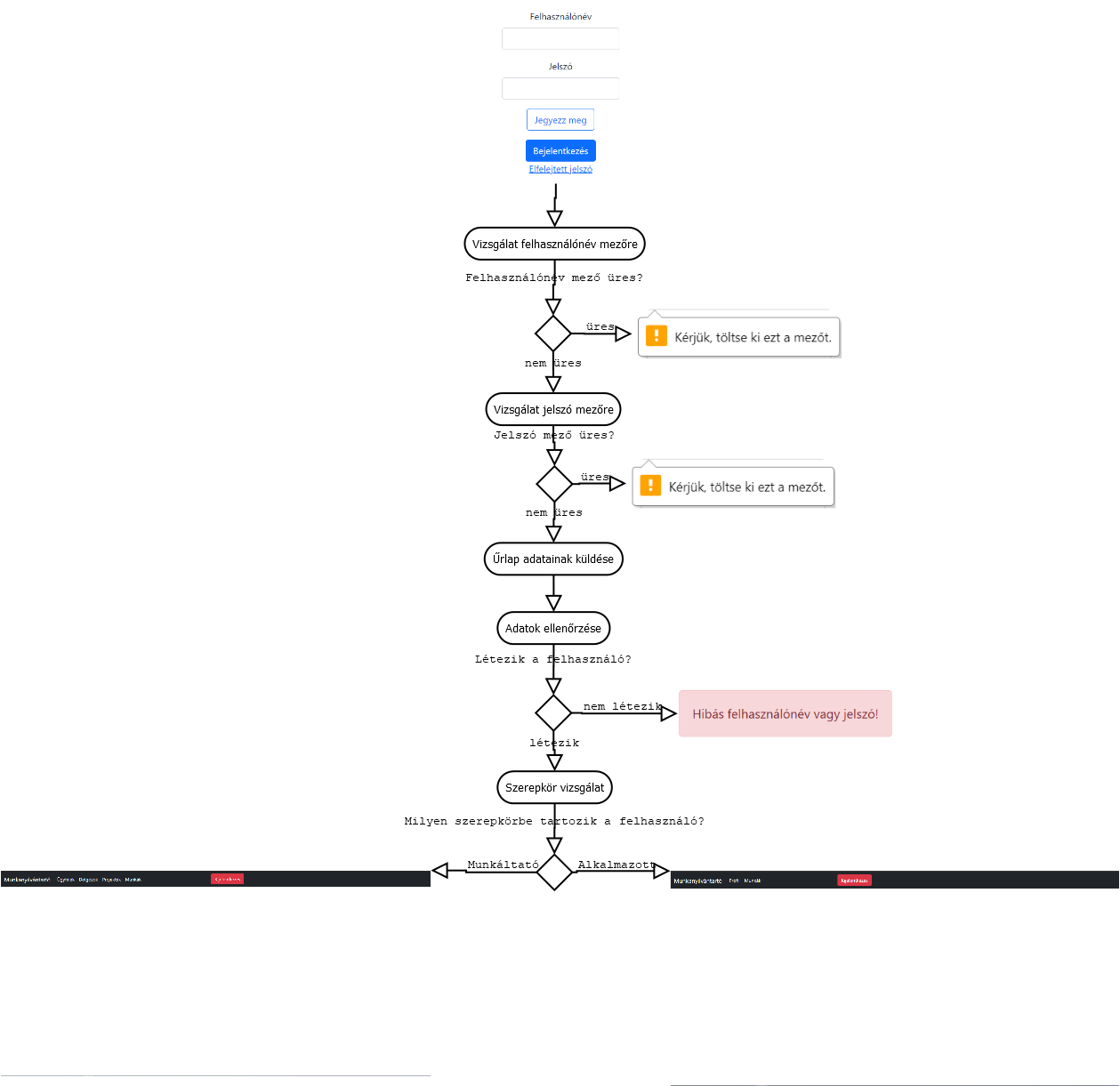


42. Elszámolás űrlap

Az űrlap megjelenésekor a mezők már automatikusan ki vannak töltve az adott sor adataival (összeg és dátum). A dátum mező a munka rögzítésének dátumával egyenlő, de ez természetesen megváltoztatható.

# **Tesztelés**

Minden komponensnél hiányosságtesztelést végeztem, minden útvonalat bejárva. Egy példában be szeretném mutatni a bejelentkezés útvonalait és teszteseteit. Mindezeket a következő ábrák mutatják.



43. Bejelentkezés lehetséges útvonalai

|  |  |  |
| --- | --- | --- |
| Tesztesetek | | |
| leírás | input | output |
| Hiányzó felhasználónév és jelszó | felhasználónév: üres  jelszó: üres |  |
| Hiányzó felhasználónév és helyes jelszó | felhasználónév: üres  jelszó: kitöltve |  |
| Helyes felhasználónév és hiányzó jelszó | felhasználónév: kitöltve,  jelszó: üres |  |
| Helytelen felhasználónév és jelszó | felhasználónév és jelszó kitöltve, de helytelenek |  |
| Helyes felhasználónév és helytelen jelszó | felhasználónév és jelszó kitöltve, felhasználónév helyes, de a jelszó nem |  |
| Helytelen felhasználónév és helyes jelszó | felhasználónév és jelszó kitöltve, felhasználónév helytelen, de a jelszó helyes |  |
| Minden adat helyes és a felhasználóhoz tartozó szerepkör munkáltató | felhasználónév, jelszó kitöltve és helyesek. |  |
| Minden adat helyes és a felhasználóhoz tartozó szerepkör alkalmazott | felhasználónév, jelszó kitöltve és helyesek. |  |

16. Bejelentkezés tesztesetei

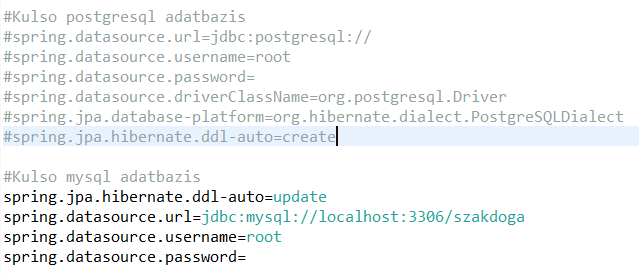
# **Telepítés szerverre**

Mindenek előtt telepíteni kell a Java 15-öt, illetve a legújabb verziójú maven-t.

Ebben a fejezetben be fogom mutatni az alkalmazás beállítását szerver környezetben. Szükség lesz egy adatbázis és egy levelező szerverre. Az **application.properties** fájl tartalmaz minden környezeti változót, amit a program használ, ezért ezeknek a szolgáltatásoknak itt kell majd megadni az elérési útjait.

## **Adatbázis**

Rengeteg féle adatbázissal képes dolgozni a rendszer. Én mysql, illetve postgresql adatbázison teszteltem (ezeknek a környezeti változói mind benne vannak ebben az application.properties fájlban).



44. Adatbázis adatainak megadása

**Környezeti változók:**

* spring.datasource.url
  + Ebben kell megadni az adatbázis szerver címét
* spring.datasource.username
  + Ez tartalmazza a felhasználónevet
* spring.datasource.password
  + Ez tartalmazza a jelszót
* spring.jpa.hibernate.ddl-auto
  + Ez tartalmazza a feltöltés típusát
    - create: Létrehozza az adatbázist az Entity-k alapján
    - create-drop: Létrehozza és eldobja az adatbázist
    - update: Változás esetén módosítja az adatbázist

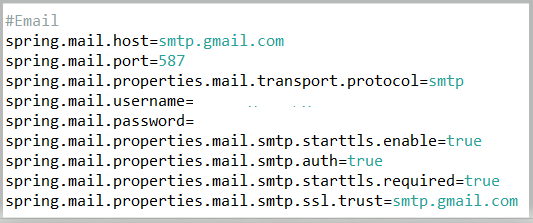
A program első indításákor create-re kell állítani, majd update-re.

* spring.datasource.driverClassName
  + Ezzel adható meg az adatbázis kezelő driver

Továbbá, van egy data.sql nevű fájl, ami tartalmaz néhány sql lekérdezést. Erre azért van szükség, mert első indításkor az adatbázis üresen generálódik le és ennek a .sql fájlnak a segítségével lehet feltölteni az alap adatokat, mint például a szerepkörök, admin felhasználó és ezek kötései. Első indítás előtt az ebben lévő e-mail címet mindenképpen meg kell változtatni egy valós e-mail címre. Illetve a **spring.datasource.initialization-mode** környezeti változó értékét always-re kell állítani, hogy le tudja futtatni a data.sql tartalmát. Miután megtörtént az első indítás és sikeresen legenerálódott az adatbázis az alap adatokkal, akkor ugyanennek a változónak meg kell változtatni az értékét always-ről never-re. Így nem fog minden indulásnál újra lefutni.

## **Levelezés**

Levelező szervernek alap esetben a gmail van beállítva, viszont bármilyen levelező szerver hozzáköthető. Szintén az **application.properties** fájlhoz kell nyúlni a szerver adatainak beállításához.



45. Levelezés adatainak megadása

**Környezeti változók:**

* spring.mail.host
  + Ez tartalmazza a levelező szerver elérését
* spring.mail.port
  + Ez tartalmazza a szerver portot
* spring.mail.properties.mail.transport.protocol
  + Küldési protokoll-t tartalmaz
* spring.mail.username
  + Felhasználónevet tartalmaz
* spring.mail.password
  + Jelszót tartalmaz

Be kell állítani továbbá a szerver portot és a logger szintet is.

## **Folyamat**

Miután be lett állítva minden az **application.properties**-ben jöhet az utolsó pár lépés.

Az adatbázis beállításánál a „**spring.jpa.hibernate.ddl-auto**” változó tartalmát át kell állítani „**create**”-re, azért hogy legyártsa induláskor a teljes adatbázist.

Amint ezek megvannak el kell indítani a programot. Indulásnál néha lehet probléma az, hogy rosszul állítja be a **UserDetailsService**-t. Ilyenkor a megoldás az, hogy annyiszor újra kell indítani a programot, ahányszor ezt a hibát dobja. Nagyjából a 2-3. indításra működni szokott. Miután teljesen elindult a program le kell állítani és az **application.properties**-ben a „**spring.jpa.hibernate.ddl-auto**” változó tartalmát meg kell változtatni „**update**”-re. Utána megint el kell indítani a programot és le kell ellenőrizni, hogy az adatbázis rendben van. Ha igen, akkor már csak át kell állítani a logger szintjét „**info**”-ra.

Ezek után már buildelhető is a program. Miután sikeres volt a build az elkészült jar fájlt meg kell nyitni cmd-ben és onnan indítani. Ez azért fontos, mert így kapunk egy konzolt, amiből kezelhető.

# **Felhasználói útmutató**

## **A program rövid ismertetése**

Ez a program egy munka kezelő alkalmazás, amely két részből áll. Van egy dolgozói és van egy munkáltatói felület. Ezeket külön-külön később fogom részletezni. Fontos, hogy mind a két felület jelszóval védett, ezért szükség van minden felhasználónak egy egyedi azonosítóra, amivel be tud lépni. Közös funkciójuk a kijelentkezés.

## **Műszaki követelmények**

A rendszernek kliens oldalról nincs különösebb műszaki követelménye, illetve telepítésre sincs szükség, ugyanis az alkalmazás web alapú. Használatához csak egy böngészőre van szükség.

## **Használati útmutató Alkalmazottaknak**

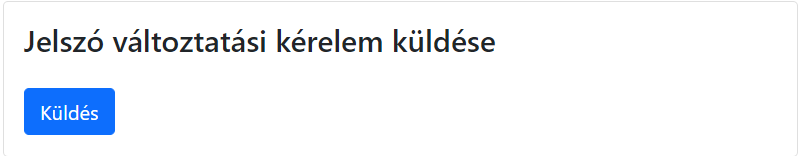
Alkalmazott oldalról a jelszó módosítás a munkaidő rögzítés és a munka módosítása funkciót fogom bemutatni. Ez a része az alkalmazásnak képes új munkák rögzítésére, a rögzített munkák megjelenítésére, módosítására, összesítésére és szűrésére. Továbbá, lehetőség van jelszó módosításra is.

**Menütérkép:**

* Profil
  + Jelszó módosítás
* Munkák
  + Munkaidő rögzítése
  + Munkák módosítása
  + Munkák szűrése év és hónap szerint
  + Munkák összesítése
  + Munkák megjelenítése

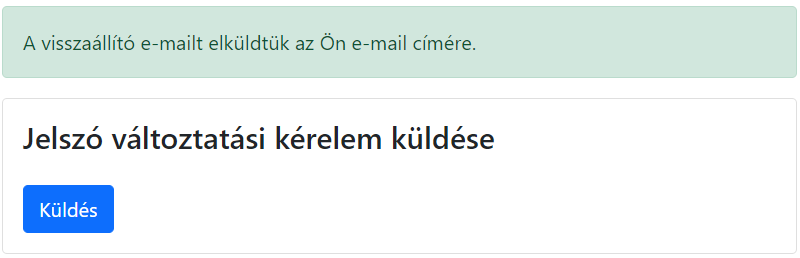
## **Jelszó módosítása**

A profil módosítása nem közvetlen módon történik, hanem egy kérelem elküldésével. Továbbá, ennek elküldéséhez szükség van egy létező e-mail címre (ez a bejelentkezéskor használt e-mail cím), szóval érdemes a kérelem elküldése előtt ellenőrizni, hogy hozzá fér-e az e-mail fiókjához. Ha igen, akkor semmi akadálya a kérelem elküldésének.



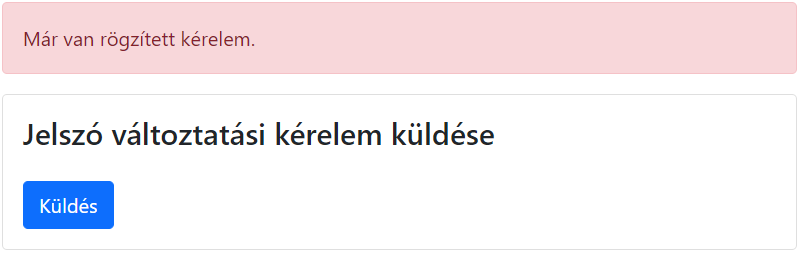
46. Jelszó változtatási kérelem küldése

Klikkeljen rá a „Küldés” gombra, majd tölteni fog egy kicsit az oldal és ha sikeres volt az e-mail küldés, akkor kapni fog egy visszajelzést a sikeres küldésről.



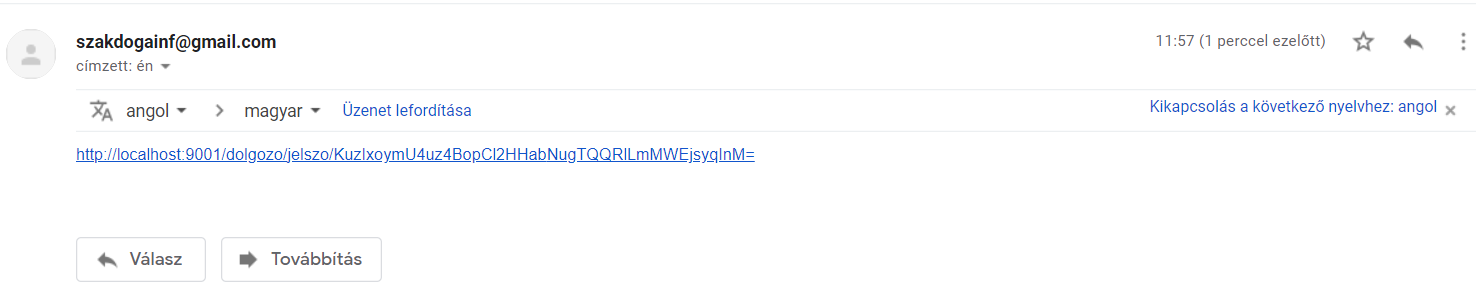
47. A küldés után sikeres üzenet

Egyszerre csak egy kérelmet lehet küldeni. Több kérelem elküldése esetén a következő hibaüzenetet fogja kapni.



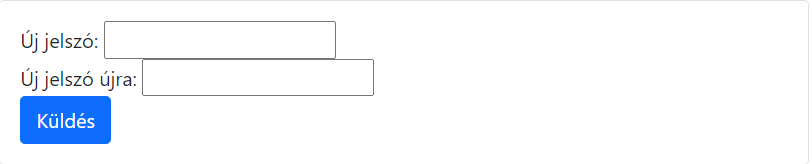
48. Kérelem elküldése úgy, hogy már van rögzített kérelem visszajelzés

Ezek után be kell jelentkeznie az e-mail fiókjába és meg kell tekintenie, hogy megkapta-e az e-mailt, ami tartalmaz egy linket.



49. Példa a beérkező e-mail-re

Erre a linkre ráklikkelve elnavigálja egy jelszó változtató oldalra, ahol meg kell adnia az új jelszavát kétszer.



50. Jelszó változtató oldal

Miután ez megvan el kell küldeni az űrlapot és ha sikeres volt a módosítás, akkor vissza navigálja a profil oldalra egy sikeres üzenettel.

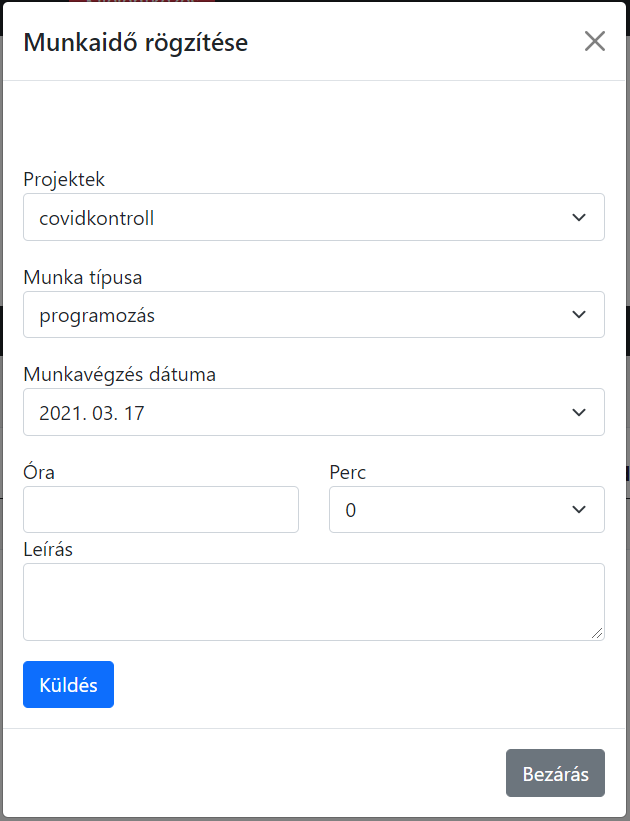
## **Munkaidő rögzítése**

A munkaidő rögzítéséhez elsőként el kell navigálnia a „Munkák” oldalra. Itt található a táblázat fölött egy „Munkaidő rögzítése” gomb.



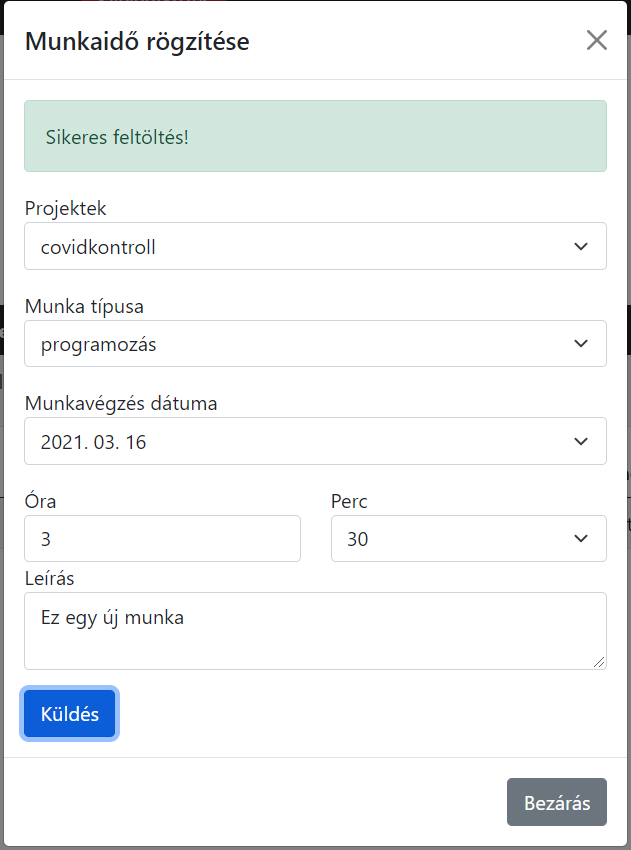
51. Munkaidő rögzítése gomb a táblázat felett

Erre rá klikkelve megnyílik egy felugró ablak. Ez tartalmaz egy űrlapot.



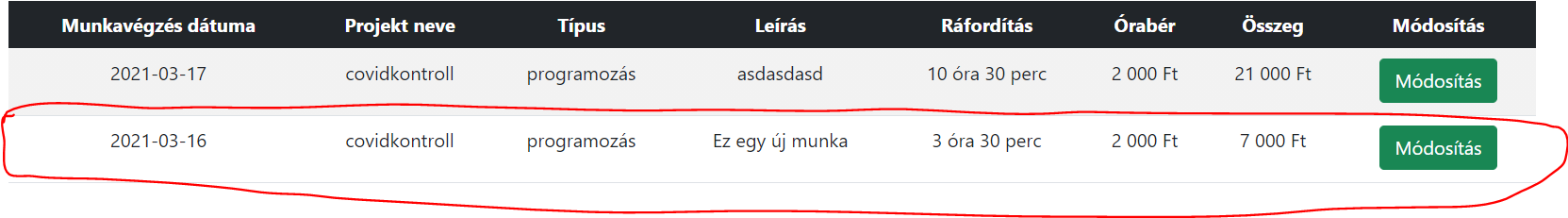
52. Munkaidő rögzítése űrlap

Az első mező egy legördülő lista, ami tartalmazza az összes olyan projektet. amihez kötve van. Természetesen, ha nincs az Ön felhasználójához kötve projekt, akkor ez a mező üres. A második legördülő lista tartalmazza a munka típusait. Mielőtt munkát szeretne rögzíteni, el kell döntenie, hogy milyen típusú munkát végzett, hogy kiválaszthassa a megfelelő típust. Miután mindent kitöltött klikkeljen rá a „Küldés” gombra. Pillanatokon belül meg fog jelenni egy visszajelzés.



53. Sikeres rögzítés esetén megjelenő visszajelzés

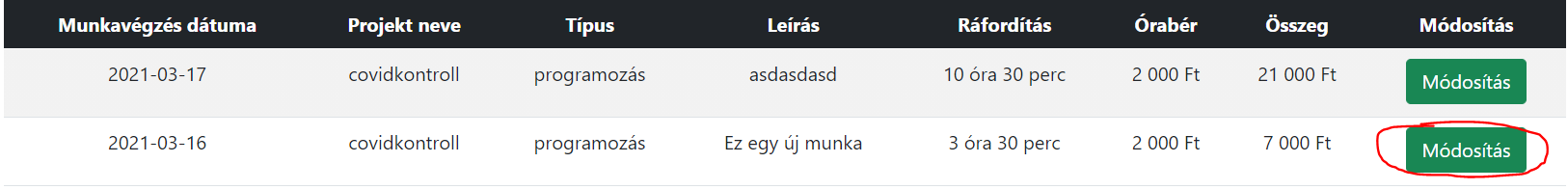
Sikeres feltöltés esetén frissülni fog az oldal és az imént rögzített munka meg fog jelenni a táblázatban.



54. Frissítés után a táblázatban megjelent sor

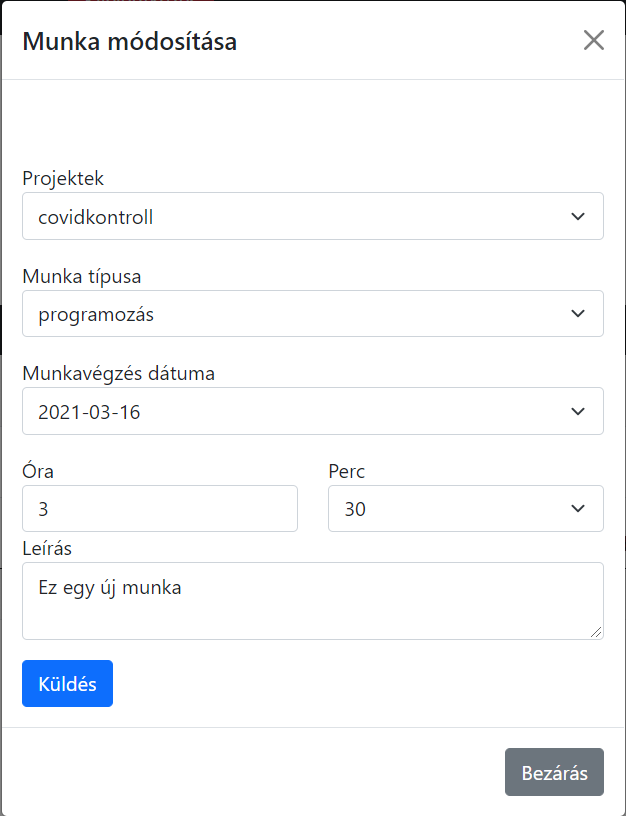
## **Munka módosítása**

Munka módosításához elsőnek el kell döntenie, hogy melyik munkát szeretné módosítani. Miután ez megvan, a táblázat utolsó oszlopában található „Módosítás” gombra kell klikkelnie a módosítandó sorban.



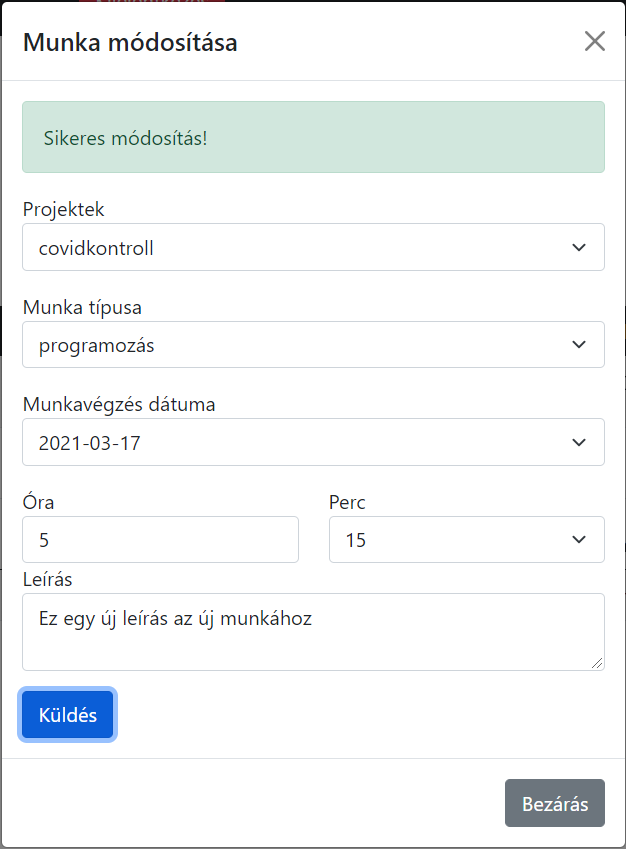
55. Munka módosítása gomb

Ennek hatására meg fog jelenni egy ablak, ami tartalmazza ugyanazt az űrlapot, mint a rögzítés, azzal a különbséggel, hogy ennél az adatok automatikusan feltöltődnek a módosítani kívánt sor (munka) adataival.



56. Munka módosítása űrlap

Ezen az űrlapon megváltoztatható bármelyik adat. Miután végzett a módosításokkal klikkeljen a „Küldés” gombra. Ez elküldi az űrlap adatait és ezek alapján módosítja az adott munkát. Ha sikeres volt a módosítás, akkor egy sikeres visszajelzést fog kapni.



57. Munka sikeres módosítása esetén visszajelzés

Ezután frissülni fog az oldal és a táblázat adott sorában már látni is fogja a módosított adatokat az adott sorban.

## **Használati útmutató Munkáltatóknak**

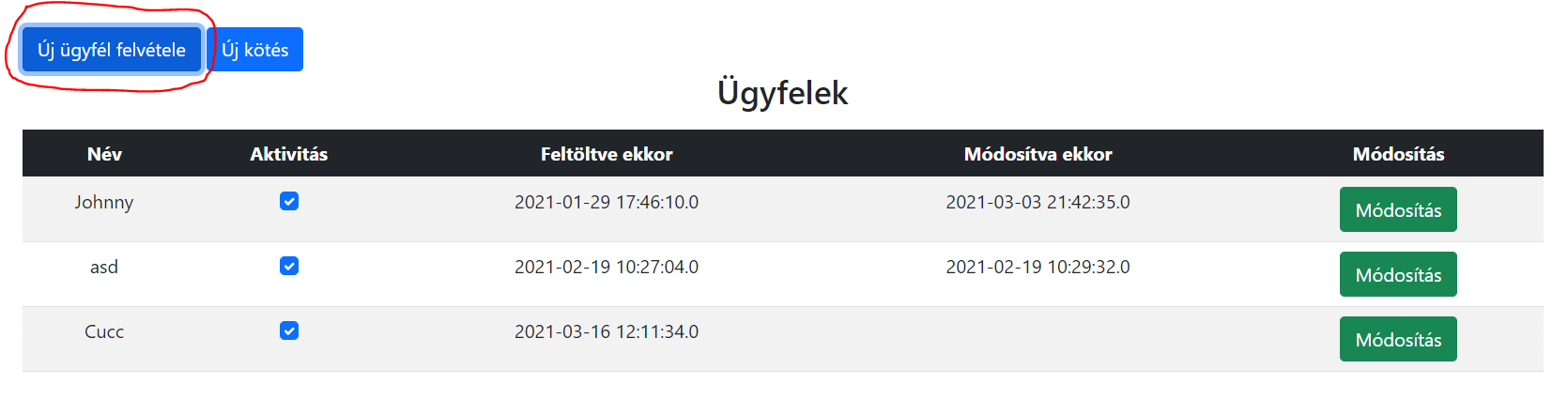
Munkáltató oldalról csak néhány fontosabb funkciót fogok bemutatni, mivel rengeteg funkciót tartalmaz a felület. Ezek a következőek lesznek: Ügyfél feltöltése, Ügyfél-projekt kötés rögzítése, Ügyfél elszámolása.

**Menütérkép:**

* Ügyfelek
  + Új ügyfél felvétele
  + Új kötés
  + Ügyfelek megjelenítése
  + Ügyfelek módosítása
  + Ügyfelek aktivitásának állítása
  + Ügyfelek-projektek kötések megjelenítése és törlése
  + Elszámolás ügyfélre, projekt és dátum alapján
* Dolgozók
  + Új dolgozó feltöltése
  + Új dolgozó kötése
  + Munkaidő rögzítése
  + Munka típus feltöltés
  + Dolgozó aktivitásának állítása
  + Dolgozó adatainak módosítása
  + Dolgozó adatainak megjelenítése
  + Dolgozó-projektek kötések megjelenítése és törlése
  + Munka típusok megjelenítése
  + Munka típusok módosítása
* Projektek
  + Új projekt létrehozása
  + Új költség feltöltése
  + Projektek megjelenítése
  + Projektek módosítása
  + Projektek költségeinek megjelenítése
  + Projektek módosítása
  + Projektek törlése
* Munkák
  + Munkák megjelenítése
  + Munkák módosítása
  + Munkák törlése

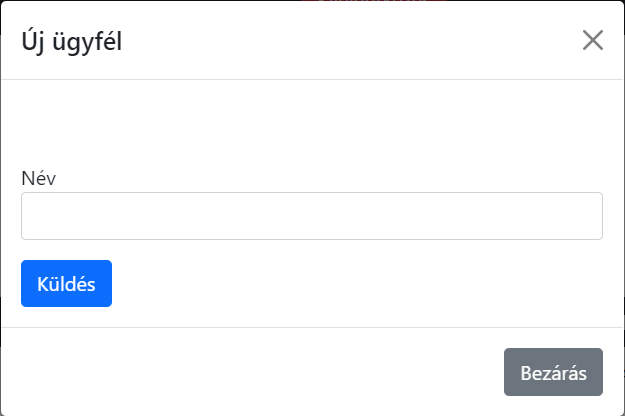
## **Ügyfél feltöltése**

Új ügyfél feltöltéséhez el kell navigálnia az „Ügyfelek” oldalra. Az „Ügyfelek” táblázat fölött található egy „Új ügyfél felvétele” feliratú gomb.



58. Ügyfél feltöltése gomb

Erre ráklikkelve megnyílik egy felugró ablak, ami tartalmaz egy beviteli mezőt, amiben az ügyfél teljes nevét lehet megadni.

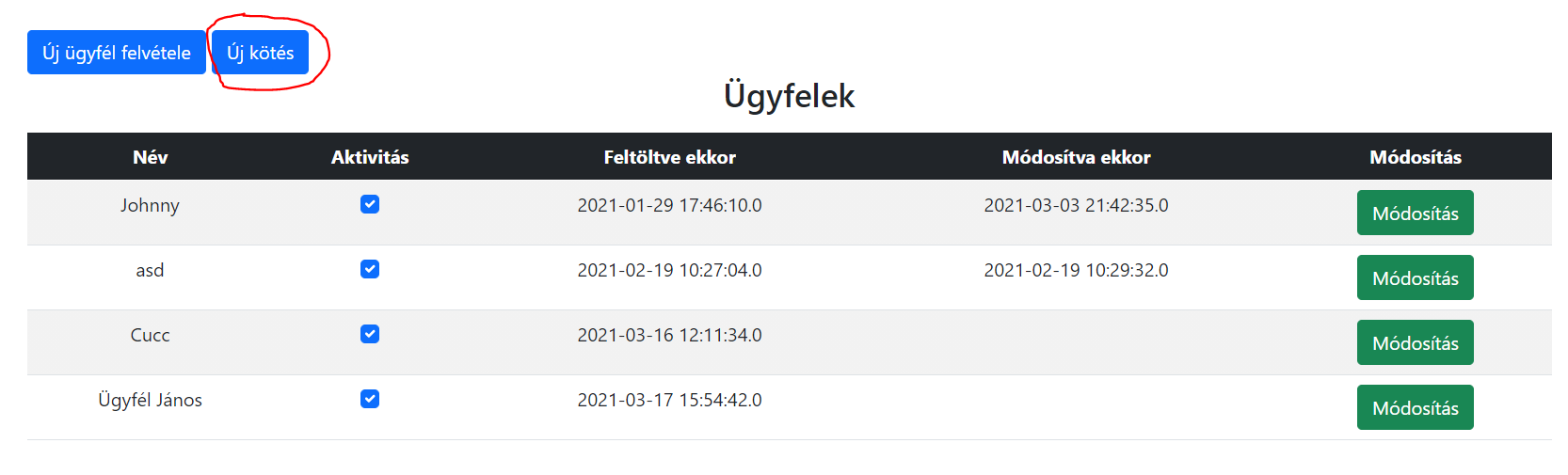


59. Új ügyfél feltöltése űrlap

A „Küldés” gombra klikkelve elküldi az űrlapot és sikeres feltöltés esetén frissül az oldal, majd az új ügyfél megtekinthető lesz az „Ügyfelek” táblában.

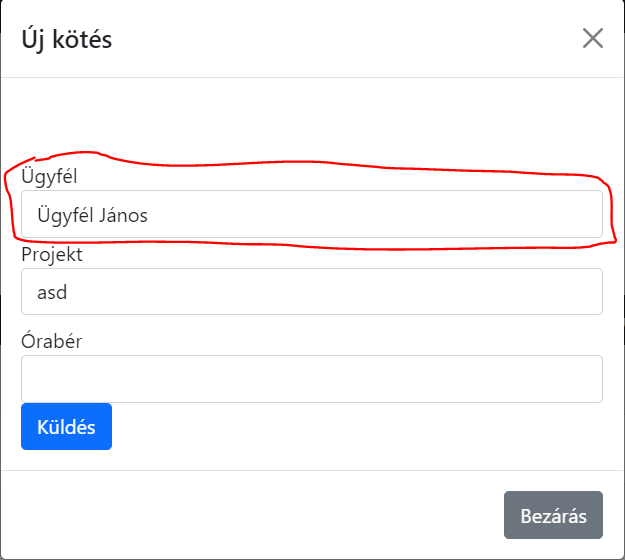
## **Ügyfél-projekt kötés rögzítése**

Új kötés rögzítéséhez az „Ügyfelek” tábla fölötti „Új kötés” feliratú gombra kell klikkelni.



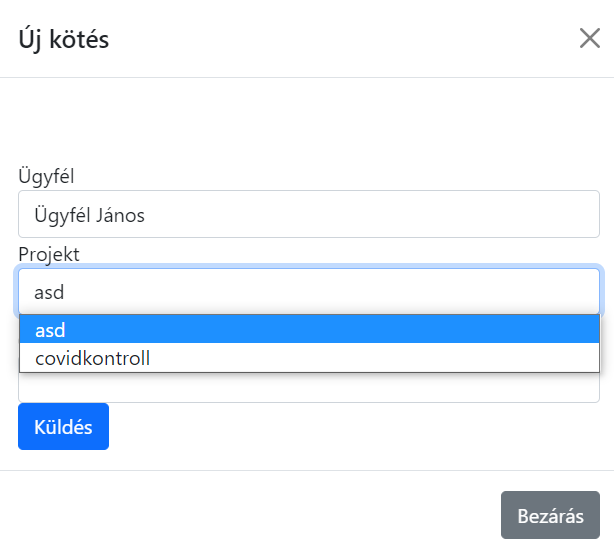
60. Új kötés gomb

Ezáltal meg fog nyílni egy felugró ablak. Első körben ki kell választani egy ügyfelet az első listából.



61. Új kötés űrlapon ügyfél lista

Kiválasztás után a „Projekt” listába be fognak töltődni azok a projektek, amihez a kiválasztott ügyfél még nincs kötve.



62. Ügyfél kiválasztás hatására projekt lista feltöltése

Ha ez megvan, akkor utolsó lépésként egy órabért kell megadni a kötéshez. Erre azért van szükség, mert az elszámolásnál ezzel az órabérrel fog számolni a rendszer. Ha kitöltötte az űrlapot, akkor a „Küldés” gombra klikkelve elküldi az űrlapot. Sikeres feltöltés esetén frissül az oldal és létrejön egy új sor az „Ügyfelek-projektek kötései” táblában.

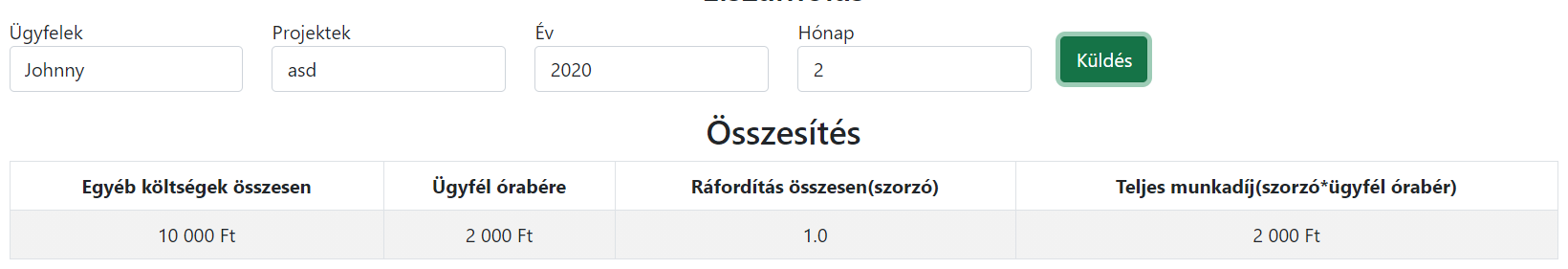
## **Ügyfél elszámolása**

Az „Ügyfelek” oldal legalján található az elszámolás rész. Itt minden lista hatással van a következőre. Tehát ki kell választania egy ügyfelet az első listából. Kiválasztás után a hozzá tartozó projektekkel feltölti a „Projektek” listát. Ebből a feltöltött listából ki kell választani egy projektet, melynek hatására feltöltődik az „Év” lista. Az „Év” listának az elemei azok az évek, amikben volt munka rögzítve az adott projekthez. Az „Év” kiválasztása után feltöltődik a „Hónap” lista, ami tartalmazza a kiválasztott évhez tartozó összes olyan hónapot, amiben volt munka rögzítve.



63. Elszámolás listák és űrlap kinézete

Miután kitöltötte az űrlapot, a „Küldés” gombra klikkelve az adatok feldolgozásra kerülnek és megjelenik az űrlap alatt egy összesítő táblázat.



64. Elszámolás kitöltés és feldolgozás után az összesítő táblázattal

A táblázat tartalmazza az összes egyéb költséget, ami a projekthez tartozik az adott hónapban. Továbbá tartalmazza az ügyfél órabérét, a teljes ráfordítást és a teljes munkaidőt.

# **Továbbfejlesztés lehetőségei**

Úgy gondolom, hogy a program egy erős, használható béta állapotban van. Gyorsan és egyenletesen működik, viszonylag magas terheltség ellenére is. Vannak funkciók, amik a könnyebb kezelhetőség és átláthatóság érdekében továbbfejlesztésre szorulnak.

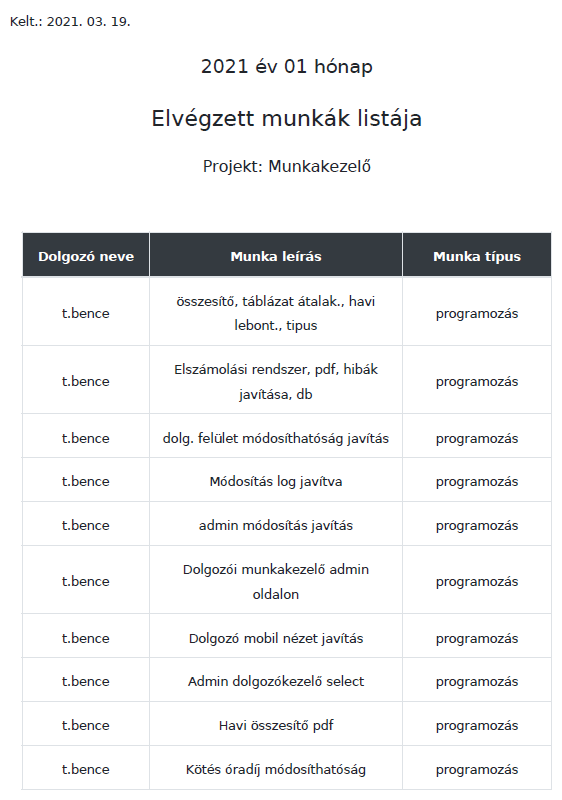
Ezek a továbbfejlesztési lehetőségek a következők:

* Alkalmazott oldalon a manuális munkaidő rögzítés mellett létre szeretnék hozni egy különálló stopper-t, amivel az alkalmazottak képesek pontosan rögzíteni a munkáikat.
  + Elsőnek ki kéne választani egy projektet
  + 30 másodperc inaktivitás után a stopper automatikusan megállna, amit aztán az alkalmazott újraindíthat. Ilyenkor az idő nem állna vissza alaphelyzetbe, hanem onnan folytatódna, ahonnan megállt.
  + 5 percenként képernyőképeket készítene az alkalmazott képernyőjéről, amiket aztán tárolna és a munkáltató le tudná ellenőrizni munkák alapján.
  + A munka végeztével a stopper-t le kéne állítani. Ennek hatására a munka adatai tárolásra kerülnének az adatbázisban.



65. Lehetséges terv a stopper kinézetére

* Munkáltató oldalon külön elszámolás rész.
  + Projektekre külön elszámolás évre és hónapra
    - Dolgozók szerint
    - Egyéb összegek szerint
    - Ügyfelek szerint
  + Elszámolásokból pdf generálás ügyfélnek és alkalmazottnak
    - Ezeknél táblázatokról van szó, tehát minden pdf generálás előtt ki lehetne választani a szükséges oszlopokat. Így többféleképpen lehetne felhasználni ezeket a pdf-eket.



66. Konkrét példa elszámolás pdf-re

* Munka típusok kötése projektekhez
  + Így csak azokat a típusok érnék el a felhasználók, amik hozzá vannak kötve ahhoz a projekthez, amihez az alkalmazott is kötve van.

Mindezek mellett lehetne csinálni mobil vagy asztali applikációt a programhoz. Mivel rengeteg funkció REST alapú és web servicekben van megoldva, ezért egy ilyen alkalmazás képes lenne használni ugyanazt az adatbázist, mint a web app.

# **Mellékelt fájlok**

Mivel a fájlok nagyrésze automatikusan generált, amikor létre lett hozva a projekt, ezért csak az „src/main” mappa tartalmát fogom leírni.

**java mappa tartalma:**

* com mappa
  + munkanyilvantarto mappa
    - config mappa
      * MethodSecurityConfig.java
      * WebConfig.java
    - controller mappa
      * ApiController
      * PostController
      * ViewController
    - entity mappa
      * Elszamolas.java
      * Kerelem.java
      * Koltseg.java
      * Munka.java
      * Projekt.java
      * Role.java
      * Tipus.java
      * TokenGenerator.java
      * Ugyfel.java
      * UgyfelekProjektek.java
      * User.java
      * UserJelszo.java
      * UsersProjektek.java
    - repository mappa
      * ElszamolasRepository.java
      * KerelemRepository.java
      * KoltsegRepository.java
      * MunkaRepository.java
      * ProjektRepository.java
      * RoleRepository.java
      * TipusRepository.java
      * UgyfelekProjektekRepository.java
      * UgyfelRepository.java
      * UserRepository.java
      * UsersProjektekRepository.java
    - service mappa
      * ElszamolasService.java
      * EmailService.java
      * KerelemService.java
      * KoltsegService.java
      * MunkaService.java
      * ProjektService.java
      * TipusService.java
      * UgyfelekProjektekService.java
      * UgyfelService.java
      * UserService.java
      * UsersProjektekService.java
    - serviceimp mappa
      * ElszamolasServiceImplementation.java
      * EmailServiceImplementation.java
      * KerelemServiceImplementation.java
      * KoltsegServiceImplementation.java
      * MunkaServiceImplementation.java
      * ProjektServiceImplementation.java
      * TipusServiceImplementation.java
      * UgyfelekProjektekServiceImplementation.java
      * UserDetailsImplementation.java
      * UgyfelServiceImplementation.java
      * UserServiceImplementation.java
      * UsersProjektekServiceImplementation.java
    - MunkaNyilvantarto.java

**resources mappa tartalma:**

* static mappa
  + css mappa
    - error.css
  + js mappa
    - adminDolgozoMunka.js
    - dolgozo.js
    - dolgozoMunka.js
    - error.js
    - functions.js
    - jelszo.js
    - munka.js
    - profil.js
    - projekt.js
    - ugyfel.js
* templates
  + auth mappa
    - login.html
  + layouts mappa
    - kinti.html
    - main.html
  + djelszo.html
  + dolgozo.html
  + elfelejtett.html
  + elszamolas.html
  + error.html
  + index.html
  + jelszo.html
  + munka.html
  + profil.html
  + projekt.html
  + ugyfel.html
  + visszaigazolas.html
* application.properties
* messages.properties

# **Irodalomjegyzék**

Baeldung weboldal. Forrás: <https://www.baeldung.com/> (2021. 02. 25)

Bootstrap 5 dokumentáció. Forrás: <https://getbootstrap.com/docs/5.0/getting-started/introduction/> (2021. 03. 15)

Spring boot videósorozatok. Forrás: <https://www.youtube.com/watch?v=pS6ZOUzgzY4&list=PLyriihBWoulywcSbZijjeSIHH19uJZG5q> , <https://www.youtube.com/watch?v=c7acCveXdNo&list=PLyriihBWoulxdsrKphv-dN022imGHDizM> (2020. 11. 25)

Java 15 dokumentáció. Forrás: <https://docs.oracle.com/en/java/javase/15/> (2021. 01. 18)

Stack Overflow platform. Forrás: <https://stackoverflow.com/> (2021. 02. 25)

W3Schools weboldal. Forrás: <https://www.w3schools.com/> (2021. 01. 30)

Spring boot dokumentáció. Forrás: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/> (2021. 03. 10)