

Munka menedzser

Programozás alapjai 3

Házi feladat

Programozói dokumentáció

Welker Gergő

ECKCA4

1. Ötlet

Házi feladatként egy munka menedzselő és nyomonkövető alkalmazást készítettem. Az alkalmazásba regisztrálhatnak felhasználók, akik ezután közzé tehetnek munkákat és hozzárendelhetnek más felhasználókat adott munkákhoz továbbá különböző beosztásokat adhatnak a hozzárendelt felhasználóknak. A munkákon dolgozó felhasználók jelentéseket tudnak írni, amiken keresztül nyomon követhető, hogy az adott munka hogyan halad.

Munkáknál megadhatóak a mérföldkövek, amik mentén teljesíteni kell az adott munkát. Ezeket a részfeladatokat egyesével teljesítik az felhasználók. Mérföldkő elkészülése esetén egy arra jogosult felhasználó ellenőrzi és elfogadja vagy elutasítja a teljesítést. Az összes mérföldkő teljesítését követően készsé lehet nyilvánítani a munkát.

Munkákon dolgozó felhasználóknak külön szerepei és beosztásai vannak. Adott szerephez és beosztáshoz különböző jogok tartoznak. Ilyen jogok pl.: felhasználók elvétele és hozzáadása munkákhoz, mérföldkő teljesítetté nyilvánítása stb...

2. Use-case-k

Cím	Regisztráció
Leírás	A felhasználók regisztrálni tudnak az alkalmazásba. A regisztráció igen egyszerű, teljes nevet, egyedi felhasználónevet és jelszót kell megadni megerősítéssel. Ezt követően be is lehet jelentkezni
Aktorok	Felhasználó
Főforgatókönyv	<ol style="list-style-type: none">1. A felhasználó regisztráció felületre navigál2. Megadja a szükséges adatokat, teljes név, egyedi felhasználónév, jelszó megerősítéssel együtt3. A regisztráció sikeres, a felhasználó beléphet
Alternatív forgatókönyv	<ol style="list-style-type: none">1.A.1. A felhasználó regisztráció felületre navigál és megadja az adatokat1.A.2. A megadott felhasználónév nem egyedi, másikat kell választani
Alternatív forgatókönyv	<ol style="list-style-type: none">1.B.1. A felhasználó regisztráció felületre navigál és megadja az adatokat1.B.2. A jelszó megerősítés nem egyezik, újra be kell írja a jelszavakat

Cím	Bejelentkezés
Leírás	A már regisztrált felhasználók bejelentkeznek az alkalmazásba. Ehhez az egyedi felhasználónevük és jelszavuk szükséges
Aktorok	Felhasználó
Főforgatókönyv	<ol style="list-style-type: none">1. A felhasználó megadja felhasználónevét és a hozzá tartozó jelszót2. Az adatok egyeznek, a bejelentkezés sikeres
Alternatív forgatókönyv	<ol style="list-style-type: none">1.A.1. A felhasználó megadja felhasználónevét és a hozzá tartozó jelszót1.A.2. Az adatok nem egyeznek, újra kell próbálnia a felhasználónak

Cím	Kijelentkezés
Leírás	A bejelentkezett felhasználó kijelentkezik a profiljából
Aktorok	Felhasználó
Főforgatókönyv	<ol style="list-style-type: none">1. A felhasználó be van jelentkezve egy fiókba2. Kijelentkezés gombra kattint és kijelentkezik

Cím	Név- és jelszóváltoztatás
Leírás	Felhasználói adatok megváltoztatása, nevet és jelszót lehet változtatni a felhasználói beállításokban, azonban az egyedi felhasználónév nem megváltoztatható
Aktorok	Felhasználó
Főforgatókönyv	<ol style="list-style-type: none"> 1. A felhasználó nevet akar változtatni 2. Megadja új nevét 3. A neve megváltoztatásra kerül
Alternatív forgatókönyv	<ol style="list-style-type: none"> 1.A.1. A felhasználó jelszót akar változtatni 1.A.2. Megadja a jelenlegi jelszavát és az új jelszavát kétszer megerősítés miatt 1.A.3. A szükséges adatok egyeznek, a jelszó megváltozik
Alternatív forgatókönyv	<ol style="list-style-type: none"> 1.B.1. A felhasználó jelszót akar változtatni 1.B.2. A szükséges adatok megadását követően valami nem egyezik 1.B.3. Újra kell próbálkoznia

Cím	Projekt létrehozása
Leírás	Egy felhasználó új projektet hoz létre
Aktorok	Felhasználó
Főforgatókönyv	<ol style="list-style-type: none"> 1. Felhasználó új projekt létrehozása gombra megy 2. Megadja a szükséges adatokat, eleinte csak a projekt neve szükséges 3. Létrejön az új projekt, "folyamatban" státusszal

Cím	Projekt státusz beállítása
Leírás	Egy arra jogosult felhasználó beállítja a projekt státuszát
Aktorok	Felhasználó
Főforgatókönyv	<ol style="list-style-type: none"> 1. A projektben lévő mérföldkövek mind elkészülnek 2. Egy arra jogosult felhasználó elkészültté nyilvánítja a projektet
Alternatív forgatókönyv	<ol style="list-style-type: none"> 1.A.1 A projekt vezetői valamiért úgy döntenek, hogy a projektet nem fejezik be 1.A.2 Egy arra jogosult felhasználó meghiúsulttá nyilvánítja a projektet

Cím	Projekt törlése
Leírás	Egy arra jogosult felhasználó törli a projektet
Aktorok	Felhasználó
Főforgatókönyv	<ol style="list-style-type: none"> 1. Egy projektet valamilyen okból kifolyólag törlésre ítélnék 2. Egy arra jogosult felhasználó törli a projektet

Cím	Mérföldkő hozzáadása projekthez
Leírás	Egy arra jogosult felhasználó mérföldkövet ad hozzá egy projekthez
Aktorok	Felhasználó
Főforgatókönyv	<ol style="list-style-type: none"> 1. A felhasználó "Új mérföldkő" gombra kattint a projekten belül

	<ol style="list-style-type: none"> 2. Megadja a mérföldkő nevét és leírását 3. A mérföldkő hozzáadódik a projekthez "Folyamatban" státusszal
--	--

Cím	Mérföldkő státusz beállítása
Leírás	Egy arra jogosult felhasználó beállítja egy projekten belül egy mérföldkő státuszát
Aktorok	Felhasználó
Főforgatókönyv	<ol style="list-style-type: none"> 1. Egy felhasználó megvizsgálja egy mérföldkő állapotát 2. Ha úgy véli, hogy a mérföldkő elkészült és van rá jogosultsága, akkor elkészültté nyilvánítja

Cím	Mérföldkő törlése
Leírás	Egy arra jogosult felhasználó töröl egy mérföldkővet egy projektből
Aktorok	Felhasználó
Főforgatókönyv	<ol style="list-style-type: none"> 1. Egy projekten belül egy mérföldkő nem szükséges a továbbiakban 2. Egy arra jogosult felhasználó törli a mérföldkővet

Cím	Megjegyzés hozzáadása mérföldkőhöz
Leírás	Egy adott projekten dolgozó felhasználó megjegyzést fűz egy, a projektben lévő mérföldkőhöz a mérföldkő állapotával kapcsolatban
Aktorok	Felhasználó
Főforgatókönyv	<ol style="list-style-type: none"> 1. Egy felhasználó elkészül egy mérföldkő részével vagy egészével 2. Megjegyzésben megírja, hogy mely résszel vagy részekkel készült el
Alternatív forgatókönyv	<p>1.A.1. Az írt megjegyzésében egy felhasználó utólag észreveszi, hogy hibát vétett</p> <p>1.A.2. A megjegyzésen belül a "Szerkesztés" gombra kattintást követően átírja a megjegyzést</p>

Cím	Megjegyzés törlése
Leírás	Egy felhasználó töröl egy általa előzőleg írt megjegyzést
Aktorok	Felhasználó
Főforgatókönyv	<ol style="list-style-type: none"> 1. A megjegyzés valamilyen okból kifolyólag irrelevánssá válik 2. A felhasználó törli a megjegyzést a megjegyzésen belül lévő "Törlés" gombbal

Cím	Felhasználó projekthez adása
Leírás	Egy arra jogosult felhasználó új felhasználót ad a projekthez
Aktorok	Felhasználó
Főforgatókönyv	<ol style="list-style-type: none"> 1. Új munkatárs érkezik a projekthez 2. Egy arra jogosult felhasználó megkapja az új felhasználó egyedi felhasználónevét 3. A felhasználónév segítségével hozzárendelésre kerül az új felhasználó a projekthez

Cím	Felhasználó eltávolítása a projektől
Leírás	Egy arra jogosult felhasználó eltávolít egy korábban projekthez adott felhasználót
Aktorok	Felhasználó
Főforgatókönyv	<ol style="list-style-type: none"> 1. Valamilyen okból kifolyólag el kell távolítani egy felhasználót egy adott projektől 2. Egy arra jogosult felhasználó kikeresi a projekten dolgozó felhasználók közül az eltávolítani kívánt felhasználót 3. Az "Eltávolítás" gomb segítségével az adott felhasználó eltávolításra kerül a projektől

Cím	Felhasználó szerepeinek beállítása
Leírás	Egy arra jogosult felhasználó beállítja egy másik felhasználó szerepeit egy adott projekten belül
Aktorok	Felhasználó
Főforgatókönyv	<ol style="list-style-type: none"> 1. Egy arra jogosult felhasználó kikeresi a kívánt felhasználót a projekten dolgozó felhasználók közül 2. Jelölőnégyzetek segítségével beállítja, hogy milyen szerepei és szerephez tartozóan jogai lesznek a felhasználónak a projekten belül

3. Feladat és megvalósításának részletezése

3.1. Use-case-k technikai részletezése

3.1.1. Regisztráció

Miután a felhasználó megadta az adatait ellenőrizzük őket. Ellenőrizzük a jelszó hosszát és hogy megfelel-e az elvárásoknak, továbbá azt, hogy nem használja-e valaki más már a megadott felhasználónevet. Ezt követően beírjuk a felhasználó felhasználónevét a „users.json” fájlba és létrehozunk egy új „<user>.json” fájlt amibe a felhasználó minden adata lesz tárolva.

3.1.2. Bejelentkezés

Felhasználónév és jelszó megadása után megnézzük, hogy létezik-e a felhasználónév a „users.json” fájlban, ha igen akkor megnyitjuk a felhasználónévhez tartozó fájlt és a benne lévő jelszót összehasonlítjuk a felhasználó által megadott jelszóval. Ha egyezik akkor a „config.json” fájlba beállítjuk, hogy az adott felhasználó be van jelentkezve majd bejelentkeztetjük a felhasználót. Megjelenik a felhasználói panel, ahol betöltésre kerülnek a projektek és a felhasználói profil.

3.1.3. Kijelentkezés

Megnyitjuk a „config.json” fájlt, beállítjuk, hogy nincs bejelentkezve senki és kiléptetjük a felhasználót a bejelentkezés panelhez.

3.1.4. Név- és jelszóváltoztatás

A szükséges adatok megadását követően ellenőrizzük azokat, ha minden rendben van, akkor megnyitjuk a felhasználóhoz tartozó „<user>.json” fájlt és átírjuk az adatokat benne a kapott új adatokra.

3.1.5. Projekt létrehozása

A szükséges adatok megadását követően ellenőrizzük azokat. Ha minden rendben van, akkor generálunk egy egyedi projekt azonosítót majd ezzel a generált azonosítóval készítünk egy „<project>.json” fájlt amibe a projekt adatai kerülnek tárolásra. Ezen kívül megnyitjuk a felhasználóhoz tartozó „<user>.json” fájlt és a készített projekt azonosítóját hozzáadjuk a felhasználó projektjeihez. A felhasználó, aki készíti a projektet mindig „Készítő” szerepkört foglal el a projektben és ezt nem lehet megváltoztatni. Mindenhez joga van és projektet is csak ő tudja törölni.

3.1.6. Projekt státusz beállítása

Megnyitjuk a projekthez tartozó „<project>.json” fájlt és átállítjuk benne a projekt státuszára vonatkozó értéket.

3.1.7. Projekt törlése

Megkeresünk mindegyik, a projekthez tartozó felhasználót, töröljük az adott projekt azonosítóját a projektjeik közül, majd töröljük a projekthez tartozó "<project>.json" fájlt.

3.1.8. Mérföldkő hozzáadása projekthez

A projekthez tartozó "<project>.json" fájlhoz hozzáadunk egy új mérföldkövet.

3.1.9. Mérföldkő státusz beállítása

Az adott mérföldkő státuszát átállítjuk a projekthez tartozó "<project>.json" fájlban.

3.1.10. Mérföldkő törlése

Töröljük a mérföldkövet a projekthez tartozó "<project>.json" fájlból a hozzá tartozó kommentekkel együtt.

3.1.11. Megjegyzés hozzáadása mérföldkőhöz

Megjegyzést adunk hozzá a megfelelő mérföldkőhöz a projekthez tartozó "<project>.json" fájlban.

3.1.12. Megjegyzés szerkesztése

Átírunk egy megjegyzést a megfelelő mérföldkőnél a projekthez tartozó "<project>.json" fájlban.

3.1.13. Megjegyzés törlése

Törölünk egy megjegyzést a megfelelő mérföldkőtől a projekthez tartozó "<project>.json" fájlból.

3.1.14. Felhasználó projekthez adása

A felhasználónév megkapását követően hozzáadjuk a projekthez tartozó "<project>.json" fájlhoz a kapott felhasználót, majd a kapott felhasználóhoz tartozó "<user>.json" fájlhoz hozzáadjuk a megfelelő projektet.

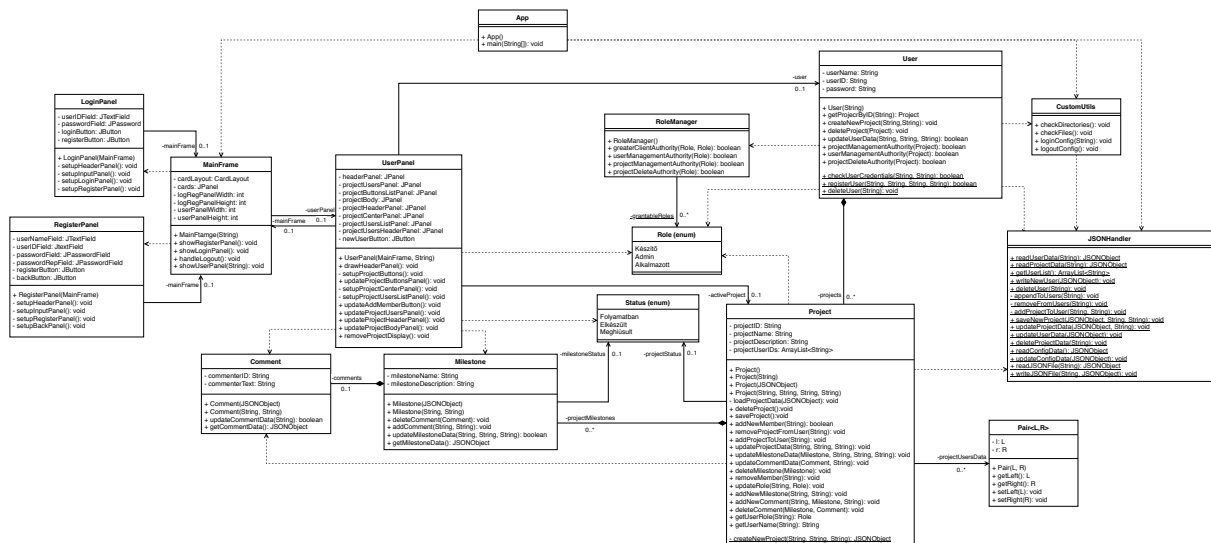
3.1.15. Felhasználó eltávolítása a projekttől

Az adott felhasználót töröljük a projekthez tartozó "<project>.json" fájlból majd a felhasználóhoz tartozó "<user>.json" fájlból töröljük a projektet.

3.1.16. Felhasználó szerepeinek beállítása

A projekthez tartozó "<project>.json" fájlban a megfelelő felhasználóhoz tartozó szerepet beállítjuk a kívánt értékekre.

3.2. Osztálydiagram



Az osztálydiagramon nem kerültek megjelenítésre a getter-setter metódusok és a listener osztályok, mert átláthatatlanná tennék a diagramot. A képen a program fő magját alkotó osztályok látszódnak, a listener osztályok a kezelhetőség és könnyebb átláthatóság külön lettek szedve. A listener osztályok a későbbiekben lesznek részletezve, továbbá a mellékelt javadoc dokumentációban is részletes leírás található róluk.

Az alábbiakban az osztálydiagramon lévő osztályok rövid részletezésére kerül sor, mélyebb áttekintést a projekt forráskódjában lévő részletes leírások vagy az abból generált javadoc dokumentáció nyújt, melyekben minden fontos metódus, osztály, tesztelés részletes kommenteket tartalmaz.

3.2.1. Osztályok rövid részletezése

- MainFrame

Az alkalmazás interfészének alapjait megvalósító osztály, egy CardLayout-on belül kezeli a LoginPanel, RegisterPanel és UserPanel osztályokat.

- LoginPanel

A bejelentkezési panel létrehozását és kezelését végző osztály. Az általa létrehozott panelben lehet bejelentkezni vagy a regisztrációs panelt megnyitni.

- RegisterPanel

A regisztrációs panel létrehozását és kezelését végző osztály. A létrehozott panelben lehet új felhasználó regisztrálását elvégezni, vagy a bejelentkezési panelbe visszalépni.

- UserPanel

A felhasználói panelt hozza létre és kezeli az interfészét. Az osztály által létrehozott panelben lehet minden, konkrét felhasználóval és projekttel kapcsolatos műveletet végrehajtani.

- User

A felhasználóhoz kapcsolódó logikai műveleteket végző osztály. Tárolja a felhasználó adatait és kezeli a felhasználó hatáskörébe tartozó projektekkel kapcsolatos logikai műveleteket. Segítségével lehet felhasználókat létrehozni, felhasználói adatokat betölteni, frissíteni-szerkeszteni és törölni.

- Project

Minden, a projektekkel kapcsolatos logikai művelet kezelését végző osztály. Segítségével lehet projekteket létrehozni, szerkeszteni és törölni. A projekten belül találhatóak a mérföldkövek és kommentek kezelésével kapcsolatos metódusok is, ezek a külvilág számára egyszerűsítik le a projektek kezelését, belül azonban a megfelelő osztályhoz tartozó metódusok kerülnek meghívásra. Erre azért volt szükség, mert a mérföldkövek és kommentek mind a projekthez tartoznak és a projekt menti ki a saját adatait JSON fájlba, amikor pedig egy projekthez tartozóan bármilyen változtatás történik az mentésre is kerül, ha ezek külön lennének kezelve mind akkor túl felaprózódnának a műveletek.

- Milestone

Mérföldkövek kezelésére szolgáló osztály, segítségével lehet mérföldköveket létrehozni, szerkeszteni továbbá a mérföldkö adatait fájlba kiíráshoz lekérni.

- Comment

Mérföldkövekhez tartozó kommentek kezelését végző osztály, segítségével lehet kommenteket létrehozni és szerkeszteni.

- RoleManager

Az alkalmazásban lévő szerepekhez tartozó jogokat kezelő osztály. Segítségével lehet megtudni, hogy egy adott szerephez milyen jogok tartoznak az alkalmazásban és ez alapján vannak bizonyos interfész elemek egyes felhasználóknak megjelenítve, másoknak pedig eltüntetve, vagy csak bizonyos részben megjelenítve. Szerepek hatáskörei:

- A legmagasabb szerep a „Készítő” szerep, egy készítő van mindig egy projektben és bármit megtehet. Szerkesztheti, törölheti a projektet, mérföldköveket készíthet, szerkeszthet, törölhet...stb.
- Azonos szereppel rendelkező felhasználók nem tudják szerkeszteni egymást, csak magasabb szintű tud alacsonyabb szintűt
- „Admin” szereppel rendelkező felhasználó abban különbözik a „Készítő”-től, hogy nem tudja törölni a projektet és nem tud „Készítő” és „Admin” felhasználókat szerkeszteni
- „Alkalmazott” szereppel csak kommenteket lehet mérföldkekhöz fűzni

- JSONHandler

JSON fájlok kezelését végző osztály, minden fájlba írás és fájlból olvasás művelet megvalósítója.

- CustomUtils

Olyan metódusokat tároló osztály, melyek egyik másik osztályhoz sem köthetők hozzá szorosan, azonban nagyon fontos szerepet töltenek be a projekt kezelésében.

3.3. Mappaszerkezet és fájlok

Minden felhasználói adat (felhasználónév, jelszó, név, projektek stb....), projekt adatok, kommentek és beállítások JSON fájlokban vannak tárolva. A program működéséhez szükség van a megfelelő könyvtár struktúrára, továbbá szükség van az alapvető fájlok meglétére. A könyvtárstruktúra kialakításáról és a fájlok meglétéről a program induláskor gondoskodik. A megfelelő struktúráról és a fájlok meglétéről a CustomUtils osztály checkDirectories és checkFiles metódusai gondoskodnak. A struktúra kinézete a következő:

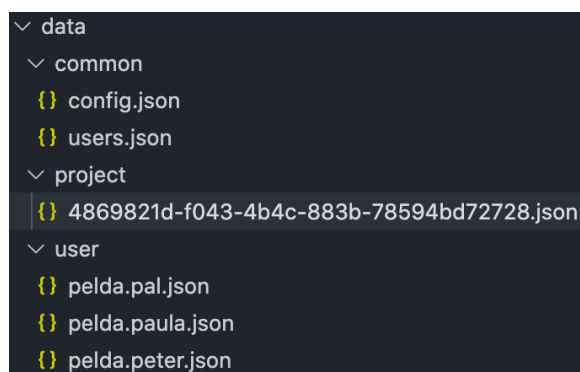
```
data
|-common
    |-users.json
    |-config.json
|-user
|-project
```

Ha bármelyik mappa vagy fájl hiányzik akkor a program azokat létrehozza induláskor.



```
data
├── common
│   ├── config.json
│   └── users.json
├── project
└── user
```

Kép az alap struktúráról és fájlokról.



```
data
├── common
│   ├── config.json
│   └── users.json
├── project
│   └── 4869821d-f043-4b4c-883b-78594bd72728.json
└── user
    ├── pelda.pal.json
    ├── pelda.paula.json
    └── pelda.peter.json
```

Példa egy struktúráról ami három felhasználót és egy projektet tartalmaz

3.4. JSON fájlok

Azért választottam JSON fájlípust a projekthez, mert viszonylag könnyen kezelhetőek és ellenőrizhetőek, emberi megtekintésre is alkalmasak továbbá a kulcs-érték szerinti tárolás kifejezetten jól illett a program által tárolandó adatokhoz.

3.4.1. A programban használt JSON fájlok:

3.4.1.1.

config.json - A program konfigurációját tartalmazó fájl, jelenleg csak azt tartalmazza, hogy melyik felhasználó van éppen az alkalmazásba bejelentkezve, de ez egyéb adatokkal szükség esetén bővíthető. A "data/common/" mappában helyezkedik el és alapvető fontosságú a program működéséhez.

3.4.1.2.

users.json - Az alkalmazásba regisztrált felhasználók felhasználó neveit tárolja, könnyebb keresés érdekében ide gyűlnek egybe a felhasználónevek. A "data/common/" mappában helyezkedik el és alapvető fontosságú a program működéséhez.

3.4.1.3.

<user>.json - A felhasználók neveit, jelszavait, felhasználóneveit és azon projektek azonosítóit tárolja amelyekben az adott felhasználó szerepel. Minden egyes felhasználóhoz tartozó felhasználói adat egy külön JSON fájlban helyezkedik el a "data/user/" mappán belül. Felhasználó regisztrálása esetén kerül elkészítésre, a <user> rész a fájlneven belül a felhasználói azonosítóra utal. Regisztráció esetén ellenőrizve van, hogy ezek mindig egyediek maradnak, mert keresési kulcsként vannak használva.

3.4.1.4.

<project>.json - A projektek nevét, státuszát, leírását, mérföldköveit, a mérföldkövek neveit, megjegyzéseit, státuszát továbbá a projekten dolgozó felhasználók felhasználóneveit és a hozzájuk tartozó szerepeket tárolja. Minden projekt egy külön JSON fájlban helyezkedik el a "data/project/" mappán belül. A <project> rész a fájlneven belül egy generált UUID ami minden project számára egyedi és keresési kulcsként is használva van a programban.

3.5. JSON fájlok szerkezetei

3.5.1. A projekt JSON fájl szerkezete

```
{
  "projectStatus": "Folyamatban",
  "projectDescription": "Példa projekt leírás",
  "projectName": "Példa projekt név",
  "milestones": [],
  "projectID": "2af2c890-7aa2-4561-8f4e-d3ca1bb5c7a8",
  "users": [{ "userRole": "Készítő", "userID": "pelda.pal" }]
}
```

A projektek a JSON fájlokban az fenti képen látható struktúrában vannak tárolva.

```

{
  "projectStatus": "Folyamatban",
  "projectDescription": "Kell egy weblap sima HTML, CSS frontend, egyszerű Node.js backend a templatelt HTML kiszolgálására",
  "projectName": "Céges weblap",
  "milestones": [
    {
      "comments": [
        {
          "commentText": "A feltöltendő infók megvannak",
          "commenterID": "pelda.pal"
        }
      ],
      "milestoneName": "HTML,CSS frontend",
      "milestoneDescription": "Frontend elkészítése, egyszerű one-pager az információkkal",
      "milestoneStatus": "Folyamatban"
    },
    {
      "comments": [
        {
          "commentText": "Kész a backend, html templating és a routing mind működik és tesztelve lett",
          "commenterID": "pelda.peter"
        },
        {
          "commentText": "Remek, ez akkor kész is van",
          "commenterID": "pelda.pal"
        }
      ],
      "milestoneName": "Node.js backend",
      "milestoneDescription": "Backend elkészítése, lehet express is, templatelt html kiszolgálása",
      "milestoneStatus": "Elkészült"
    },
    {
      "comments": [
        {
          "commentText": "A fejléc design elkészült",
          "commenterID": "pelda.paula"
        }
      ],
      "milestoneName": "Weblap design elkészítése",
      "milestoneDescription": "Grafikai anyagok elkészítése a projekthez",
      "milestoneStatus": "Folyamatban"
    }
  ],
  "projectID": "4869821d-f043-4b4c-883b-78594bd72728",
  "users": [
    { "userRole": "Készítő", "userID": "pelda.pal" },
    { "userRole": "Alkalmazott", "userID": "pelda.peter" },
    { "userRole": "Admin", "userID": "pelda.paula" }
  ]
}

```

A fenti képen egy olyan projekt látszik, aminél létre lettek hozva mérföldkövek és hozzájuk kommentek is lettek fűzve, továbbá „users” tag alatt látható az is, hogy három felhasználó van a projekthez rendelve.

```

{
  "loggedin": "pelda.pal"
}

```

A fenti képen a „config.json” fájl tartalma látható, jelenleg csak azt tárolja, hogy melyik felhasználó van bejelentkezve. Hogyha a „loggedin” tag értéke egy létező felhasználó azonosítója, akkor program indulása esetén az adott felhasználó adatai kerülnek rögtön betöltésre.

```

{
  "users": ["pelda.pal", "pelda.peter", "pelda.paula"]
}

```

A fenti képen a „users.json” fájl tartalma látszik, a fájl tárolja az összes, az alkalmazásba regisztrált felhasználó azonosítóját.

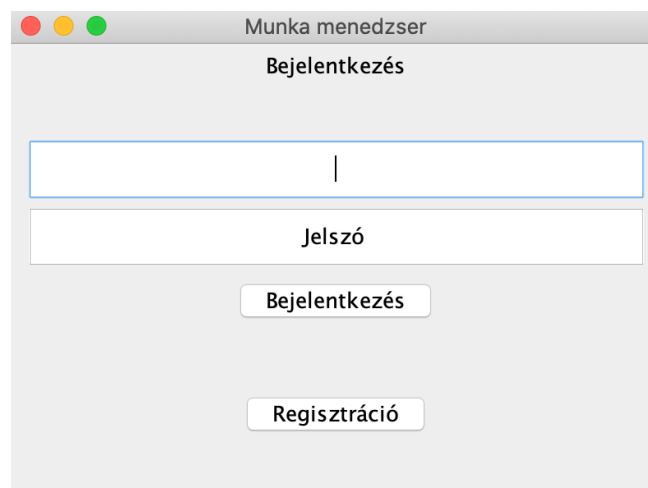
```
{
  "password": "palpal",
  "projectIDs": [
    "4869821d-f043-4b4c-883b-78594bd72728",
    "2af2c890-7aa2-4561-8f4e-d3ca1bb5c7a8"
  ],
  "userName": "Példa Pál",
  "userID": "pelda.pal"
}
```

A fenti képen egy felhasználó JSON fájlja és annak szerkezete látható. Látszanak a felhasználó adatai, továbbá látszik az is, hogy a felhasználó két projekthez van hozzárendelve.

3.6. Felhasználói interfész megvalósítása

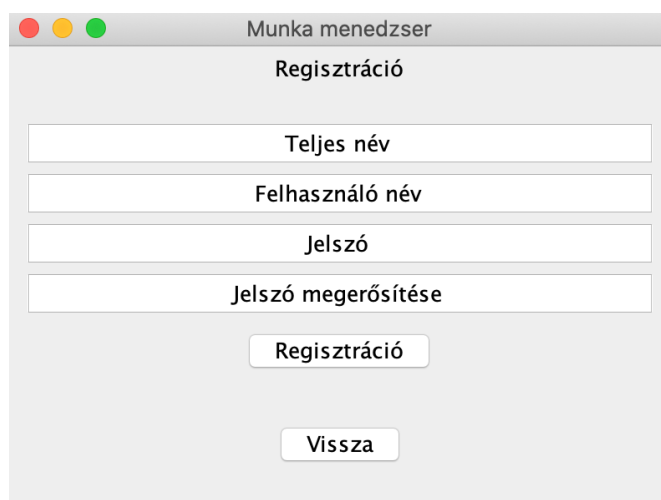
Az alkalmazás alapvetően három alap panelre van osztva, ezek közösen egy CardLayout-ban foglalnak helyet és váltogatni lehet közöttük. A három panel nevezetesen: bejelentkezési panel, regisztrációs panel és felhasználói panel. A három panelt a MainFrame osztály tartalmazza. Egy MainFrame referenciát minden panel a konstruktorában kap annak az érdekében, hogy a CardLayout elemei között egyszerűen lehessen igény esetén váltogatni.

3.6.1. Bejelentkezési panel



A fenti képen látható a bejelentkezési panel. Lehetőség van bejelentkezésre a szükséges adatok megadását követően, vagy átlépni a regisztrációs panelbe.

3.6.2. Regisztrációs panel



Munka menedzser

Regisztráció

Teljes név

Felhasználó név

Jelszó

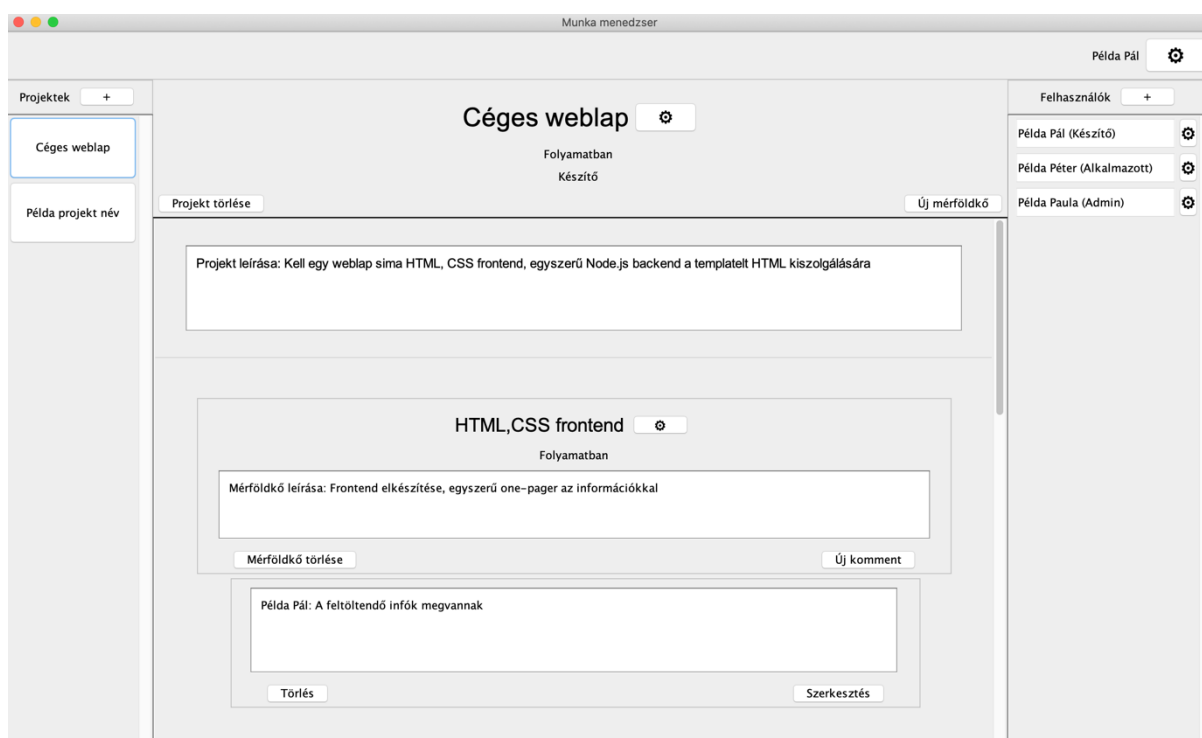
Jelszó megerősítése

Regisztráció

Vissza

A fenti képen látható a regisztrációs panel, itt lehet megadni a regisztrációhoz szükséges adatokat, majd ezt követően a regisztráció gombra kattintással lehet regisztrálni. A vissza gombbal a bejelentkezési panelbe lehet visszalépni.

3.6.3. Felhasználói panel



Munka menedzser

Példa Pál

Projekt +

Céges weblap

Példa projekt név

Céges weblap

Folyamatban

Készítő

Projekt törlése

Új mérőldő

Projekt leírása: Kell egy weblap sima HTML, CSS frontend, egyszerű Node.js backend a template HTML kiszolgálására

HTML, CSS frontend

Folyamatban

Mérőldő leírása: Frontend elkészítése, egyszerű one-pager az információkkal

Mérőldő törlése

Új komment

Példa Pál: A feltöltendő infók megvannak

Törölés

Szerkesztés

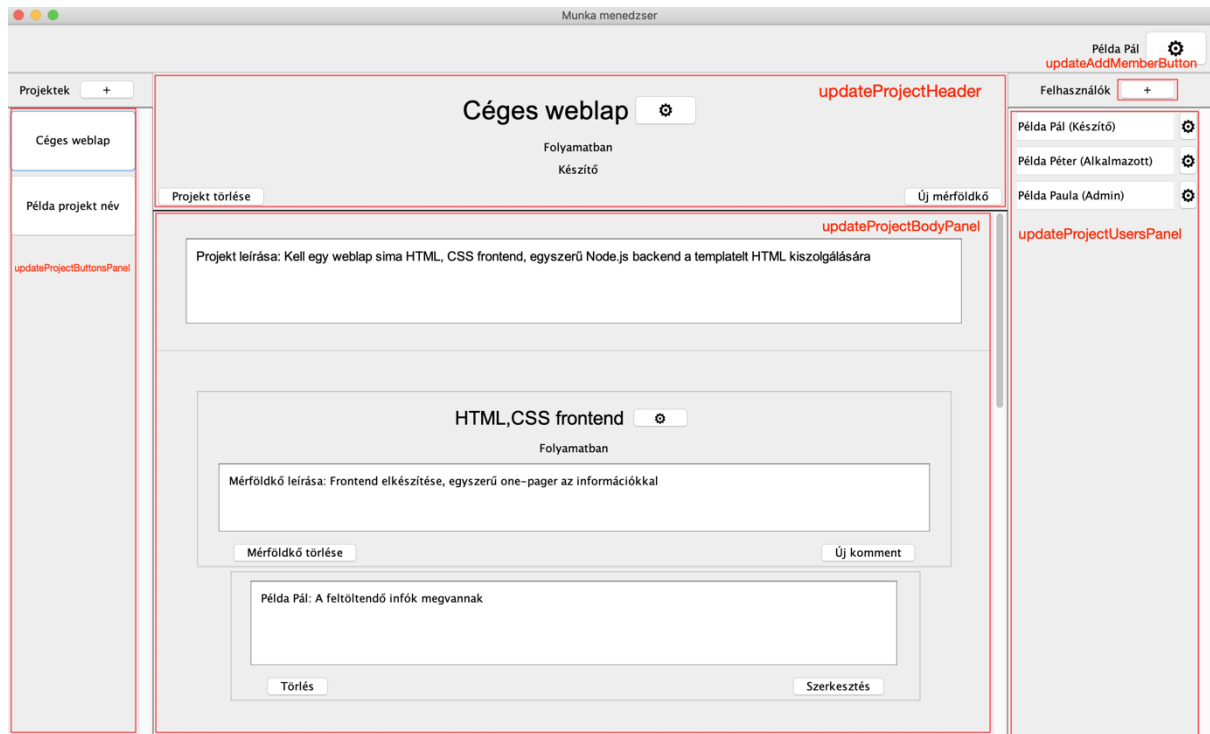
Felhasználók +

Példa Pál (Készítő)

Példa Péter (Alkalmazott)

Példa Paula (Admin)

A fenti képen látható a felhasználói panel, ez a program legösszetettebb része, itt zajlik a legtöbb folyamat. Az interfész itt kisebb egységekre van bontva, amik adatok változtatása esetén részletekben kerülnek frissítésre. Az alábbi képen láthatóak a különböző panelek, a jelzett szövegek pedig azoknak a funkcióknak a nevét jelölik, amellyel, hogyha a panelen belül lévő adat változik, frissíteni lehet a panelt.



3.7. A felhasználói interfészhez tartozó osztályok, listenerek és algoritmusok részletezése

A felhasználókon és projekteken belüli módosítások olyan módon vannak elkészítve, hogy először módosításra kerülnek a programban aktív objektumok, majd ezeknek az állapotai kerülnek kimentésre és az aktív objektumok módosított adatai kerülnek mindig betöltésre és azok látszódnak az interfészen.

3.7.1. Bejelentkezési panel

3.7.1.1. LoginPanel

A bejelentkezési panelhez tartozó interfészt hozza létre és regisztrálja a listener osztályokat a megfelelő gombokhoz.

3.7.1.2. LoginListener

A bejelentkezés gombhoz tartozó listener osztály. A bejelentkezés gomb megnyomását követően megkapja a bejelentkezési panelben megadott információkat és a `User.checkUserCredentials` metódus segítségével leellenőrizteti a belépési adatok helyességét. Ha a hitelesítési metódus igaz értékkel tér vissza akkor a felhasználói panel kerül betöltésre, ellenkező esetben egy hibaüzenetet követően újra meg lehet próbálni a bejelentkezést. Továbbá, hogyha sikeres a bejelentkezés, akkor meghívásra kerül a `CustomUtils` osztály `loginConfig` metódusa is, ami beállítja, hogy ki van bejelentkezve az alkalmazásba.

3.7.1.3. RegisterPanelListener

A login panelen belül lévő „Register” gombhoz tartozó listener, a gomb megnyomását követően a regisztrációs panel kerül betöltésre.

3.7.2. Regisztrációs panel

3.7.2.1. RegisterPanel

A regisztrációs panelt és a benne lévő input mezőket és gombokat hozza létre.

3.7.2.2. RegisterListener

A regisztrációs panelen belül lévő „Register” gombhoz tartozó listener. A register gomb megnyomását követően megkapja a regisztrációs panelben megadott információkat és meghívja a User.registerUser metódust, ami egy igaz-hamis értékkel tér vissza a regisztráció sikerességével kapcsolatban.

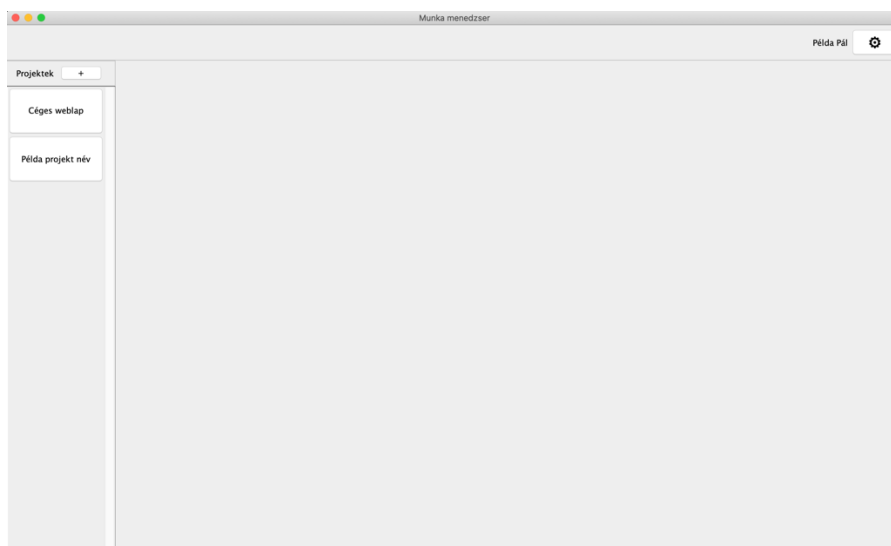
3.7.2.3. LoginPanelListener

A regisztrációs panel „Vissza” gombjához tartozó listener, megnyomását követően a bejelentkezési panel kerül megjelenítésre.

3.7.3. Felhasználói panel

3.7.3.1. UserPanel

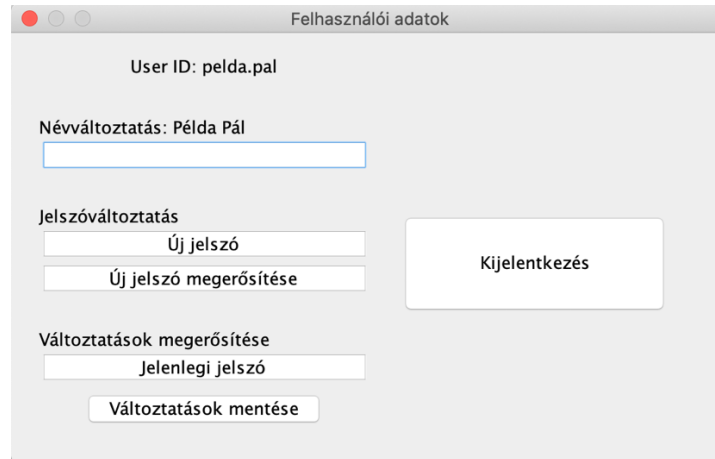
A felhasználói panel betöltéséért és kezeléséért felelős osztály. Konstruktorában megkap egy felhasználói azonosítót, aminek segítségével létrehoz egy User osztályt, ami a logikai műveleteket fogja végezni a UserPanel hívásainak hatására, amiket felhasználó inputok váltanak ki. A User osztály betölti a kapott felhasználóhoz tartozó adatok, majd a felhasználói panel a tőle kapott adatokat jeleníti meg. A felhasználói panel alapja egy „BorderLayout”, amiben eleinte két kisebb panel jelenik meg, az egyik a felhasználói panel fejléce, a másik pedig a bal oldalon megjelenő projekt gombok, amik megnyomás hatására az adott, a felhasználóhoz tartozó projekt adatait jelenítik meg. Konstruktorában továbbá megkap egy „MainFrame”-t is amit eltárol. Erre akkor van szükség, amikor kijelentkezik valaki a profiljából és vissza kell térni a CardLayout-on belül a LoginPanelbe.



Az fenti képen látható a bejelentkezést követő felhasználói panel.

3.7.3.2. UserSettingsListener

Ez a listener akkor hívódik meg, hogyha valaki a felhasználói panelben a fejlécen belül elhelyezkedő beállítások gombra kattint. Hatására előjön egy JOptionPane ami a következőképpen néz ki:



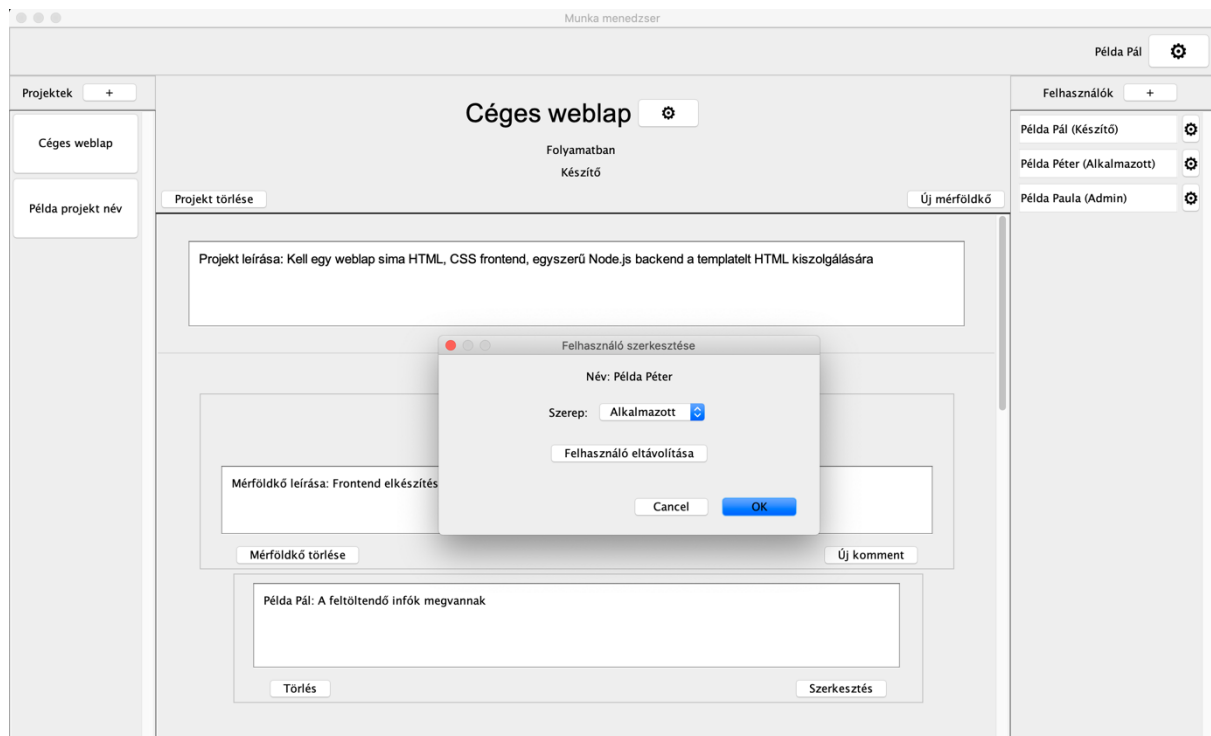
Ebben a panelben tudják a felhasználók az adataikat megváltoztatni. A panelben név és jelszóváltoztatásra van lehetőség a megváltoztatni kívánt adatok és a jelenlegi jelszó megadását követően. A „Változtatások mentése” gomb megnyomására meghívódik egy „UpdateUserDataListener”, ami a további lépéseket kezeli.

3.7.3.3. UpdateUserDataListener

Ez a listener megkapja a „Felhasználói adatok” panelben megadott új adatokat és a meghívja a User osztály „updateUserData” metódusát. A User osztály kezeli az adatok frissítését és egy igaz-hamis értékkel tér vissza, ami az adatok frissítésének sikerességét jelzi. Ha az adatok frissítésre kerültek, a felhasználó azt egy új JOptionPane-ben feltüntetett üzenetben láthatja. Ezt követően az adatok teljes frissítéséhez a felhasználónak újra be kell jelentkeznie. Azért esett ilyen megoldásra a választás, mert túl sok helyen kellett volna frissíteni az adatokat, ha a felhasználó a nevét változtatja meg, újabb bejelentkezés esetén viszont ezek alaptól jól töltődnek be. Ha az adatfrissítés sikertelen, akkor a felhasználó arról is egy JOptionPane-ben feltüntetett üzenet segítségével szerezhet tudomást.

3.7.3.4. RemoveMemberListener

Ez a listener arra szolgál, hogy egy adott felhasználót eltávolítsanak egy adott projektről. Ha egy arra jogosult felhasználó el akar távolítani a projektből egy másik felhasználót, akkor a projekten belül az eltávolítani kívánt felhasználó mellett található kis fogaskerékre kattintást követően előkerülő JOptionPane-ben teheti ezt meg, a JOptionPane-ben található „Felhasználó eltávolítása” gomb segítségével.



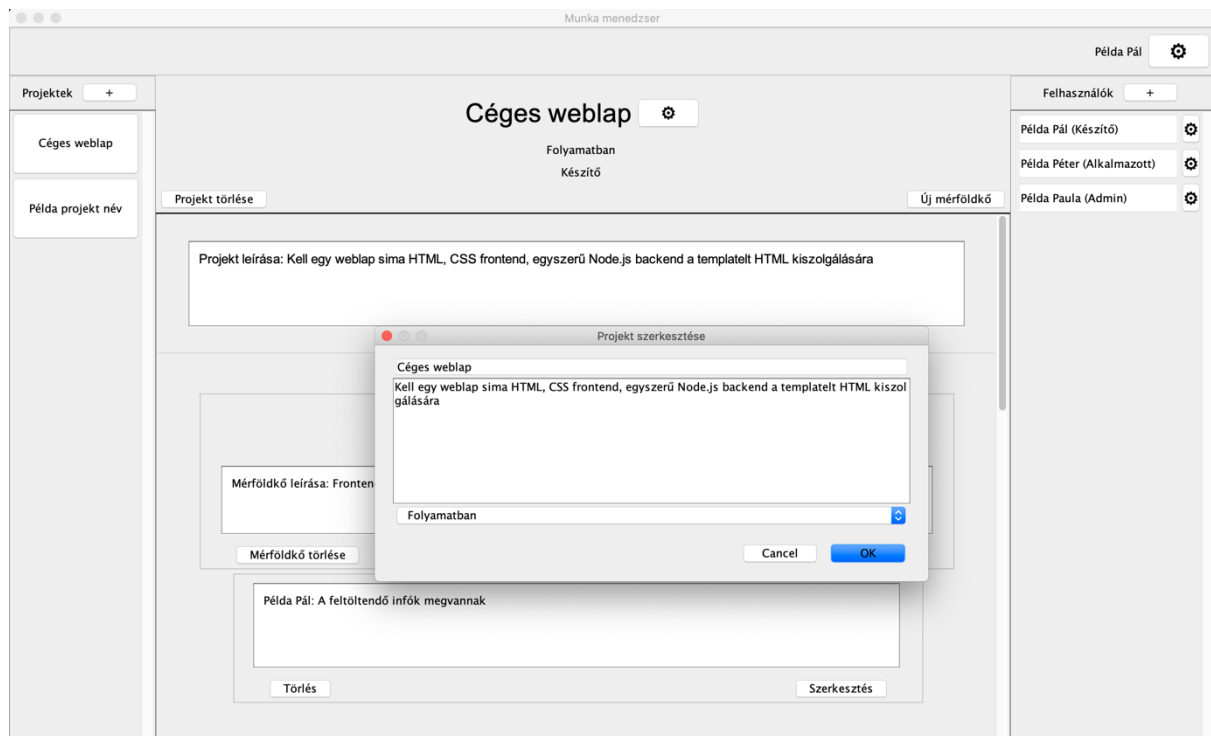
A fenti képen látható a projekt felhasználói panelben a szerkesztésre szolgáló fogaskerés, és a „Felhasználó szerkesztése” panel, amelyben a „Felhasználó eltávolítása” gomb található. Az eltávolítás gomb megnyomását követően meghívásra kerül az adott projekt `removeMember` metódusa, ami paraméterként átveszi az eltávolítani kívánt felhasználó azonosítóját. Az eltávolítást követően frissül a projekt felhasználóinak panelje.

3.7.3.5. ProjectUserEditListener

Ez a listener akkor hívódik meg, amikor egy arra jogosult felhasználó egy projekten belül egy másik felhasználót szeretne szerkeszteni és a felhasználó mellett lévő fogaskerékre kattint. Egy `JOptionPanel`-ben megjeleníti a szerkeszteni kívánt felhasználó nevét, szerepét és az eltávolítására szolgáló gombot. A 3.7.3.4 pontban szereplő kép „Felhasználó szerkesztése” panelt hozza fel konkrétan.

3.7.3.6. ProjectSettingsListener

A projekt fejlécében található, a projekt név mellett lévő fogaskerékre kattintást követően ez a listener hozza fel egy `JOptionPane`-ben a projekt szerkesztésére szolgáló ablakot.



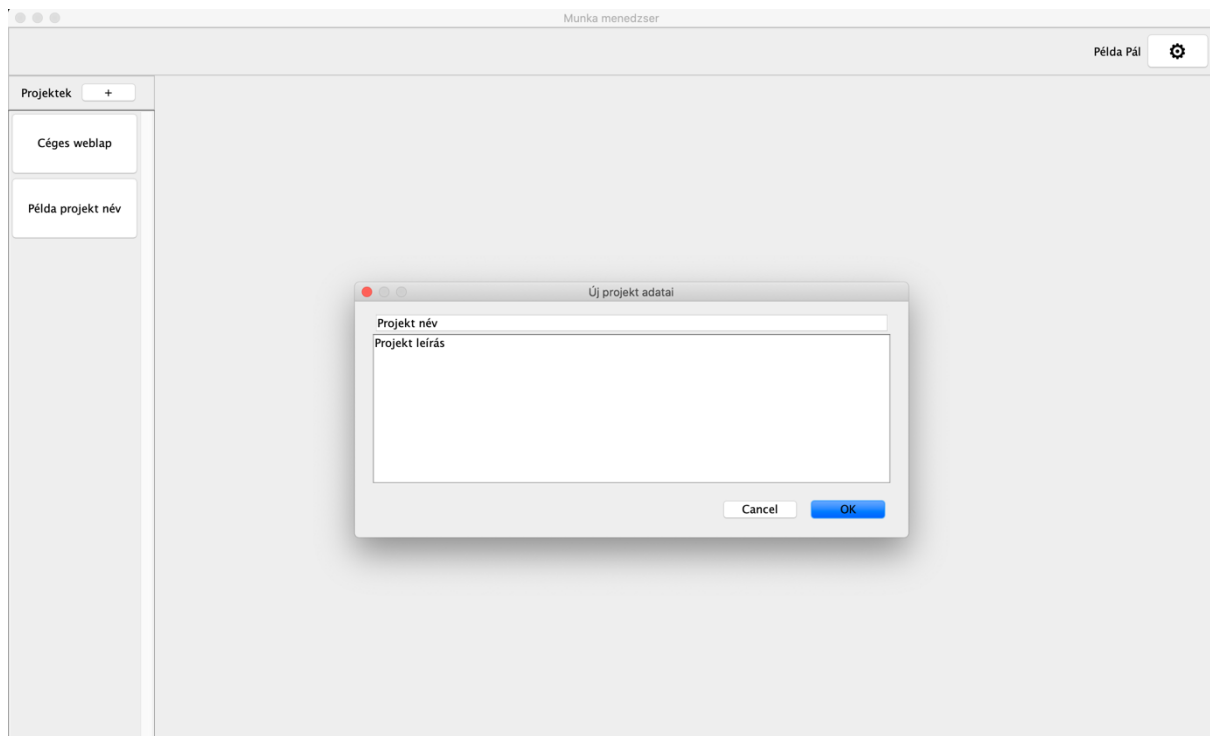
A fenti képen látszik a „Projekt szerkesztése” panel. Az új adatok megadását követően meghívódik az adott projekt `updateProjectData` metódusa, ami a megkapott új adatokkal kezeli a projekt frissítését. Ezt követően frissítésre kerülnek a panelek, amelyekben adatváltozás következhetett be, ezek lehetnek a projekt fejléce, projekt törzse vagy a felhasználó projektjeinek gombjai. Ezek az alábbi, a `UserPanel`-ben elhelyezkedő metódusokkal kerülnek frissítésre: `updateProjectHeaderPanel`, `updateProjectBodyPanel`, `updateProjectButtonsPanel`.

3.7.3.7. ProjectButtonListener

A felhasználói panel bal oldalán elhelyezkedő gombokon lévő listenerek. A listenerek a gombokhoz rendelés közben megkapják azt a projekt azonosítót, amelyet a gombra kattintást követően be kell tölteniük. Ha valamelyik gombra rákattint a felhasználó, akkor a `UserPanel`-ben lévő `activeProject` változó értéke a betölteni kívánt projekt értékét fogja felvenni. Ezután frissítésre kerülnek a nézetek, amelyek a projekt minden adatát megjelenítik, ezek a frissítések a következő, a `UserPanel` osztályban lévő metódusok segítségével mennek végbe: `updateProjectHeaderPanel`, `updateProjectBodyPanel`, `updateAddMemberButton`, `updateProjectUsersPanel`.

3.7.3.8. NewProjectListener

Ez a listener akkor hívódik meg, ha valaki a „Projektek” kiírás mellett lévő plusz gombra kattint. A megjelenő `JOptionPane` a következő:



A megjelenő panelben lehet megadni egy új, létrehozni kívánt projektnek az adatait. Adatok megadását követően az „OK” gombra kattintással lehet a projektet elkészíteni, ennek a hatására meghívódik a jelenleg bejelentkezett felhasználó, azaz User osztály, createNewProject metódusa, ami a kapott adatokkal az új projekt létrehozását végzi. Projekt létrehozását követően frissül a projekt gombok listája a UserPanelen belül lévő updateProjectButtonsPanel metódus segítségével.

3.7.3.9. NewMilestoneListener

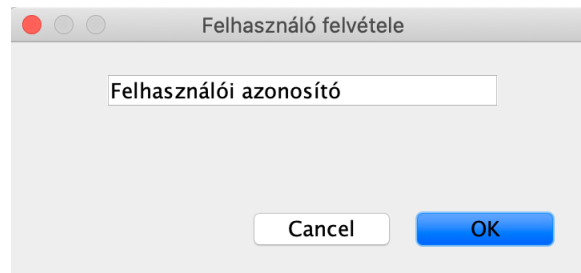
Projekten belül lévő „Új mérföldkő” gombra kattintást követően hívódik meg az adott listener, a megjelenő panel a következő.



A megjelenő „Új mérföldkő adatai” panelen belül lehet új mérföldkövet létrehozni. A szükséges adatok megadását követően meghívódik az aktív projekt addNewMilestone metódusa, ami a panelben megadott adatokat megkapva létrehozza a projekthez tartozó új mérföldkövet. Ezután frissítésre kerül a projekt törzse a UserPanel osztályban lévő updateProjectBodyPanel metódus segítségével és láthatóvá válik az új mérföldkő.

3.7.3.10. NewMemberListener

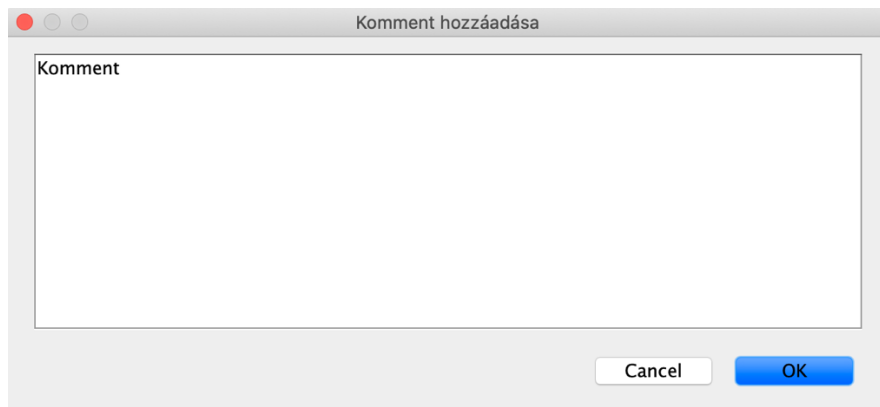
Egy projekten belül a jobb oldalon lévő „Felhasználók” kiírás mellett lévő plusz gombra kattintást követően ez a listener hívódik meg. A megjelenő panel a következő:



A „Felhasználó felvétele” panelen belül miután a felhasználói azonosító megadásra került, az „OK” gomb segítségével lehet a felhasználót a projekthez rendelni. Az „OK” gombra kattintást követően meghívásra került az aktív projekt `addNewMember` metódusa, ami megkapja a felvenni kívánt felhasználó azonosítóját. A metódus egy igaz-hamis értékkel tér vissza, ami felhasználó felvételének sikerességét jelzi. Ha nem sikeres a felhasználó felvétele, akkor ezt egy `JOptionPane`-el megjelenített hibaüzenet jelzi, ellenkező esetben a `UserPanel` osztályban lévő `updateProjectUsersPanel` metódus segítségével frissül a projekten belül lévő felhasználók listája és megjelenik az új felhasználó.

3.7.3.11. NewCommentListener

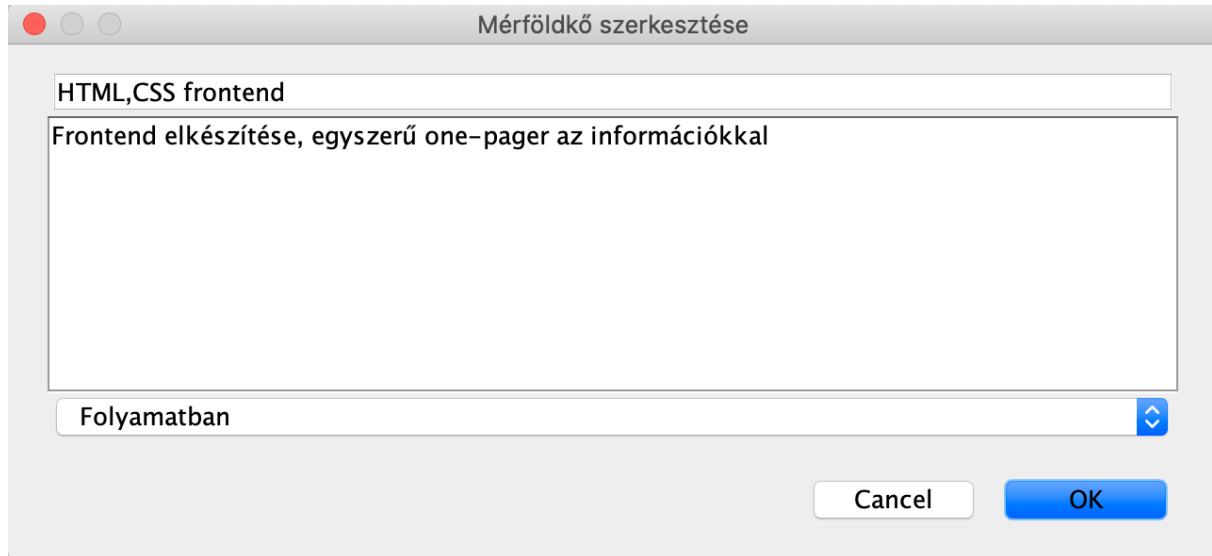
A mérföldkövek paneljein belül elhelyezkedő „Új komment” gombra kattintást követően ez a listener hozza fel a panelt melyben kommentet lehet írni. A listener gombhoz rendelése során a listener megkapja, hogy melyik mérföldkőhöz tartozóan ad hozzá kommenteket. A feljövő panel a következő:



A komment begépelését és az „OK” gomb megnyomását követően meghívódik az aktív projekt `addNewComment` metódusa, ami az adott mérföldkőhöz hozzáadja a kommentet, majd a `UserPanel` osztály `updateProjectBodyPanel` metódusával frissül a projekt törzse és láthatóvá válik az új komment.

3.7.3.12. MilestoneSettingsListener

Projektben belül, a mérföldkövek neve mellett lévő kis fogaskerékre kattintást követően hívódik meg és megjeleníti a mérföldkő szerkesztésére szolgáló panelt, ami a következő képpen néz ki:



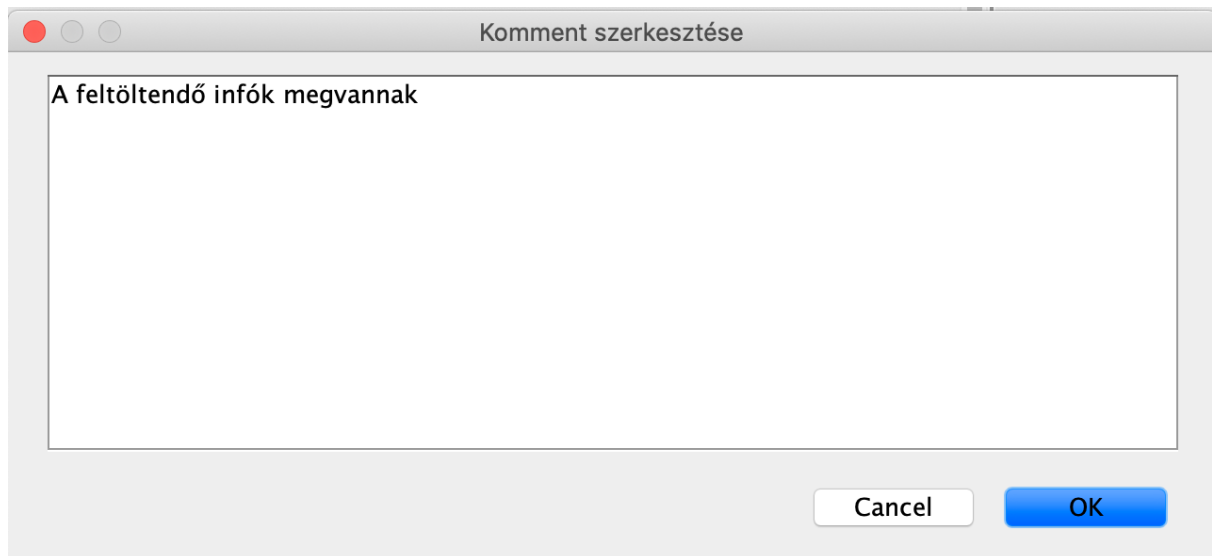
A kívánt adatok megváltoztatását és az „OK” gombra kattintást követően meghívódik az aktív projekt updateMilestoneData metódusa és átadásra kerülnek neki az adatok. Az adatok frissítését követően a nézet is frissítésre kerül a UserPanel osztály updateProjectBodyPanel metódus segítségével, és láthatóvá válnak a mérföldkő módosításai.

3.7.3.13. LogoutButtonListener

A felhasználói panel fejlécében lévő fogaskerékre kattintást követően feljövő „Felhasználói adatok” panelen belül elhelyezkedő „Kijelentkezés” gomb megnyomására hívódik meg. Hatására meghívódik a CustomUtils osztály logoutConfig metódusa, ami a config fájlban beállítja, hogy senki nincs bejelentkezve.

3.7.3.14. EditCommentListener

Projektben belül egy adott mérföldkőhöz tartozó komment paneljén belül elhelyezkedő „Szerkesztés” gomb megnyomását követően meghívásra kerülő listener. Megjeleníti a „Komment szerkesztése” JOptionPane alapú panelt, amelyben az előzőleg megírt komment átírása végezhető. Komment átírását követően az „OK” gombra kattintással meghívódik az aktív projekt updateCommentData metódusa, ami a komment frissítését elvégzi, majd a UserPanel updateProjectBodyPanel metódus meghívását követően frissítésre kerül a komment az interfészen is. Az alábbi képen látható a „Komment szerkesztése” panel.



3.7.3.15. DeleteProjectListener

Projekt fejlécében elhelyezkedő „Projekt törlése” gombra kattintást követően meghívódó listener. Hatására az aktív projekten meghívásra kerül a deleteProject metódus, ami törli minden egyes felhasználótól az adott projekt azonosítóját, majd törli a projekt fájlját is. Törlést követően a UserPanel osztályban lévő activeProject értéke null-ra állítódik és az interfész frissítésre kerül.

3.7.3.16. DeleteMilestoneListener

Projekten belül lévő mérföldkő fejlécében elhelyezkedő „Törlés” gomb hatására meghívódó listener. Meghívja az aktív projekt deleteMilestone metódusát, ami törli a kapott mérföldkövet, ezt követően frissítésre kerül a projekt törzse a UserPanel updateProjectBodyPanel metódusának segítségével, és láthatóvá válik az, hogy a mérföldkő törlésre került.

3.7.3.17. DeleteCommentListener

Projekten és mérföldkőn belül elhelyezkedő kommenthez tartozó „Törlés” gomb megnyomására meghívódó listener. Hatására meghívásra kerül az aktív projekt deleteComment metódusa, ami törli a projektből a kapott kommentet. Ezt követően a UserPanel updateProjectBodyPanel metódusának meghívására történő interfész frissülés után láthatóvá válik, hogy a komment törlésre került.

4. Program tesztelése

A program azon osztályai melyek fontos logika műveleteket végeznek mind tesztelésre kerültek, ezek a tesztesetek és a tesztesetek részletes leírásai mind megtalálhatóak a forráskódban. A tesztesetek a projekt forráskódjának /test mappájában találhatóak meg, a tesztelésre került osztályok nevei a következők: CustomUtils, JSONHandler, Milestone, Project, User.