Sörös Ádám

Papp János István

Tálas Martin

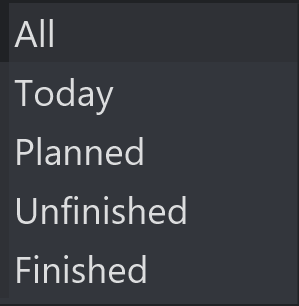
Adatbázis-vezérelt Todo Lista

# A feladat leírása

A feladat egy olyan teendő (todo) lista kezelő szoftver megtervezése és megvalósítása volt, amely képes több felhasználót kezelni, a felhasználók között képes kapcsolatot létrehozni és a felhasználókhoz tartozó teendő objektumokat megfelelően tudja rendszerezni, hogy a felhasználó minél letisztultabb képet kapjon a teendőiről. A szoftver rendelkezik bejelentkezési és regisztrálási funkciókkal, bejelentkezés után pedig a felhasználót rögtön fogadja az összes általa specifikált teendő, táblázatba szedve. A felhasználó képes szűrni a teendőit mai napi, tervezettségi, be nem fejezett és befejezett menüpontok szerint. Felhasználók képesek csoportokat létrehozni egymás között, csoportba csatlakozni és meghívót küldeni. A szoftver rendelkezik erős háttér adatbázissal is, ami az összes felhasználóhoz, csoporthoz és teendőhöz kapcsolódó adatot eltárolja.

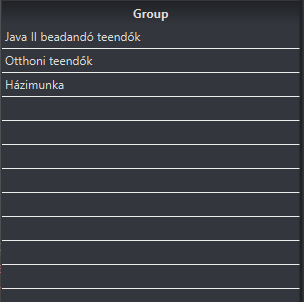
# A szoftver funkciói

1. A teendők kategória szerinti rendszerezése
2. A felhasználók és teendők csoportba osztása
3. Meghívó küldése felhasználótól felhasználónak csoporthoz való csatlakozás szándékával
4. Teendő hozzáadása a felhasználóhoz vagy a csoporthoz
5. Csoport létrehozása
6. Meghívók kezelése
7. Teendők megjelenítése
8. Teendőkhöz tartozó leírás megjelenítése
9. Teendők lejárati dátum szerinti szűrése
10. Bejelentkezés
11. Regisztráció
12. A teendőket kategória szerint rendszerező menüpontok a szoftverablak bal szélén helyezkednek el (1. ábra).



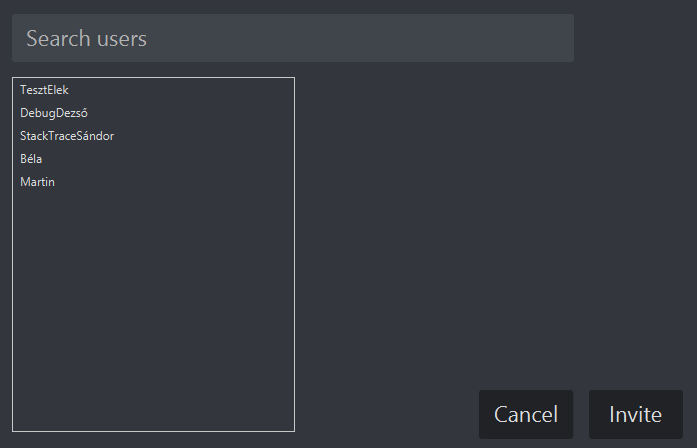
1.Ábra: A teendők szűréséhez használt menüpontok

1. Ezen menüpontok alatt helyezkednek el a táblázatba rendezett felhasználó csoportok (2. ábra), amiket bármely felhasználó tud készíteni, azonban csoportba való belépéshez meghívó szükséges a csoport készítőjétől. A csoportok feladata, hogy a felhasználók gyorsan és szervezetten tudjanak olyan teendőt létrehozni, amely minden felhasználó számára szükséges lesz, eliminálva ezzel azt a munkát, hogy minden felhasználónak külön kelljen manuálisan teendőt létrehoznia. Minden csoportsor rendelkezik kontextus menüvel, ami a meghívást a csoportból való távozást és a csoporthoz teendő hozzáadását kezeli.



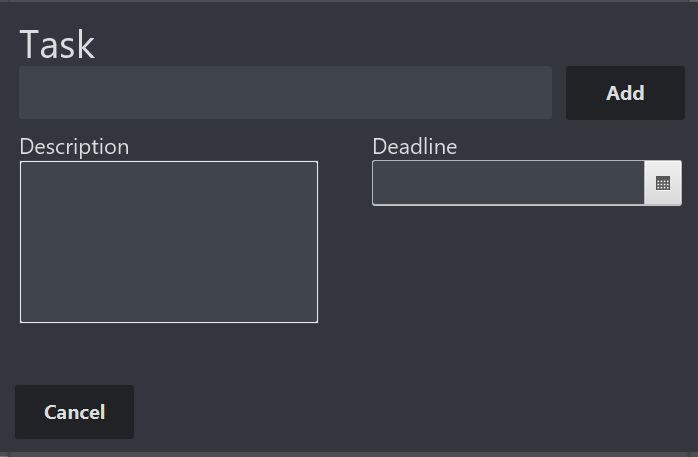
2.Ábra: A csoportok táblázat

1. A meghívó ablak (3. ábra) összegyűjti az összes regisztrált felhasználót és az általunk kiválasztottaknak az „Invite” gomb megnyomásával meghívót küld az adott csoportba. A meghívó ablak rendelkezik még keresési mezővel is, ami gombnyomás nélkül, billentyűlenyomásra keres a felhasználó listában.



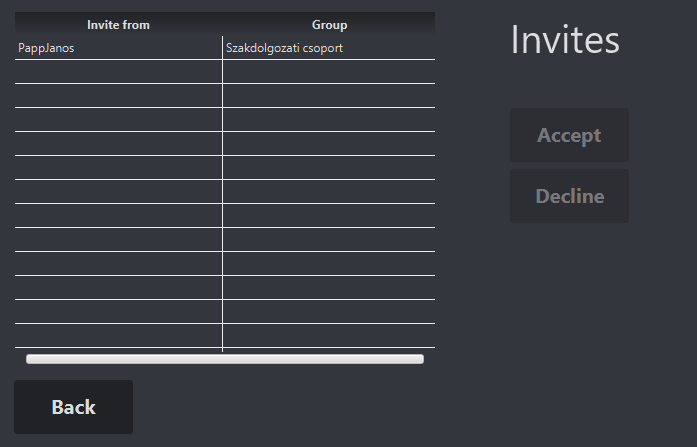
3.Ábra: A meghívó küldését lebonyolító ablak

1. A csoporthoz tartozó teendők, külön nyithatók meg a saját teendőinktől, kétszer kell gyors egymásutánban kattintanunk a csoport nevére. A csoporthoz való teendő hozzáadásának ablaka (4. ábra) megegyezik a saját teendőt hozzáadó ablakkal, annyi különbséggel, hogy a csoporthoz és készítőhöz is hozzáadja a teendőt.

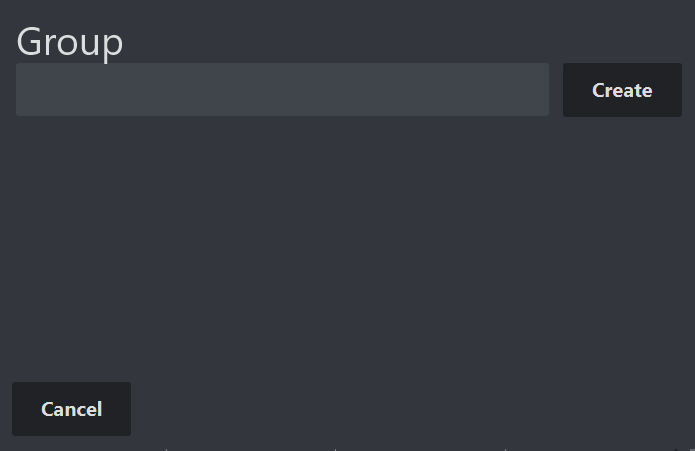


4.Ábra: A teendő hozzáadását lebonyolító ablak

1. Csoport létrehozását lebonyolító ablak (5. ábra) név megadása után létrehoz egy csoportot, amit rögtön megjelenít a csoportokat rendszerező táblázat.
2. Abban az esetben, ha egy felhasználótól meghívót kapunk, megjelenik a meghívó az „Invites” táblázatban (6. ábra) és elfogadhatjuk a meghívót a meghívó kijelölésével és az „Accept” gomb megnyomásával. Ha esetleg nem kívánunk csatlakozni a meghívott csoporthoz, a „Decline” gombra nyomva elutasíthatjuk a kérelmet.

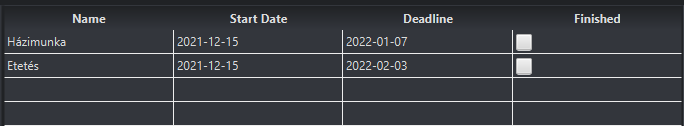


6.Ábra: A meghívókat kezelő ablak



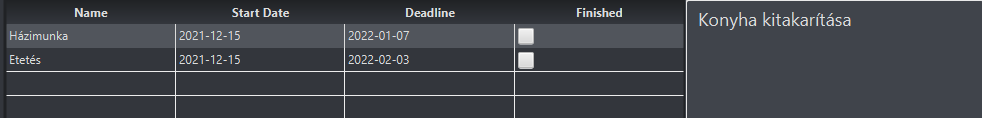
5.Ábra: A csoport hozzáadását lebonyolító ablak

1. A teendőinket jeleníti meg a szoftverablak közepét kitöltő táblázat (7. ábra). Ehhez a táblához kapcsolódnak az előbb említett szűrők és csoportok. A sorok ebben a táblázatban is rendelkeznek kontextus ablakkal, ennél a táblázatnál azonban csak törlési funkció áll rendelkezésünkre.



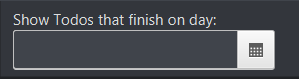
7.Ábra: A fő teendő táblázat

1. Táblázat megjeleníti a teendő nevét, a dátumot a teendő készítésekor, a határidőt, valamit azt, hogy befejeztük-e a teendőt. Ha rákattintunk a teendőre a jobb oldalon megjeleníti a szoftver a teendőhöz tartozó leírást is (8. ábra).



8.Ábra: A fő teendő táblázat a leírással

1. A teendőket lehet még határidő szerint is szűrni, ezt egy dátum választó bemeneti mező segítségével érjük el, ami a dátum kiválasztásakor csak azokat a teendőket jeleníti meg amik a specifikált napon járnak le (9. ábra).



9.Ábra: Lejárati dátum szerinti szűrés

# Feladatmegosztás

A feladatot backend, frontend és a kettő között kapcsolatot létrehozó részekre osztottuk.

* Sörös Ádám, SQL nyelvel való széleskörű tapasztalata miatt a backend feladatköreit látta el. Az ő feladata volt azokat az osztályokat, adattagokat és függvényeket megírnia, amik lefuttatásával adatokat adhattunk át és nyerhettünk ki az adatbázisból. Hozzájárulása elengedhetetlen volt az adatok megfelelő tárolásában és kezelésében, valamint az adatbázis szervezése mellett bizonyította tudását a Java programozási nyelvben is, hiszen gond nélkül futó, könnyen kezelhető és letisztult funkciókat biztosított a projekt számára.
* Tálas Martin, ahol lehetséges volt, CSS styling nyelv segítségével oldotta meg a szoftver kinézetét, valamint a SceneBuilder szoftver segítségével alakította ki a GUI elrendezését. A SceneBuilder szoftver a frontend feladatokon kívül a csapat többi tagjának is biztosított funkciókat, mint például id-k és függvények hozzárendelése a különböző komponensekhez és eseményekhez. Tálas Martin hozzájárulása a feladathoz biztosította a fejlesztés zökkenőmentes folyamatát, mivel mindig időben és pontosan végezte a frontend frissítéseket a csapat többi tagja számára. Java programozási tudása kimagasló volt a CSS utasítások Java kód béli implementálása során.
* Papp János István a beillesztett komponensek és az adatbázis funkciók közötti kommunikációt valósította meg, az ő hozzájárulásának köszönhető, hogy a megfelelő adatok a megfelelő időben és helyen jelentek meg a szoftver táblázataiban és felhasználói felületein. Feladatköre főként a SceneBuilder szoftverben beillesztett komponensek körül mozgott, azonban kitért bizonyos olyan elemekre is, amelyek a szoftver hátterében helyezkedtek el, távol a felhasználó szemeitől. Tudása a Java programozási nyelvben biztosított egy stabil, letisztult munkafolyamatot a szoftver számára, hozzájárulása pedig biztosította a kapcsolatot a szoftver főbb elemeinek számára.

# Önreflexió

Papp János István

Első találkozásomkor a feladattal félelem, de ugyanakkor izgalom fogott el. Rengeteg olyan ötlettel és tervvel találkoztam, amikre akkor még nem tudtam a választ, de a gondolat, hogy nekiveselkedhetek és megoldást találhatok ezekre a problémákra elöntött önbizalommal és kitartással. A JavaFX komponensekkel ezen félév során találkoztam először, azonban a dokumentációk átnézésével, órai anyag felhasználásával és különböző online források segítségével szert tettem olyan tudásra belőlük, ami lehetővé tette számomra, hogy igényes munkát végezzek az általunk fejlesztett szoftverben. Csapattársaim kifejezték elégedettségüket a munkámmal kapcsolatban és örömömre vált, hogy ez a projekt olyan új ismereteket tett a tudásomhoz, amiket jövőbeli pályafutásom során magabiztosan alkalmazhatok majd. Összességében pozitív élményeim voltak a projekt elkészítése során, és még ha néha el is fogott a kétség, sikerült hamar megnyugvást találnom akár egyedül vagy akár a csapattársaim segítségével.

Sörös Ádám

Adatbáziskezeléssel java1-en és egyéb tantárgyakon is találkoztam de a Jávában még nem volt sok tapasztalatom, de eddig főleg egyéni feladatoknál. A fő kihívást azt jelentette, hogy az elkészített adatbázis elérések olyan formába adják vissza az adatokat, hogy azok a lehető legáltalánosabbak legyenek (hogy a legtöbb helyen újra felhasználtóak legyenek) de ugyanakkor a frontend számára a lehető lekevesebb (vagy semmilyen) extra művelettel felhasználható legyen és emelet még kényelmesen is lehessen használni.

A projekt végére szerintem a 2.célt (lehető legkevesebb extra műveletet) sikerült elérnem, de az elsőt kárára sikerült csak megvalósítani. A projekt sok tapasztalatot adott nekem nem csak az adatok kezelésében, hanem abban is hogy menyire fontos a tudatos előre tervezés nem csak az objektumok és azok használata hanem azoknak a jövőbeli tarolásai szempontjából is.

Tálas Martin

A JavaFX frontend fejlesztéssel már korábban is találkozta középiskolai éveim alatt, viszont akkoriban még nem mélyültem el benne. Most a Java II. tantárgy keretein belül volt alkalmam jobban beleásni magam a témába és a JavaFX dokumentációk és az órai anyagok segítségével meg tudtam birkózni a témával. Tervei között szerepelt egy letisztult felhasználói felület (GUI) elkészítése, amelyen könnyedén el lehet navigálni, és a szemünket is kíméli, ezért sötét témában készült el. A Megvalósítást CSS használatával oldottam meg. Bár hallottam korábban, hogy a JavaFX használ CSS-t, de még ne próbáltam, eddig. Nehézség volt az első pillanatokban, ugyanis nem minden elemet lehet CSS fájl segítségével formázni, valamit az FXML fájlban kell megtennünk, ezeket főként Scene Builder segítségével oldottam meg. Úgy érzem sikerült megbirkóznom ezzel a feladattal is, és egy letisztult kinézetet adni az alkalmazásunknak. Végezetül azt kell, hogy mondjam, az elején aggódtam, hogy hogy oldom meg, de végül sikerült mindent megcsinálni, csapatként, amit elterveztünk.