

SAPIENTIA ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM MAROSVÁSÁRHELYI KAR,
INFORMATIKA SZAK



SAPIENTIA
ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM

To Do App

Software rendszerek tervezése

Projektet készítette:

Nagy Hunor-Zalán

Vezető tanár:

Dr. Szántó Zoltán

Tartalomjegyzék

| | |
|---------------------------------------|----|
| 1. Bevezetés..... | 3 |
| a. Bemutató3 | 3 |
| b. Célkitűzések3 | 3 |
| 2. Követelmények..... | 4 |
| a. Felhasználói követelmények4 | 4 |
| b. Rendszer követelmények5 | 5 |
| 3. Diagramok..... | 7 |
| 4. Technológiák..... | 9 |
| a. Web9 | 9 |
| b. C#.....15 | 15 |
| c. Firebase18 | 18 |
| d. Verziókövetés19 | 19 |
| e. Design.....21 | 21 |
| 5. Továbbfejlesztési lehetőségek..... | 21 |
| 6. Összefoglalás..... | 22 |

1. Bevezetés

b. Bemutató

A projekt célja, hogy egy jól felépített feladatlista létrehozása segítségével, egy XXI. századi ember könnyedén eligazodni tudjon a mindennapi feladatai közt. Kigondolja saját feladatait, melyeket bevezetheti a felületre, akár az applikáció, akár a weboldal által. Azon a napon, mikor az adott feladatot el kell végeznie, a program figyelmeztet és értesítést küld. A felület egy demo, mely egy felhasználóra van elkészítve.

b. Célkitűzések

Az applikáció és weboldal tervezése közben számos cél lett kitűzve, amelyeket úgy gondolok, hogy fontos a szoftver működése szempontjából, illetve a jó felhasználói élmény eléréséhez szükségesek.

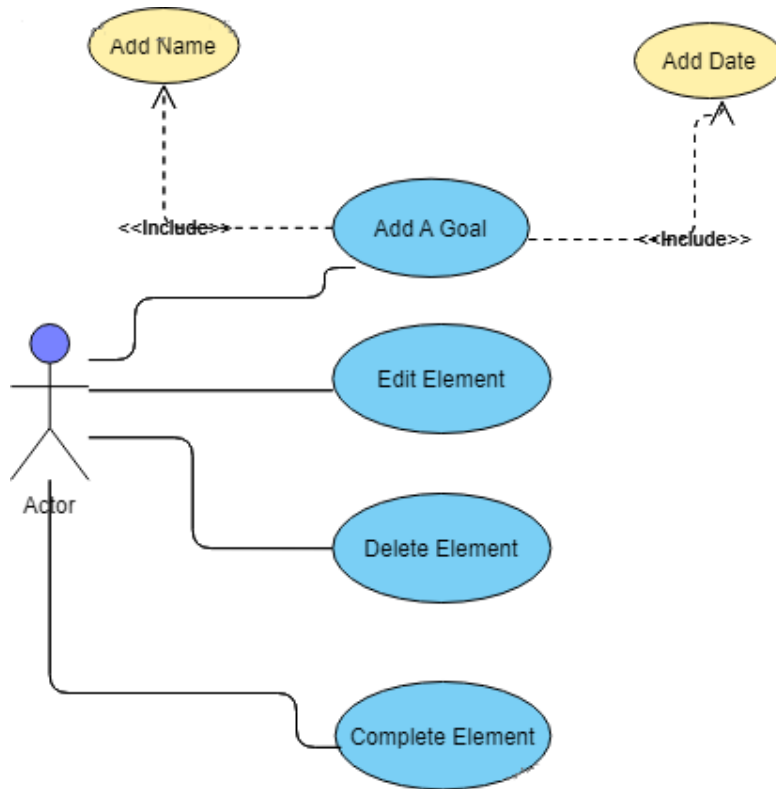
Ezen célok közé tartozik az, hogy a felhasználó nem csak hozzáadni tudjon egy tevékenységet, hanem azt szerkeszteni és törölni is tudja. Valamint ha azt nem teljesítette és a megadott időpont lejárt, akkor azt átlássa.

További célkitűzés az is, hogy a felhasználó a pontos időt ne keljen keresgélje, hanem az is feltüntetve legyen számára.

2. Software Követelmények

a. Felhasználói követelmények

Use case diagramm:



| Use Case | Add A Goal |
|---|---|
| Pre-condition (előtte) | Internet hozzáférési lehetőség szükséges. |
| Post-condition (utána) | Az adott ToDo bekerül az adatbázisba és a képernyőre. |
| Basic path (út) | A felhasználó megnyitja a weboldalt/Desktop appot. Beírja a ToDo nevét és kiválaszt egy dátumot. Rákattint a gombra, ami ellenőrzést hajt végre, majd megjelenik a képernyőn és az adatbázisban |
| Alternative path (más opció) | - |
| Exceptional path (lehetséges hiba) | Ha nem üresek a mezők, akkor megjelenik a képernyőn és az adatbázisban. |

b. Rendszer követelmények

Az továbbiakban néhány főbb funkcionális és nem funkcionális követelmény következik:

- Funkcionális követelmények:

Az app/weboldal, ha meg van nyitva értesítést küld az aznapi tevékenységekről, melyek be vannak már vezetve. Felsorolja azokat, vesszővel elválasztva. A felhasználó hozzáadhat egy eseményt, dátummal párosítva. Ha valamelyik részt nem tölti ki, egy ablak jön elő, ami jelzi a helytelen adatbevitelt és így nem adhat hozzá feladatot. Ha hozzáadtunk egy eseményt az megjelenik a folyamatban lévő tevékenységek részben. Itt az eseményt lehet szerkeszteni, mint név, mint pedig idő szempontjából is. Az eseményt lehet törölni. Valamint hozzá lehet adni a befejezett események listájához. A befejezett eseményeket lehet törölni vagy visszavonni a nem elvégzettek közé. Ha a hozzáadott tevékenység régebbi, mint az aznapi időpont, akkor a lejárt események listájába kerül. Itt szintén lehet törölni vagy hozzáadni az elvégzett eseményekhez.

Gombokkal elvégezhető események:

- add: A To Do List-hez elemet ad hozzá. Dátum és név formájában.
(Enter)
- done: A To Do List-ből elemet helyez át a Finished List-be. (Plusz)
- delete: Töröl a teljes adatbázisból és képernyőről. (Kuka)
- edit: Szerkesztés lehetőségét adja meg név szerint. (Ceruza)
- minus: Elemet helyez át az Expired List-ből a Finished List-be. Ide akkor kerül elem a To Do List-ből, ha az aktuális dátumhoz képest a kurrens elem dátuma már a múltnak tekinthető. (Minus)

- frissítés: A Desktop appot frissíti, ugyanis ha sok módosítást, vagy törlést csináltunk már nem átlátható (Refresh)

Az applikáció, illetve a weboldal mutatja az aktuális időt.

Alert-eket, illetve MessageBox -ok használata.

Web, alert-ek:

This page says

Today's program: Gym, Study

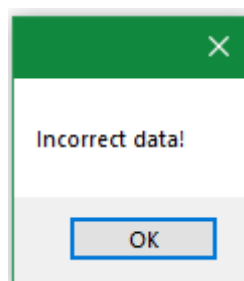
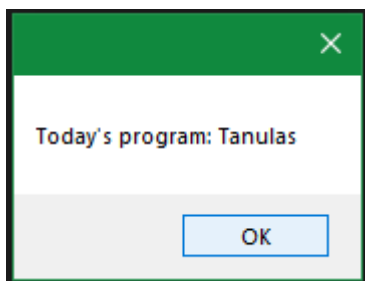
OK

This page says

Incorrect data!

OK

C#, MessageBox:



- Nem funkcionális követelmények:
 - Termék követelmények:

A felhasználhatóság szempontjából a weboldal az ismertebb browser-ekben (chrome, mozilla, opera, edge) működik és használható, míg a Desktop app csakis windowson használható, azon belül igényel minimum 7-es win-t.

Hatékonyság szempontjából, ha megfelelő az internet sebessége, az alkalmazás, illetve weboldal néhány mini szekundum alatt betölt és műveleteket végez.

Megbízhatóság szempontjából a fejlesztés során a hibalehetőségek tesztelve voltak, de nem lehet kizárni a potenciális összeomlás lehetőségét sem.

- Külső követelmények:

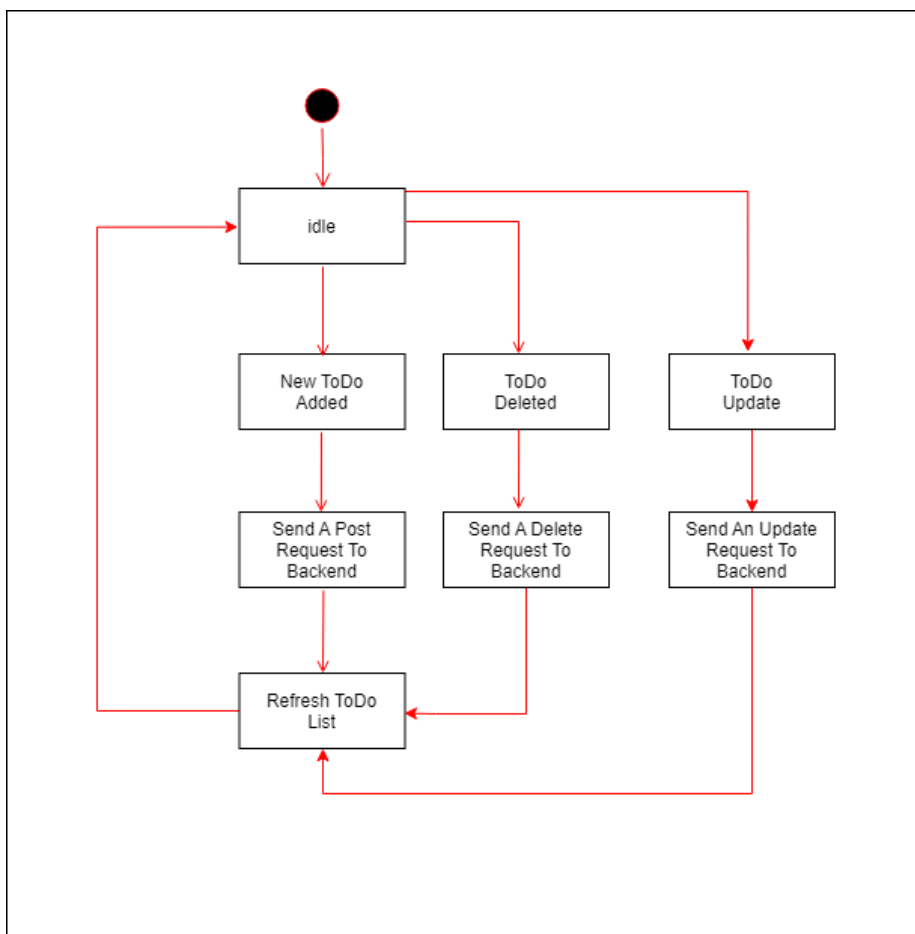
Összeférhetőségi szempontból az alkalmazás semmilyen más befolyással nem bír más alkalmazásokra tekintve, tehát teljesen elkülönülhető a már meglévő alkalmazásoktól.

Az alkalmazás etikai szempontból semmilyen adatot nem szolgáltat ki küldő forrásoknak.

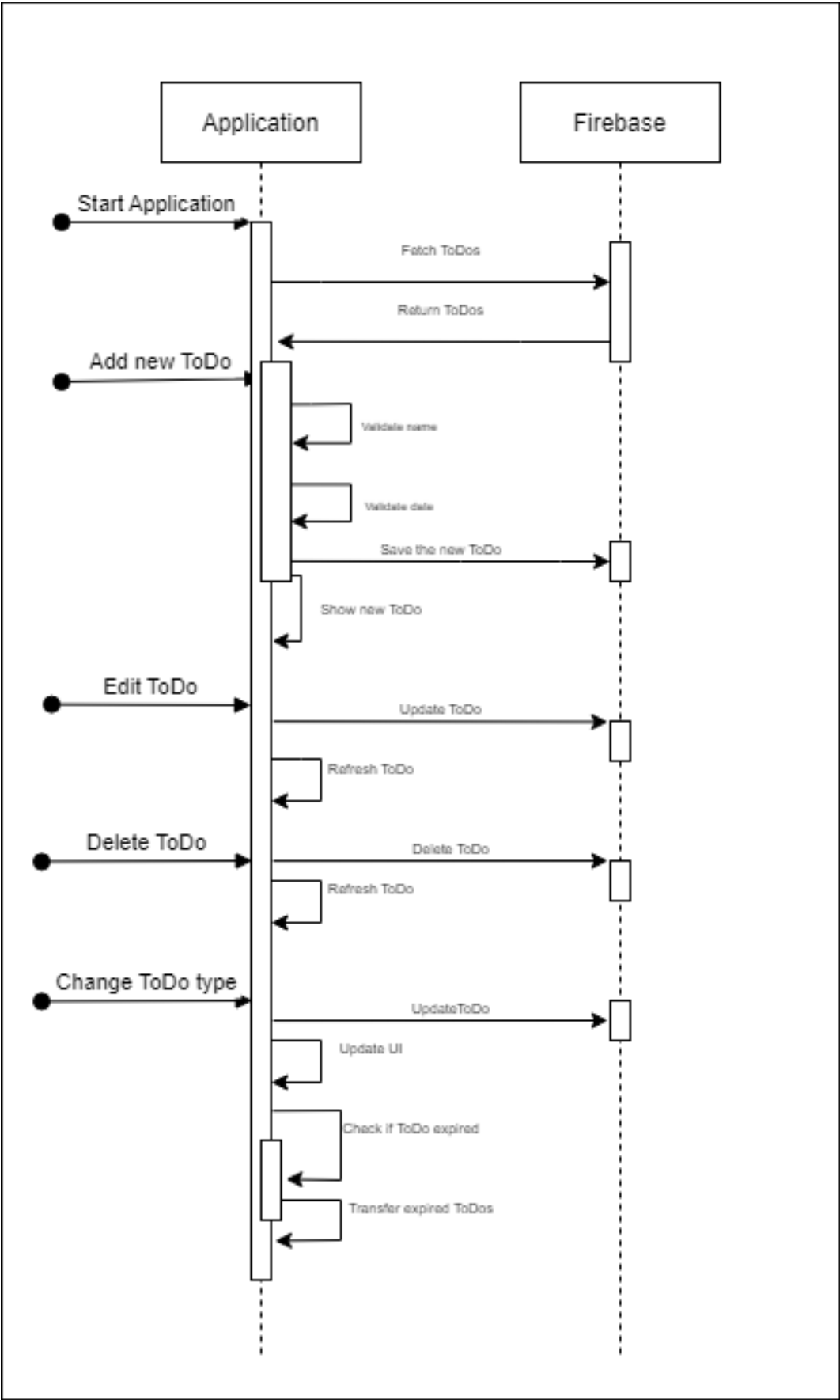
3. Diagramok

Mivel a projekt két felületből tevődik össze, így szükségessé vált egy közös adatbázis használata, mely mindkét platformot egységesen kiszolgálja.

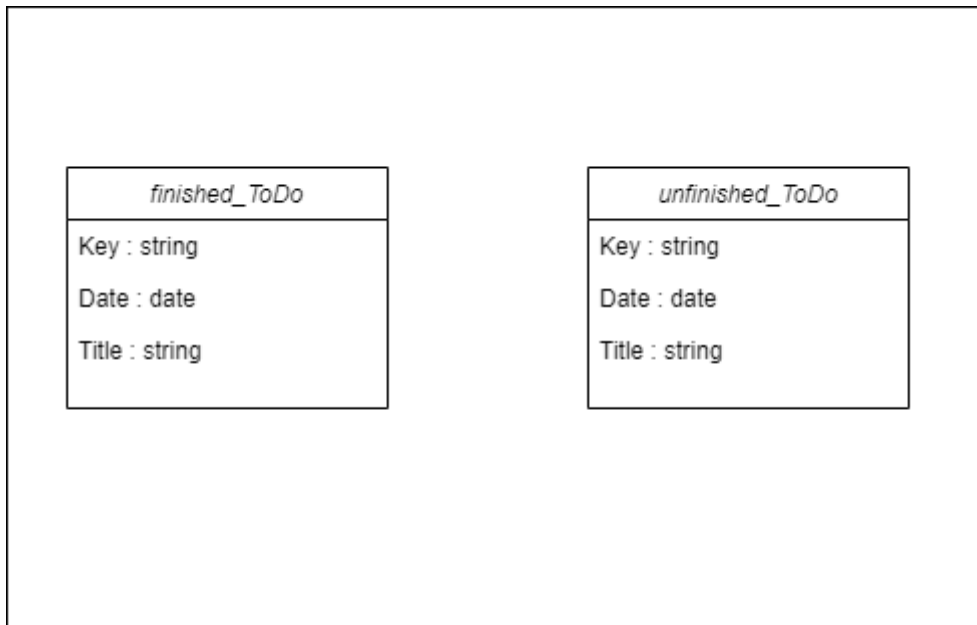
Activity diagram:



Szekvencia diagram:



Class diagram:



4. Technológiák

A továbbiakban a felhasznált programozási nyelvekről és technológiákról lesz szó.

a. A weboldal megvalósítása a következőképpen történt:

A weboldal elkészítéséhez nincs szükségünk semmi féle nagyméretű applikációhoz, vagy programhoz. Ennek megvalósítása akár egy sima Notepad-ban is elkészíthető, és browser segítségével futtatható.

Jómagam, Visual Studio Code-ot használtam, amihez telepítettem néhány extension-t, amik segítik/segítették munkám. Három, illetve három féle file típust használtam a weboldal megvalósításához (html, css, js).



A weboldal értesít, ha van aktuális aznapi tevékenység. Az oldal újra tölti önmagát 5 percenként, így ha még mindig van aktuális tevékenység újra értesít küld.

A weboldal linkje: <https://nhztodoapp.netlify.app/>

A kód, mely működteti a weboldalt, a következő fájlokban található:

- index.html
- index.js
- index.css

A Firebase használata a weboldalon, az adatbázishoz való kapcsolódási JavaScript kódja:

```
1  // My web app's Firebase configuration
2  var firebaseConfig = {
3    apiKey: "AIzaSyDRs4q6iC5ZC-2ByTcM-TaAmT_iEixYcKA",
4    authDomain: "database-651ea.firebaseio.com",
5    databaseURL: "https://database-651ea.firebaseio.com",
6    projectId: "database-651ea",
7    storageBucket: "database-651ea.appspot.com",
8    messagingSenderId: "302308004819",
9    appId: "1:302308004819:web:eae8b75637681777da1526",
10   measurementId: "G-7HZR15SRXJ"
11 };
12 // Initialize Firebase
13 firebase.initializeApp(firebaseConfig);
14
```

A JavaScript függvények és azok használati módja:

- CustomAlert: Saját alert létrehozására szolgál, ezt html tag-ekkel oldja meg.
- add_todo: Ellenőrzi a beírt, megadott adat helyességét, majd ha megfelel, hozzá adja az adatbázishoz és kiírja a képernyőre.
- create_unfinished_ToDo: Betöltődik, amint elindul a weboldal. Először kitörli a To Do List és az Expired List tartalmát, majd lekéri az unfinished_ToDo adatait az adatbázisból, azt berakja egy tömbbe és kiírja a képernyőre. Ha az adat régebbi az aktuális időnél, akkor az Expired List-be rakja, ellenben pedig a To Do List-be.
- create_finished_ToDo: Betöltődik, amint elindul a weboldal. Először kitörli a Finished List tartalmát, majd lekéri a finished_ToDo adatait az adatbázisból, azt berakja egy tömbbe és minden adatot kiír a képernyőre.
- todo_done: Valamelyik eseménynél a plusszra kattintva hívódik meg. Az adott eseményt kitörli a To Do List-ről és áthelyezi a Finished listába. Mindezt a képernyőn és az adatbázisban is egyaránt.
- todo_minus: Valamelyik eseménynél a minusszra kattintva hívódik meg. Az adott eseményt kitörli a Finished listából és áthelyezi a To Do List-be. Mindezt a képernyőn és az adatbázisban is egyaránt.
- todo_edit: Valamelyik eseménynél a ceruzára kattintva hívódik meg. Ha szerkeszteni akarunk egy eseményt, akkor hívódik meg. Engedélyezi a szerkesztést.
- finish_edit: Ha befejeződött a szerkesztés, akkor lezárja azt és frissíti az adatbázist is.

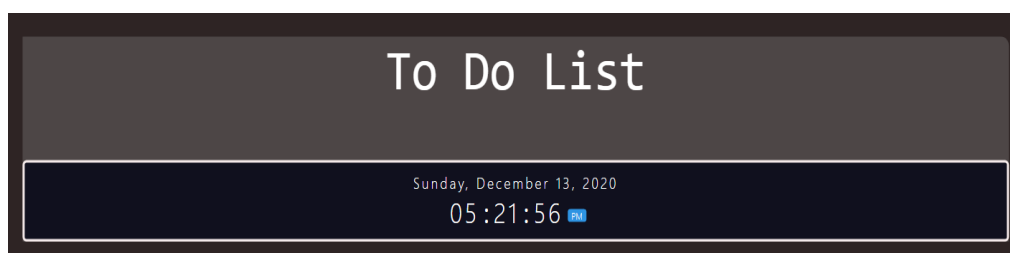
- `todo_delete`: Valamelyik eseménynél a szemetes kukára kattintva hívódik meg. Töröl az adatbázisból és a képernyőről is egyaránt.
- `updateClock`: Frissíti az aktuális órát folyamatosan.
- `initClock`: Betöltődik, amint elindul a weboldal. Elindítja az aktuális órát és kiírja azt a képernyőre.

A html rész felépítése a következő képen történik:

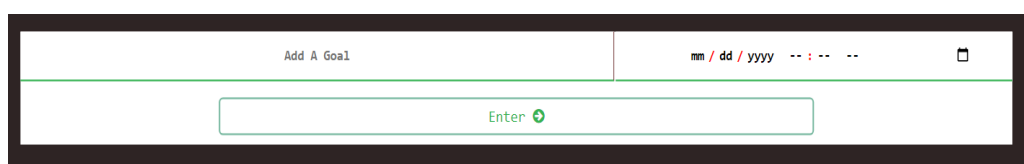
- `head`:
 - ♥ Első rész: cím, szerző, oldal újratöltése
 - ♥ Második rész: app neve
 - ♥ Harmadik rész: icon beillesztése
 - ♥ Negyedik rész: css hozzáadása a html oldalhoz
 - ♥ Ötödik rész: javascript hozzáadása a html oldalhoz

Használt elemek: `meta`, `link`/`style` és `script`

- `body`:
 - ♥ Első rész: cím, illetve aktuális időt.



- ♥ Második rész: adat feltöltési lehetőségek.



♥ Harmadik rész: A To Do List.

| To Do | | |
|---------|------------------|-------|
| Picnic | 2020-12-15 00:17 | + ✎ 🗑 |
| Gym | 2020-12-13 22:30 | + ✎ 🗑 |
| Go home | 2020-12-19 15:20 | + ✎ 🗑 |
| Meeting | 2020-12-18 20:45 | + ✎ 🗑 |

♥ Negyedik rész: A Finished.

| Finished | | |
|------------|------------------|-----|
| Gym | 2020-12-11 20:10 | - 🗑 |
| Gym | 2020-12-21 10:06 | - 🗑 |
| Party | 2020-12-24 18:00 | - 🗑 |
| Relax time | 2020-12-08 10:57 | - 🗑 |

♥ Ötödik rész: Az Expired List.

| Expired | | |
|-----------|------------------|-----|
| Coding | 2020-12-01 16:52 | + 🗑 |
| Call mome | 2020-12-03 21:00 | + 🗑 |
| Cook | 2020-12-12 12:15 | + 🗑 |
| Gym | 2020-12-03 07:55 | + 🗑 |

♥ Hatodik rész: Copyright és évszám mutatása JavaScript kódot használva.

Copyright © 2022, Software rendszerek tervezése

Használt elemek: onload, header, main, div, p, span, input (text és datetime-local), button, és footer.

A css rész felépítése:

- . direkt hivatkozások
- i. id-k
- ii. class-ek
- iii. felülírás: calendar, scrollbar

Miután a weboldal teljesen kész közzé teszem az interneten, de ingyenes domaint csak úgy szerezhetek, ha egy platformot használok, ami ingyenes domaint, illetve tárhelyet biztosít. Hátulütője annyi, hogy az url címet nem teljesen szekeszthetem, hozzá teszi a maga "reklámját".


A használt platform: <https://www.netlify.com/>


Használt ikonok és képek:

- to-do-list.png
- valamint egy nyílt forrású kód ikonjai
(<https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css>):


♥ fa fa-arrow-circle-right 

♥ fa fa-plus +

♥ fa fa-pencil 

♥ fa fa-trash 

♥ fa fa-minus -

♥ fa fa-check 

b. A Desktop applikáció megvalósítása:

A Desktop appot Visual Studio-ban valósítottam meg, és C# programozási nyelvet használtam. Létrehoztam egy Windows Forms App-ot, aminek a segítségével a könnyen létrehozhattam a program UI részét.



Az app értesít, ha van aktuális aznapi tevékenység. Az oldat újra lehet tölteni egy gomb segítségével, és ha még mindig van aktuális tevékenység, újra értesít küld.

Ahhoz hogy Firebase adatbázist tudjak használni, külön telepíteni kellett a “FireSharp” NuGet-et, ami egy kiegészítő a programhoz. A Firebase használata a desktop appban, az adatbázishoz való kapcsolódási C# kód:

```

21
22     IFirebaseConfig config = new FirebaseConfig()
23     {
24         AuthSecret = "R9pEe3HX2b1Lh0BmHBSQrfqzApclaLX02fE091py",
25         BasePath = "https://database-651ea.firebaseio.com/"
26     };
27
28
29     2 references
30     private async void Form1_Load(object sender, EventArgs e)
31     {
32         try
33         {
34             client = new FirebaseClient(config);
35
36             export();
37         }
38         catch
39         {
40             MessageBox.Show("There was a problem in your internet.");
41         }
42     }
43

```

Függvények és azok használati módja:

- Form1_Load: betölti az adatbázist és hibát ad, ha helytelen a betöltés.
- export: Lekéri az unfinished_ToDo adatait az adatbázisból, azt kiírja a képernyőre. Ha az adat régebbi az aktuális időnél, akkor az Expired Listbe rakja, ellenben pedig a To Do Listbe. Majd lekéri a finished_ToDo tartalmát is az adatbázisból és szintén megjeleníti azt. Egy string-ben elhelyezi az aktuális napi tevékenységeket és megjeleníti azokat.
- Add_Click: Ellenőrzi a beírt, megadott adat helyességét, majd ha megfelel, hozzá adja az adatbázishoz és kiírja a képernyőre.
- get_unique_string: Ezen függvény segítségével tudok generálni egy adott hosszú random kódot, amit majd a Key-nek használhatók.
- refresh_Click: Az appot újratölti. Segítségével a változtatások és a törölt elemek frissülnek.
- addItem: Az új elemek hozzáadását vezérli, hogy a megfelelő panelon jelenjenek meg és itt hívódik meg a usercontrol is.

A UserControl egy különálló C# fájl, ami különböző függvényeket tartalmaz és különálló a UI-ja is. Ezeket a UserControl-okat könnyedén

behelyezhetjük a fő programba. A UserControl neve ToDoItem és ennek a függvényei a következők:

- ToDoItem: Egy adott adatbázis elem megjelenítése és elhelyezése változóknak.
- trash_Click: Adott elem törlése az adatbázisból és a képernyőn való törlés megjelenítése.



- pipe_Click: Ha az adott ToDo teljesítve van, a gomb segítségével áthelyezhetjük a teljesített listába.
- minus_Click: Ha azonban az adott ToDo még nincs teljesítve, a gomb segítségével visszahelyezhetjük a nem teljesített listába.
- pencil_Click: Adott elemet lehet szerkeszteni az adatbázisban és a képernyőn is egyaránt.

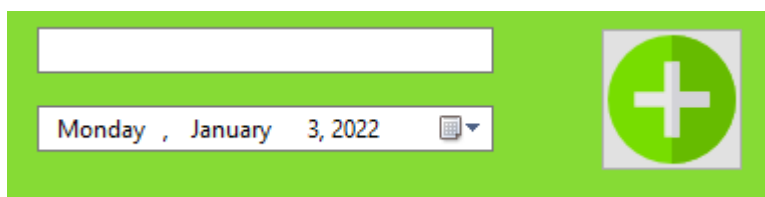
Továbbá használtam egy class-et is, aminek 3 eleme van: Title, Date, Key. Ezeknek külön külön get és set függvénnyel rendelkeznek.

A UI felépítése a következő képpen történik:

Újratöltés gomb:



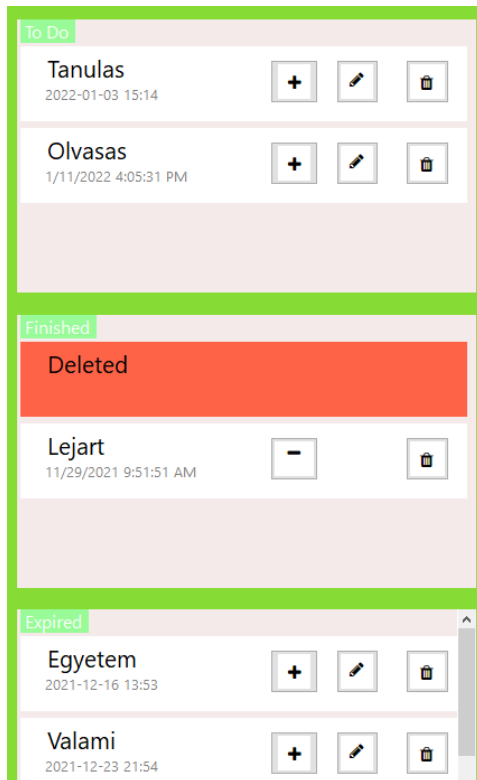
Adat hozzáadási rész:



Egy UserControl kinézete:



A ToDo, a Finished és az Expired rész, valamint azok tartalmai:



Használt ikonok és képek:

- to-do-list-min.ico
- add.png
- arrow.png
- minus.png
- pencil.png
- plus.png
- refresh.png
- trash.png

c. A Firebase adatbázis elkészítése, és azzal kapcsolatos feladatok leírása:

A weboldalon és a Desktop appan egyaránt használt közös adatbázis:
Firebase <https://firebase.google.com/>

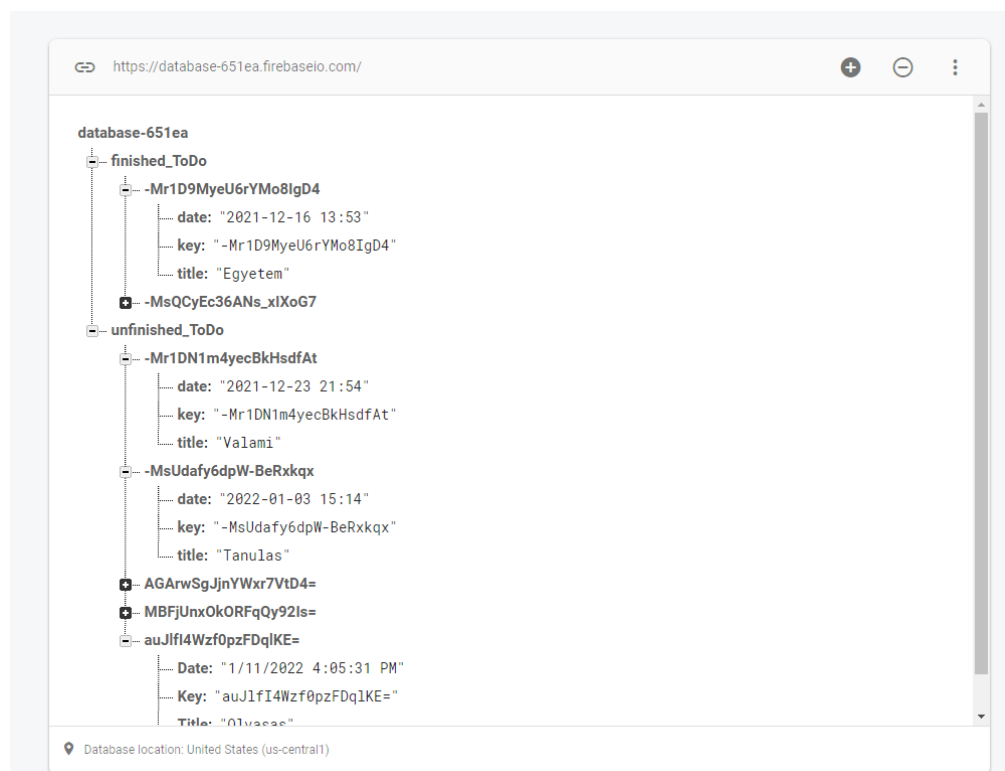


App name: ToDoList

Sorok: unfinished_ToDo, finished_ToDO

Sorok adatai: date, key, title

Adatbázis kinézete



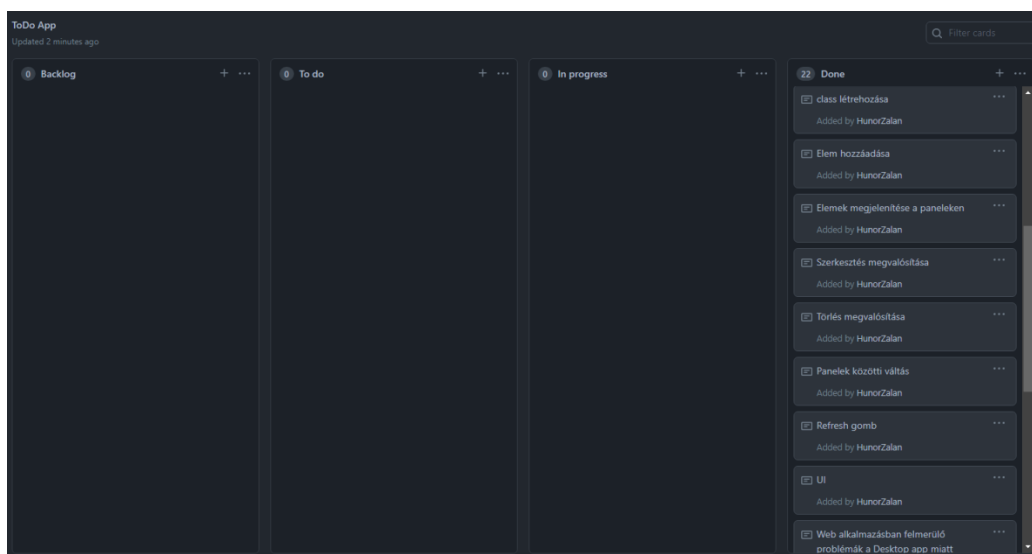
d. Verziókövetés:

Napjainkban a verziókövetés igen fontos egy informatikai projekt megalkotása közben. A projekt nem csak egy forráskódból áll, de ez nem okoz problémát. Jómagam GIT-et használtam és a SourceTree programot.

<https://github.com/HunorZalan/TODO-app2.0>



Továbbá project managment board-ot is használtam, a GitHub által. Kezdetben meghatároztam a „ToDo” oszlopot, ahova sorokat raktam be, és funkcionalitásokat, valamint szükséges lépéseket fogalmaztam meg beléjük. Ha egy soron dolgoztam már, akkor az „InProgress” oszlopba került. Ha pedig befejeződött, akkor a „Done”-ba. A „Backlog” oszlop a háttéranyagokat tartalmazta.



e. Design

A design rész css(web), illetve cs(C#) fileokban történik.

a. színek:

- háttér: #86db35, #4d4646, #f5eaea, #4d4646, #10101E
- betű: #fff, #000, grey
- gomb: #36b353, #75b79e, #ffed83, #c42a2a, #4d4646

b. betűtípusok: monospace, Segoe UI

c. gombok: 5 különböző + 1 a Desktop App-ban

d. képek: to-do-list.png → A kép az applikáció, illetve a weboldal icon-ja.

6. Továbbfejlesztési lehetőségek

A web és Desktop alkalmazás elnyerte tetszését azon embereknek, akik megtekintették és kipróbálták. Viszont vannak még lehetőségek, amelyekkel élvezetesebbé, hasznosabbá és optimálisabbá lehet tenni az appot, illetve weboldalt.

Elsősorban ez egy demo verzió, ami azt tükrözi, hogy lennének még megvalósítható ötletek és fejlesztési lehetőségek. Ami egyből szembe tűnő, hogy egyetlen egy személyre szabott. Tehát ki lehetne fejleszteni úgy, hogy legyen egy bejelentkezés és minden felhasználónak külön saját feladatlistája. A felhasználóknak külön kód, melyet akár szerkeszteni is tudnak.

Másodsorban azt tudjuk elmondani, hogy hozzá lehetne adni egyéb olyan lehetőségeket, melyek még jobban megkönnyíthetik a feladatlista használatát, illetve egyéb opciókat is tartalmazhatna.

7. Összefoglalás

Az elmúlt időszakot, melyben a projekten dolgoztam, kijelenthetjük, hogy sikeresen létrejött egy működő platform. Amit egy felhasználó nyugodtan tud használni. A főbb funkciók implementálva vannak, mint hozzáadás a feladatlistához, annak szerkesztése, visszavonása és törlése.

Úgy gondolom tudás szempontjából is igen hasznos volt a projekt és a végeredményre tekintve, ki tudom jelenteni azt, hogy megérte a befektetett munkát.