

**SZEGEDI SZAKKÉPZÉSI CENTRUM VASVÁRI PÁL GAZDASÁGI ÉS  
INFORMATIKAI SZAKGIMNÁZIUMA**

**54 213 05 számú Szoftverfejlesztő szakképesítés záródolgozata**

# **Iskolai digitális napló**

**Készítette:**

**Török Bendegúz Krisztián**

**Hódmezővásárhely**

**2021**

# TARTALOM

Bevezetés .....	2
1. A rendszer célja .....	3
2. A rendszerrel szemben felállított követelmények.....	3
3. A rendszer funkcionális és nem funkcionális igényei.....	3
3.1 Asztali alkalmazás funkcionális igények .....	4
3.2 Az asztali alkalmazás nem funkcionális igényei.....	4
3.3 A webalkalmazás funkcionális igényei .....	5
3.4 A webalkalmazás nem funkcionális igényei .....	5
4. A rendszer adatbázisa .....	5
4.1 Egyedek és tulajdonságaik (adattípusok): .....	6
4.2 Az adatbázis célja.....	9
5. A rendszer megvalósításának a tervezése .....	10
5.1 A Használati-Eset diagram.....	10
5.2 Folyamat programozás .....	10
5.3 Asztali alkalmazás felületének tervezése .....	11
5.4 Weboldal felületének tervezése.....	12
6. A rendszer megvalósítása .....	12
6.1 A weboldal megvalósítása.....	12
6.2 Az asztali alkalmazás megvalósítása.....	14
7. Felhasználói dokumentáció.....	16
7.1 Weboldal használata.....	16
7.2 Asztali alkalmazás használata .....	18
8. Összegzés.....	27
Irodalom jegyzék .....	29
Mellékletek .....	30

## BEVEZETÉS

A záródolgozat fő témája egy digitális iskolai napló és a program illetve a weboldal a Tábla nevet kapta. A feladat során szem előtt kellett tartani a követelményeket, miszerint mint a program mint a weboldal képes legyen adatbázishoz hozzáférni, onnan adatokat lekérni, törölni illetve módosítani. Mindazonáltal további követelmény a webes felületre, hogy az oldal illetve oldalak legyenek reszponzívok, jól átláthatóak és megfelelően megszerkesztettek. A profil jogosultságokra is külön figyelmet kell szentelni, tehát szerkeszthető legyen az asztali alkalmazásban és a webes felületen is a megfelelő jogosultsággal beállított felhasználók számára. Az asztali alkalmazás létrehozásakor külön figyelmet fordítottam arra, hogy megfelelően fusson a megadott operációs rendszereken és, hogy a program indulásakor autentikációt végezzen illetve a program formailag is megfeleljen a követelményeknek, azaz legyen áttekinthető és egységes kinézetű.

A témaválasztásom fő indoka az volt, hogy a fentebb leírt követelményeket megtudjam valósítani. A témaválasztásomat az indokolja még, hogy elég sok alkalommal használunk hasonló digitális naplókat a tanulmányaink során és egy remek lehetőségnek ígérkezett a korábban ismertetett problémák megoldására. Úgy gondolom, hogy egy e-napló keretein belül remekül lehet az adatbázis használatát szemléltetni, az adatok lekérdezésére, módosítására és törlésére is kiváló példa lehet. A témaválasztásom során szem előtt tartottam, hogy megfeleljen mindenféle elvárásnak így esett a választásom egy digitális iskolai napló elkészítésére.

Az asztali alkalmazás főleg funkcionál egyfajta osztályozó alkalmazásként, ahonnan a „Tanár” illetve „Osztályfőnök” jogosultsággal rendelkező felhasználók tudják módosítani az adatbázis tartalmát a hozzájuk tartozó „Osztályok” számára. A webes felületen pedig remekül lehet szemléltetni az adatbázisból lekért adatok elhelyezését egy weboldalon. A weboldal továbbá bemutatja, hogy milyen irányba halad a társadalmunk a 21. században, ezért szükséges a reszponzivitás, hiszen manapság főleg mobil azaz hordozható eszközökről használunk webes alkalmazásokat. Mindazonáltal a választás során megfelelő hangsúlyt fektettem az oldal megjelenítésére, hogy elég modernitást sugalljon. Illetve a megírt metódusokat igyekeztem úgy kialakítani, hogy a webhely használata során az adatok leggyorsabban jelenjenek meg illetve megfelelő gyorsasággal töltődjenek fel az adatok az adatbázisba.

## **1. A RENDSZER CÉLJA**

A záródolgozatomként elkészített asztali alkalmazás és webalkalmazás fő célja az, hogy szemléltessem az adatbázissal kapcsolatos műveleteket, mint egy számítógépes programban, mint egy weboldal létrehozásával.

Egy másik fő célja a rendszernek, hogy megkönnyítse a tanárok, diákok és szülők mindennapjait azzal, hogy egy könnyen elérhető, könnyen használható azaz felhasználó barát rendszert kapnak. Mindemellett nagyon fontos, hogy az asztali alkalmazás elkészítése közben igyekeztem, hogy a program ne csak a kezelésében legyen felhasználóbarát, hanem abban is, hogy szinte bármely erősségű Windows operációs rendszerrel ellátott számítógépről futtatni lehessen.

A könnyű és gyors használhatóságtól eltekintve, a célom az volt a korábban említetteken kívül, hogy teljes mértékben kiválthassák az iskolák a papír alapú naplózást, mind a hiányzások, mind az osztályzásokat tekintve.

## **2. A RENDSZERREL SZEMBEN FELÁLLÍTOTT KÖVETELMÉNYEK**

A rendszerrel az elkészítése előtt megfogalmaztam néhány alapvető elvárás, követelményt. Ezeket az elvárásokat szeretném most ismertetni egy lista segítségével:

- A rendszer legyen stabil
- A rendszer legyen könnyen használható
- Felhasználóbarát legyen a megvalósítás
- Gyors memóriakezelés
- Átlátható rendszerezés a megjelenítésben

Ezek voltak a fő követelmények, amelyre nagy figyelmet fordítottam, voltak kisebb elvárások, mint például, hogy megfelelő dizájnnal készüljön el a web- és asztali alkalmazás is.

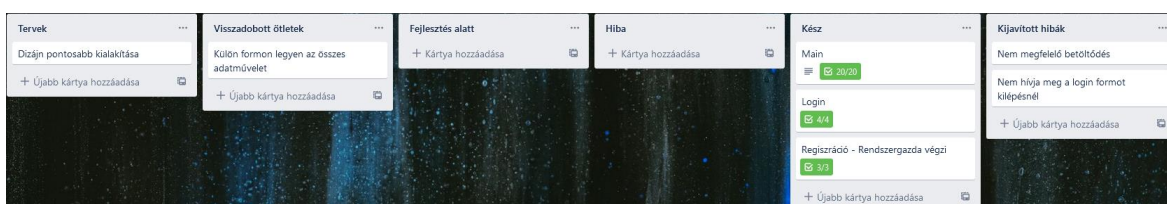
## **3. A RENDSZER FUNKCIONÁLIS ÉS NEM FUNKCIONÁLIS IGÉNYEI**

A rendszer fejlesztése közben felmerültek funkcionális és nem funkcionális igények, mint az asztali alkalmazásnál, mint a weboldal terén. Ezeket az igényeket szeretném kifejezni. A

funkcionális igények követéséhez használtam a [www.trello.com](http://www.trello.com) nevezetű oldalt ami egy webalkalmazás ahol bármilyen projektet lehet menedzselni bizonyos kártyák segítségével. Általában az oldal használata akkor ajánlott ha több ember dolgozik a közös projekten, ugyanakkor jelen esetben kiválóan tudtam hasznosítani az oldal által kínált lehetőségeket.

### 3.1 Asztali alkalmazás funkcionális igények

- Az alkalmazás indulásakor autentikációt végezzen.
- A beléptetést követően jelenjen meg a tanári vagy rendszergazdai felület a jogosultságoknak megfelelően.
- Lehetséges legyen az adatbázishoz adatot hozzáadni
- Az adatbázisból lehessen adatokat megjeleníteni, lekérni
- Az adatbázisban lévő adatokat lehessen módosítani
- A rendszergazda felületen táblázatban jelenjenek meg a tanárok, diákok és osztályok.
- A rendszergazdai felületen lehessen felhasználókat létrehozni a diákokhoz, tanárokhoz és rendszergazdáknak.



1. ábra: A webalkalmazás funkcionális igényeinek nyomonkövetésére használt trello tábla.

### 3.2 Az asztali alkalmazás nem funkcionális igényei

Az alkalmazás fejlesztése előtt nem csak funkcionális hanem nem funkcionális igények is felmerültek. Ilyen igények voltak például, hogy átlátható és könnyen kezelhető felület hozzak létre. Ezen felül a jogosultsági hozzáférés is a nem funkcionális igények közé tartozott. Ezért hoztam létre a fejlesztés során három jogosultsági szintet amik fejlesztésre kerültek, ezek a szintek: Diák, Tanár és Rendszergazda. Az asztali alkalmazásban ténylegesen csak két jogosultsági szintet használok ami a rendszergazda aki képes a legtöbb műveletre, mint például osztályok létrehozása, diákokat és tanárokat osztályokhoz rendelni, a felhasználók módosítására és így tovább még néhány fontos funkció keresztül. A tanárok viszont nem képesek csak osztályzatok és hiányzásokat könyvelni a diákok számára.

### 3.3 A webalkalmazás funkcionális igényei

- A weboldal betöltésekor csak bejelentkezés után férjen hozzá a felhasználó a tartalomhoz.
- A felhasználó csak a saját adatait legyen képes lekérni.
- Legyen lehetősége a felhasználónak, jelen esetben diáknak üzenetet küldeni a tanárok számára.
- A diák képes legyen a hiányzásait táblázatosan megtekinteni.
- A diáknak legyen lehetősége az osztályzatait megtekinteni a beírás időpontja alapján.

### 3.4 A webalkalmazás nem funkcionális igényei

A webalkalmazásnál szem előtt kellett tartani a letisztult és átlátható dizájnt, amire igyekeztem hangsúlyt fordítani ugyanakkor sajnos figyelembe kell venni, hogy jómagam nem vagyok valami túl kreatív ember és eléggé alacsony a szépérzőm, ettől eltekintve törekedtem, hogy a felhasználó átláthassa a weboldalt. Továbbá a weboldalon is legyen autentikáció csak úgy, mint az asztali alkalmazásban. Ezt az igényt is sikerült teljesítenem, ugyanis egyelőre csak diák felhasználók léphetnek be a weboldalra. Előre bocsájtanám, hogy további fejlesztést igényelhet a weboldal, ezeket illetően már vannak is tervek amelyek megvalósításra fognak kerülni.

## 4. A RENDSZER ADATBÁZISA

### 4.1 Tervezés

**Egvedek:** diákok, hiányzasok, jegyek, login, osztályok, tanárok, tanítja, tantargyak, üzenetek

**Tulajdonságok:**

Diákok (**ID**, Nev, **OsztalyID**, logged, regisztralt)

Hiányzasok (**ID**, Datum, **TanarID**)

Jegyek (**ID**, Datum, **TantargyID**, **TanarID**, **DiakID**, Jegy)

Login (**ID**, **DiakID**, **TanarID**, RendszergazdaE, username, password)

Osztalyok (**ID**, OsztalyNev, **TanarID**)

Tanarok (**ID**, Nev, OsztalyfonokE, logged, regisztralt)

Tanitja (**ID**, **TantargyID**, **TanarID**) - kapcsolótábla

Tantargyak (**ID**, Nev)

Uzenetek (**ID**, **DiakID**, TanarNev, Uzenet)

#### 4.1 Egyedek és tulajdonságaik (adattípusok):

##### *Diakok*

---

**ID** szám (INT) – AUTO\_INCREMENT – elsődleges kulcs

Nev szöveg (VARCHAR(50)) – a diák neve

logged szám (tinyint) – az adott diák bejelentkezési státusza

regisztralt szám (tinyint) – az adott diák regisztrált-e

##### *Hianyzasok*

---

**ID** szám (INT) – AUTO\_INCREMENT – elsődleges kulcs

Datum dátum(date) – Rögzítés dátuma

TanarID szám(INT) – idegen kulcs, a rögzítést végző tanár ID-ja.

##### *Jegyek*

---

**ID** szám (INT) – AUTO\_INCREMENT – elsődleges kulcs

Datum dátum(date) – Rögzítés dátuma

TantargyID szám(INT) – idegen kulcs, a tantárgy amiből kapta a jegyet a diák

TanarID szám(INT) – idegen kulcs, a rögzítést végző tanár ID-ja.

Jegy szám (INT) – a kapott jegy

##### *Login*

---

**ID** szám (INT) – AUTO\_INCREMENT – elsődleges kulcs

DiakID    szám(INT) – idegen kulcs, a tantárgy amiből kapta a jegyet a diák

TanarID    szám(INT) – idegen kulcs, a rögzítést végző tanár ID-ja.

RendszergazdaE    szám (tinyint) – az adott felhasználó rendszergazda-e

username    szöveg (VARCHAR(50)) – felhasználónév

password    szöveg (VARCHAR(50)) – jelszó

### *Osztalyok*

---

**ID**    szám (INT) – AUTO\_INCREMENT – elsődleges kulcs

OsztalyNev    szöveg (VARCHAR(50)) – az osztály neve

TanarID    szám(INT) – idegen kulcs

### *Tanarok*

---

**ID**    szám (INT) – AUTO\_INCREMENT – elsődleges kulcs

Nev    szöveg (VARCHAR(50)) – az osztály neve

OsztalyfonokE    szám (tinyint) – az adott tanár osztályfőnök-e

logged    szám (tinyint) – az adott tanár bejelentkezési státusza

regisztralt    szám (tinyint) – az adott tanár regisztrált-e

### *Tanitja*

---

**ID**    szám (INT) – AUTO\_INCREMENT – elsődleges kulcs

TantargyID    szám(INT) – idegen kulcs

TanarID    szám(INT) – idegen kulcs

### *Tanitja*

---

**ID**    szám (INT) – AUTO\_INCREMENT – elsődleges kulcs

Nev    szöveg(VARCHAR(50)) – a tantárgy neve



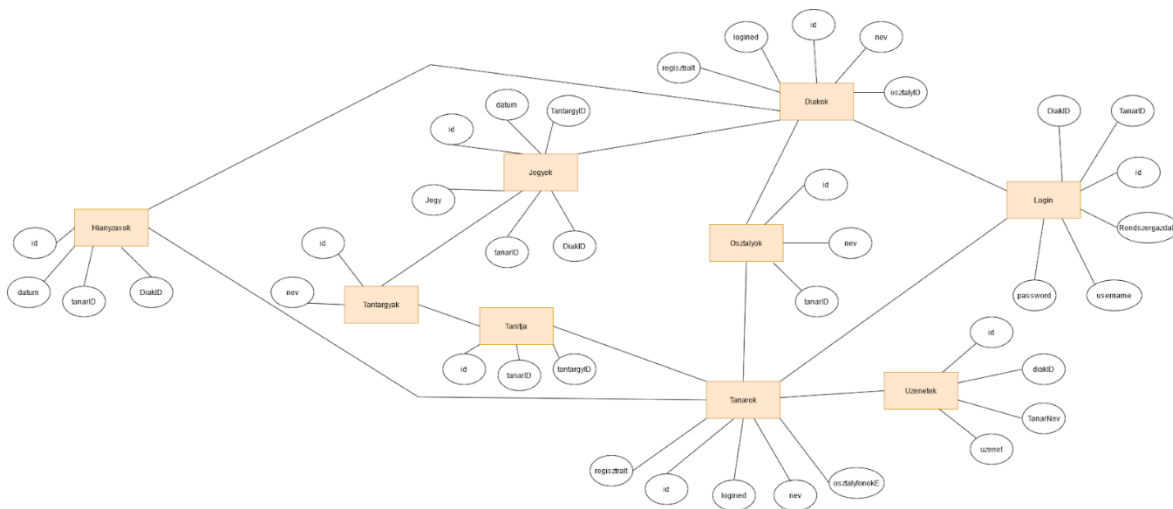
Uzenetek (**ID**, **DiakID**, TanarNev, Uzenet)

**ID** szám (INT) – AUTO\_INCREMENT – elsődleges kulcs

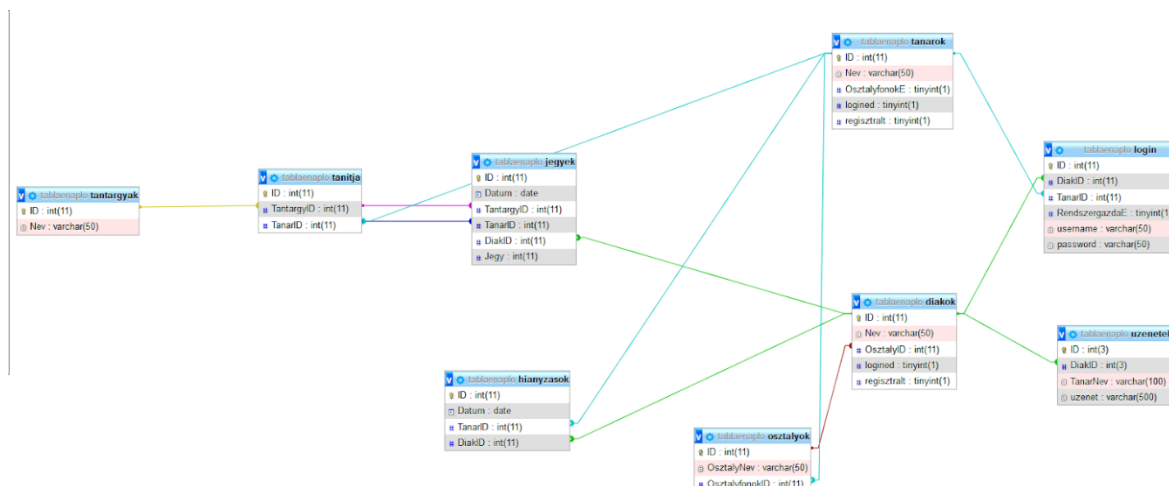
DiakID szám(INT) – idegen kulcs

TanarNev szöveg(VARCHAR(100)) – az üzenetet fogadó tanár neve

Uzenet szöveg(VARCHAR(500)) – az üzenet



2. ábra: Az adatbázis E-K diagramja



3. ábra: Az adatbázis Bachmann-ábrája

## 4.2 Az adatbázis célja

Az adatbázis egy digitális napló azaz egy e-naplóban végzett tevékenységek, mint például a jegyek beírása, a hiányzások könyvelése, az osztályok létrejötte, törlődése, átnevezése, továbbá diákok és tanárok osztályokhoz való rendelését hivatott tárolni. Ezen felül a diákok és tanárok bejelentkezési lehetőségeit tárolja az adatbázis, a jelszavakat titkosított formában. A tanár jogosultsággal ellátott személyek képesek a korábban kifejtett hiányzásokkal és jegyekkel kapcsolatos műveletek elvégzésére. A rendszergazda jogosultsággal ellátott tanárok pedig képesek osztályok létrehozására, osztályokhoz rendelni tanulókat és tanárokat. A felhasználók adatbázisba történő rögzítését azaz a regisztrációt szintén a rendszergazda valósítja meg, az asztali alkalmazáson keresztül. Az adatbázisban lévő adatokat, főként a hiányzási és a jegyek adatait képesek a tanulók egy weboldalon keresztül megsejmlélni.

Adatbázissal kapcsolatos keresések és kimutatások

### Keresések

- Egy adott diákhoz tartozó osztályzatok lekérdezése tantárgyanként vagy dátumonként rendezve.
- Hiányzások adatainak megjelenítése egy adott diákra nézve.
- A diákok adatainak listázása egyfajta táblázatban.
- Keresés egy adott tanárra a neve alapján.
- Lehetséges keresni tanárokat a tanított tantárgyak alapján.

- Ezeken felül, még lehetséges keresést indítani az adott tanárookra az ID-juk alapján.
- A legutóbbi három keresési lehetőség a diákokra is működhet, kivétel ez alól a tantárgy, viszont azt lehet az osztállyal helyettesíteni.
- Ki lehet listázni az osztály tanulóit.

### **Kimutatások**

- Egy adott osztály létszámára lehet kimutatást készíteni
- Kimutatást lehet készíteni, hogy hány adott tanuló van az adatbázisban.
- Lehetséges kimutatást készíteni arról, hogy hány darab matek, magyar stb. tanár van az iskolában.
- Kimutatás készíthető arról is, hogy hány üzenetet váltanak a diákok a tanárokkal.

## **5. A RENDSZER MEGVALÓSÍTÁSÁNAK A TERVEZÉSE**

### **5.1 A Használati-Eset diagram**

A szoftver rendszer működését szeretném szemléltetni UML diagrammon keresztül. Az UML egy "...szabványos, általános célú modellező nyelv, üzleti elemzők, rendszertervezők, szoftvermérnökök számára." (Wikipédia, 2019). Ehhez egy Use Case Diagram-mot azaz egy Használati-Eset diagrammot használnék.(1. melléklet) A Use Case Diagram "Megadja, hogy a felhasználó mire tudja használni a rendszert. Funkcionális követelményeket ábrázoló diagram."(Dr. Szepesné Stiftinger Mária, 2010).

### **5.2 Folyamat programozás**

A folyamat programozását szeretném bemutatni egy Üzleti Szekvencia Diagramon keresztül. (2. számú melléklet. )"A szekvencia diagram feladata objektumok egymás közti üzenetváltásainak ábrázolása egy időtengely mentén elhelyezve." (Ficsor Lajos, Krizsán Zoltán, Mileff Péter, 2001-2004).

Az ábrán jól látható, hogy egyes jogosultsággal ellátott felhasználók milyen műveleteket végezhetnek. A mellékelt ábrán látható a legtöbb művelet, ez alól kivételt képez a

rendszergazda műveletei, mivel azok kiterjednek szinte a teljes program módosítására, mint például a felhasználóknak való jogosultság beállítása illetve a Diák és Tanár jogosultsággal ellátott felhasználók osztályhoz való sorolása.

### 5.3 Asztali alkalmazás felületének tervezése

Az asztali alkalmazás felületének tervezésében segítségemre volt a paint illetve a draw.io weboldal ahol tudtam néhány tervet készíteni. Ugyanakkor ezek a tervek a megvalósítás során csak támpontként szolgáltak, hiszen mint tudjuk gyakran a terv nem egyezik meg a megvalósítással. Ugyanakkor a felület tervezése közben tudatosult bennem, hogy az asztali alkalmazásban szükségem lesz több Form-ra és használni szeretném a TabView-t az alkalmazásban. Az alkalmazás felületének kialakítása közben figyeltem, hogy alapvetően felhasználó barát legyen ugyanakkor a törekedtem, hogy letisztult legyen a megjelenítés. Úgy gondolom, hogy azokat az alap célokat amiket kitűztem magam elé sikerült teljesítenem. Továbbá a tervezésnél használtam, mint már említettem a trello.com weboldalt, ahol egyfajta listaként jelenítettem meg, hogy milyen kinézeti elvárásaim vannak.



4. ábra: A program főoldalának az alap tervezete a trello.com-on.

## **5.4 Weboldal felületének tervezése**

A weboldal tervezése közben igyekeztem egyfajta letisztult megjelenést kialakítani. Továbbá mivel a rendszer főleg az asztali alkalmazásra épül, ezért a weboldal megjelenése további fejlesztést még igényel. A weboldal megtervezéséhez a korábban említett draw.io weboldalt használtam, viszont itt is csak egy kezdetleges képet tudtam kialakítani, hogy hogyan is szeretném megvalósítani az oldalt. Ezt a harmadik mellékleten lehet megtekinteni. Későbbiekben ezt a tervet áthelyeztem egy későbbi megvalósítás formátumba, helyette került megvalósításra, hogy a diákok üzenetet tudjanak küldeni a kiválasztott tanár(ok)nak.

## **6. A RENDSZER MEGVALÓSÍTÁSA**

### **6.1 A weboldal megvalósítása**

A weboldal megvalósításához a Microsoft Visual Studio Code 1.55-ös verziószámú fejlesztői környezetet és a XAMPP nevű programot használtam fel. Azért esett a választásom a VS Code-ra, hiszen remek kiegészítő lehetőségei vannak a szoftvernek főleg a webfejlesztés tekintetében, ilyen például a kódkiegészítés, tesztelési lehetőségek illetve egyfajta webview funkció ahol nem szükséges külön böngészőben ellenőrizni az általunk készített kódot, hanem maga a szoftverben megtudjuk jeleníteni a weboldalt. Ugyanakkor a weboldalt teszteltem a Mozilla Firefox és Google Chrome böngészőkben.

A weboldal kialakítása során igyekeztem az átláthatóság érdekében mappákba rendezni a fájlokat kiterjesztés alapján. Ez látható a 4. mellékleten.

A mellékleten látható, hogy a PHP fájlokat nem rendeztem működési szempont szerint külön mappába. Az összes PHP fájl azért található a gyökér mappában, mivel egyfajta elérési hibába ütköztem, amelyet nem sikerült egyelőre orvosolni, ezért ezt is a további fejlesztési státuszba helyeztem amelyet meg fogok valósítani további kutató munka után.

Visszont az összes stílust fájlt (CSS) külön mappába rendeztem, ezeket láthatjuk az ábra felső részén a CSS mappában. Igyekeztem a fájloknál a megnevezéseket a tanultaknak megfelelően kialakítani, hogy egyértelmű legyen melyik fájl mégis mire funkcionál.

Js nevezetű mappát létrehoztam amiben egyelőre egy darab JavaScript fájlt található, ugyanakkor ez a mappa szükséges a tovább fejleszthetőség érdekében. Szintén törekedtem a fájl illetve a későbbiekben is igyekezni fogok a fájlok megfelelő elnevezésére.

Található a mellékleten továbbá egy SQL mappa amelyben az épp aktuális adatbázist tárolom, ezt nem feltétlen szükséges elhelyezni a weboldal működéséhez ugyanakkor így könnyen elérhető ha esetleg új helyre telepítjük a rendszert.

A fejlesztés során törekedtem átlátható kódot készíteni. Törekedtem kommentekkel magyarázni, hogy melyik kód mit is végez. A kód kommentelésére nagyobb hangsúlyt kell fektetni a továbbfejlesztés során. A következő ábrán látható a korábban említett kommentelés egy töredéke.

```
<!-- Bootstrap beszúrása -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
<!-- JQuery beszúrása -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<!-- Bootstrap beszúrása ismét, nem tudom miért de szükség van rá, így működik megfelelően. -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
<!-- A CSS-ek beszúrása -->
<link rel="stylesheet" href="CSS/header.css">
<link rel="stylesheet" href="CSS/navbar.css">
<link rel="stylesheet" href="CSS/sidenav.css">
<!-- Az indexScript beszúrása -->
<script src="/Js/IndexScript.js"></script>
```

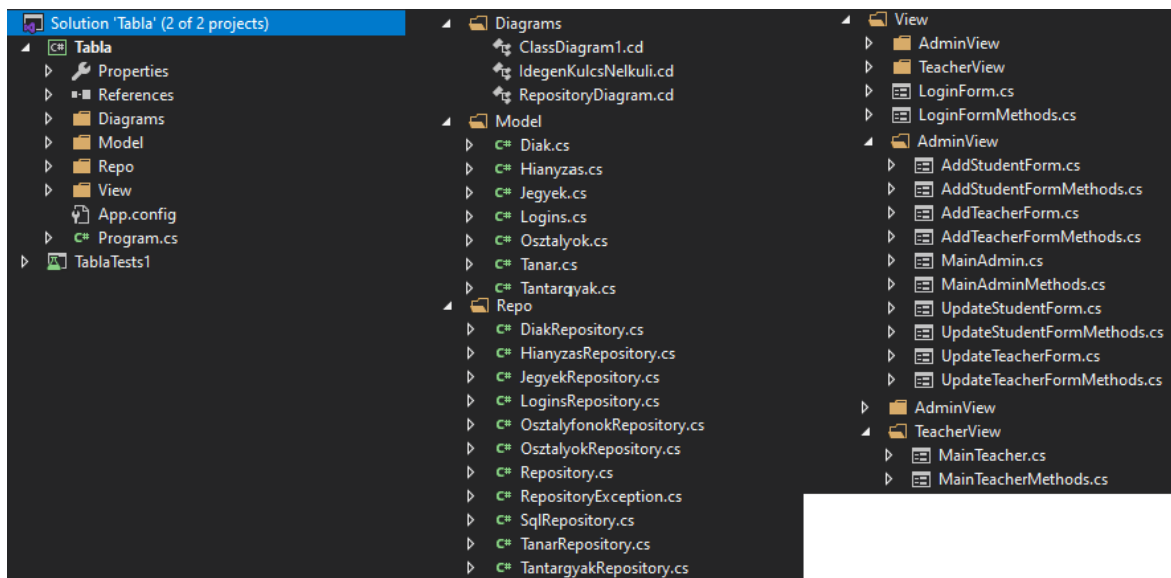
5. ábra: Kommentek a kódban, az átláthatóság és egyértelműség értelmében.

Úgy gondolom, hogy a megfelelő kommentelés a továbbfejlesztést könnyedé teheti. Továbbá a legtöbb IDE-be beépített billentyűkombinációval könnyedén lehet kommenteket létrehozni, ezért is célszerű a használatuk.

## 6.2 Az asztali alkalmazás megvalósítása

Az asztali alkalmazás fejlesztése során a Microsoft Visual Studio szoftverfejlesztő környezetét használtam illetve a XAMPP szoftvert az adatbázis futtatásához. Azért esett a VS-re a választásom, hisz az általam választott C# nyelv alap IDE-je, e főértől eltekintve kifejezetten jó beépített eszközökkel rendelkezik a szoftver, például beépített UnitTest lehetőség, továbbá kódkiegészítés, ClassDiagram-ok elkészítésének a lehetősége, Form-ok drag&drop módszerű készítése. Ezeket a lehetőségeket ki is használtam a fejlesztés során, viszont rengeteg egyéb lehetőség rejlik a szoftverben. Csak, hogy néhány olyan példát is említsek amit nem használtam ki: Adatbázis létrehozása, több féle modul használata.

Az asztali alkalmazás fejlesztése során hasonló rendezést használtam a fájlknál, mint a webfejlesztés során. A fő különbség az volt, hogy a fejlesztés közben szem előtt tartottam az MVC-t azaz a Model View Controller-t ami a szoftvertervezésben használatos programtervezési minta. Majdnem teljesen az MVC alapelvei szerint kerültek elrendezésre a fájlok, ugyanakkor más elnevezéssel. Ez látható a következő ábrán.



6. ábra: Az asztali alkalmazás fájljainak elrendezése.

Ahogy az ábrán is látható négy fő mappa található a projektben. Az első fő mappa a Diagrams ami tárolja a VS-ben elkészített diagrammokat.

A következő mappa a Model mappa, ami maga a model réteg ahol az objektumokat tárolom. Ezek az objektumok a Diak, Hianyzas, Jegyek, Logins, Osztalyok, Tanar és a Tantargyak.

A következő rétegünk lenne a controller réteg, amit kiváltottam a Repository réteggel. Ebben a rétegben találhatóak a legtöbb objektum azon osztálya ami tárolja a metódusokat és ezekben az osztályokban hajtódnak végre az objektumok metódusai. A fő osztály a Repository mappán belül a Repository ami egy objektum, ebből öröklődik az összes többi osztály. Maga a Repository objektum egy üres osztály, melynek külön tulajdonságai nincsenek, ugyanakkor egy üres konstruktorral rendelkezik, hogy példányosítani lehessen a Repository-t ezáltal az összes örökös osztályt metódusait fellehessen használni.

```
1  using System;
2      using System.Collections.Generic;
3      using System.Linq;
4      using System.Text;
5      using System.Threading.Tasks;
6
7  namespace Tabla.Repo
8  {
9      20 references
10     public partial class Repository
11     {
12         11 references
13         public Repository()
14         {
15         }
16     }
```

7. ábra: Repository objektum kódja.

Mint a hetedik ábrán látható maga a Repository-t 11 helyen példányosítottam. Továbbá 20 helyen lett hivatkozva az objektumra.

A fejlesztés során szintén törekedtem a kódot kommentelni ezzel is magyarázatot adni, hogy az adott kódrész vagy épp metódus mire szolgál. A legtöbb megírt metódus előtt egy fajtam summázás található, amiben részletezve van, hogy a metódus melyik paramétere mi célt szolgál illetve maga a metódus mit is csinál. Ez látható a nyolcadik ábrán.



```

/// <summary>
/// Egy stringet készít abból, hogy mit tanítanak a tanárok.
/// </summary>
/// <param name="tanitjaLista">Egy list ami tárolja, hogy egy adott tanár mely tantárgyakat tanítja.</param>
/// <returns></returns>
1 reference
public string getSubjectsString(List<string> tanitjaLista)
{
    string subjectsString = "";
    if (tanitjaLista != null)
    {
        for (int i = 0; i < tanitjaLista.Count; i++)
        {
            subjectsString = tanitjaLista.ElementAt(i) + ", ";
        }
    }

    return subjectsString;
}

```

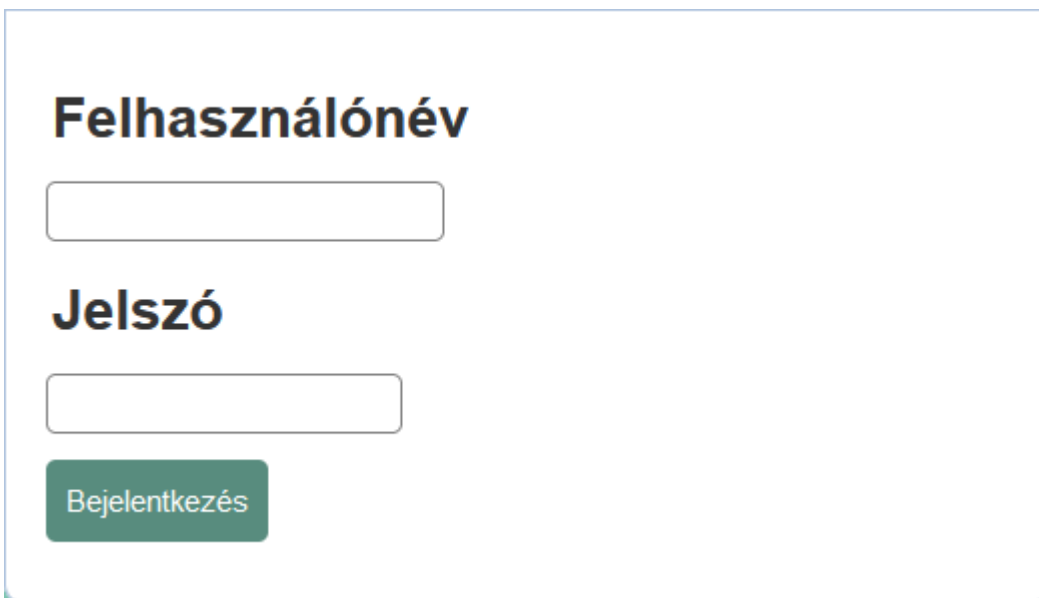
8. ábra: Összesített komment paraméterrel a kódban.

## 7. FELHASZNÁLÓI DOKUMENTÁCIÓ

Ebben a fejezetben a rendszer használatáról szeretnék szót ejteni, alaposan bárki számára érthető módon.

### 7.1 Weboldal használata

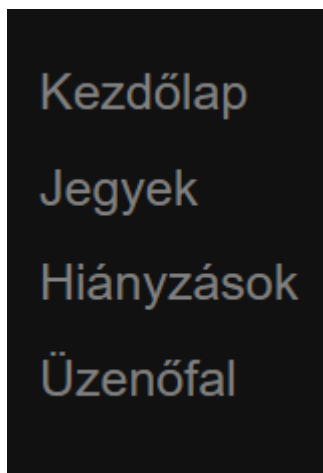
A weboldal használata egyszerű. Az oldal betöltése után a következő bejelentkezési lehetőség fogad minket.



The image shows a login interface within a light blue rounded rectangle. At the top, the text 'Felhasználónév' (Username) is displayed in a large, bold, dark blue font. Below it is a white rectangular input field with a thin grey border. Underneath the first field, the text 'Jelszó' (Password) is displayed in a large, bold, dark blue font. Below the second field is another white rectangular input field with a thin grey border. At the bottom of the panel is a green rectangular button with the text 'Bejelentkezés' (Login) in white.

9. ábra: Bejelentkezési panel.

Ezt követően a rendszergazda által adott bejelentkezési adatokkal bejelentkezünk és máris használatra kész a weboldal. Bejelentkezés után választhatunk a 10. ábrán látható menüpontok közül.



10. ábra: Weboldal menüpontjai.

A jegyek menüpontra kattintva a diákok jelen esetben a bejelentkezett diák jegyei jelennek meg egyfajta napló szerűen táblázatba rendezve. Ez látható a 11. ábrán.

Dátum	Tantárgy	Jegy
2020-09-10	Magyar nyelvtan	2
2021-01-19	Kémia	3
2021-03-02	Biológia	2
2021-03-10	Biológia	2
2021-03-11	Biológia	2
2021-03-25	Foglalkoztatás I.	2

11. ábra: A kapott jegyek rögzítés sorrendje szerint.

A táblázatban láthatjuk a jegyeinket dátum szerint sorrendbe helyezve, a táblázat második oszlopában látható, hogy mely tantárgyból kaptuk a jegyet és a táblázat utolsó oszlopában pedig a kapott jegyet láthatjuk. A későbbi fejlesztések során megjelenik egy negyedik oszlop amiben a jegyet beíró tanár neve fog megjelenni.

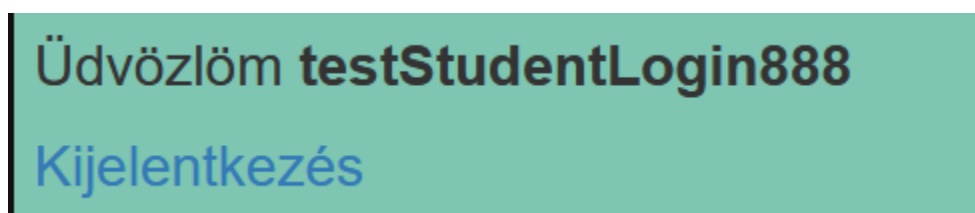
A hiányzások menüpontra kattintva egy hasonló táblázat tárul elénk, mint a jegyeknél, viszont itt csupán két oszlop található. Az első oszlop a dátum, hogy mikor hiányoztunk a második és egyben utolsó oszlopban pedig a rögzítést végző tanár neve látható. Ez a táblázat látható a 12. ábrán.

Dátum	Tanár
2021-05-11	Szabó Árpád

12. ábra: A hiányzások táblázat.

A weboldalon az utolsó menüpont amit találhatunk az az üzenőfal, mely egyelőre egyirányú kommunikációt tesz lehetővé a diák és a tanár között. Jelenleg a weboldalt használó diák tud üzenetet küldeni a kiválasztott tanár számára, melyet egy legördülő listából választhat ki. Ez az 5. mellékleten látható. Ezután már csak a szövegdobozba kell beírunk az üzenetet és a küldés gombra kattintani. Ezt követően az üzenet elküldésre kerül.

A weboldalon a főoldalra kattintva található a kijelentkezés gombja, az üdvözlő szöveg alatt.



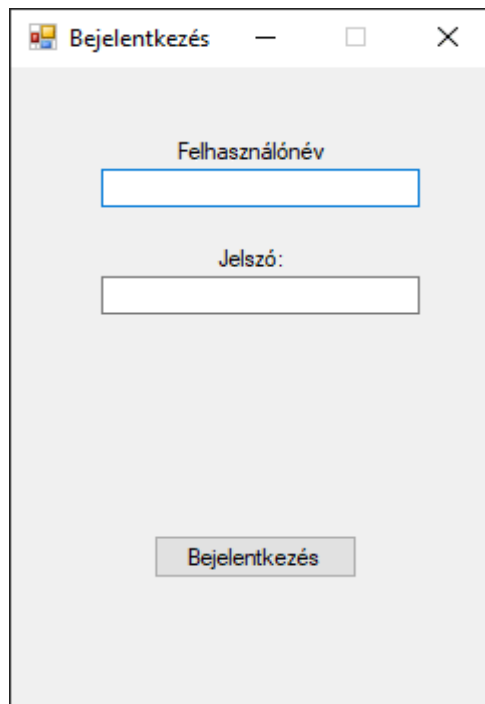
13. ábra: Kijelentkezés gombja a weboldalon.

## 7.2 Asztali alkalmazás használata

Az asztali alkalmazás futtatásához szükségünk van egy adatbázisra, amely ha minden rendben működik akkor a szerveren fut, tehát nincs vele dolgunk. Ha esetleg nem lenne a szerverünk akkor könnyen létrehozható a XAMPP program segítségével egy webszerver. A XAMPP program indítása után kattintsunk az Apache és a MySQL feliratok mellett a START nevezetű gombra. Ha ezzel kész vagyunk nyissuk meg a böngészőt és gépeljük be a következőt: `localhost/phpmyadmin`. Megjelenik egy felület a felső menüsávban az Importálás gombra kattintsunk, keressük meg a fájl tallózási lehetőséget és kattintsunk rá. Ha ezzel kész vagyunk, keressük ki a `tablaenaplo.sql` fájlt ami a program Web részének az SQL mappájában található. Ha ezzel végeztünk a program működőképes több dolgunk nincs vele.

A program futtatásához nagyon minimális gépigényre van szükségünk. 1 gigahertzes vagy gyorsabb processzorra. Körülbelül 1GB memóriára RAM-ra, Nagyjából 1GB szabad helyre. Figyelem! A program csak Windows operációs rendszeren futtatható, ajánlott legalább a Windows 7 vagy magasabb operációs rendszer. Ha ezek fenn állnak akkor sikeresen futtatható a program.

Az asztali alkalmazás indításakor egy bejelentkezési felület jelenik meg, melybe a rendszergazdától kapott bejelentkezési adatokkal léphetünk be. Szigorúan csak tanárok és rendszergazdák léphetnek be az asztali alkalmazásba ezért felesleges diák adatokkal megkísérelni a bejelentkezést.



14. ábra: Asztali alkalmazás bejelentkezési felülete.

A bejelentkezést követően két féle felület jelenhet meg.

- Rendszergazdai felület
- Tanári felület

Kezdjük a rendszergazdai felület magyarázatával.



15. ábra: Rendszergazdai felület megjelenése.

Négy úgynevezett Tab táruel élénk, melyek közül szabadon választhatunk. Az első Tab, avagy oldal a diákok menedzselésére szolgál. A jelenlegi diákok listáját a „Betölt” nevezetű gombbal jeleníthetjük meg. Az adatok egy táblázatban jelennek meg, illetve a sikeres betöltést követően megjelennek a további interakciókhoz szükséges gombok.



16. ábra: Rendszergazdai felületen az interakciós gombok.

	Azonosító	Név	Osztály név
▶	1	Kovács János	9.b
	2	Kis Pista	9.b
	3	Kerek Ferenc	9.c
*			

17. ábra: A rendszergazdai felületen a betöltött diákok listája.

A táblázat csak olvasható, beleírni nem lehet. A táblázat szigorúan megjelenítésre szolgál, további akciókat nem lehet vele végrehajtani. A 16. ábrán látható gombok közül válasszuk ki a „Keresés” nevezetűt. Ekkor megjelenik egy kereső melybe a megadott módon keresni tudunk diákokat a listából. Ezen funkció hasznossága akkor jelenik meg amikor több száz vagy több ezer diák közül szeretnénk megtalálni egyet.

Azonosító

Név

Osztály

Keresés Mégsem

18. ábra: Kereső panel az asztali alkalmazásban.

További interakcióra ad lehetőséget a „Feltölt” nevezetű gomb, mely megnyomása után egy új ablak fog megjelenni.

Tanuló hozzáadása

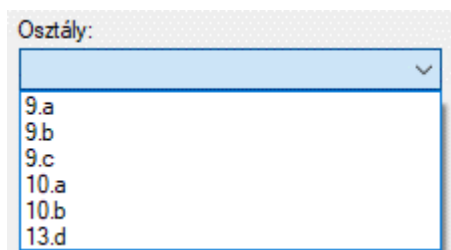
Teljes név:

Osztály:

Hozzáad Mégsem

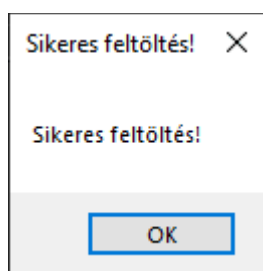
19. ábra: Tanuló hozzáadása ablak.

Rém egyszerű és felhasználó barát a megjelenítés, egyszerűen írjuk be a hozzáadni kívánt tanuló nevét a szövegdobozba illetve válasszuk ki a legördülő listából, hogy mely osztályhoz szeretnénk sorolni.



20. ábra: Tanuló hozzáadásánál az osztály kiválasztása legördülő listából.

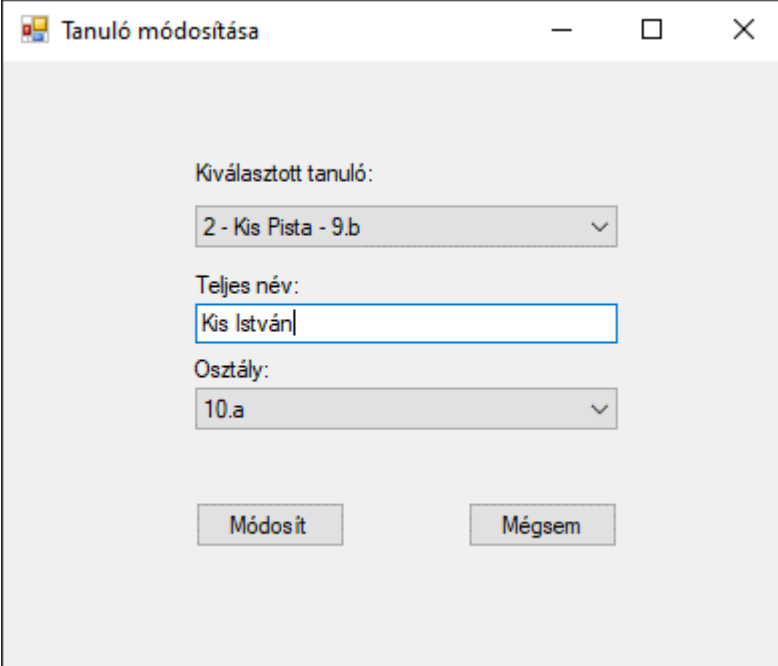
Ha kiválasztottuk az osztályt és megadtuk a tanuló nevét akkor nincs más dolgunk, mint a „Hozzáad” gombra kattintani és a rendszer értesít, hogy sikeres volt-e a feltöltés.



21. ábra: Értesítés a sikeres feltöltésről.

A következő tevékenység amelyre alkalmas a rendszergazdai felület az a felhasználó módosítása. Ha rákattintottunk a „Módosítás” gombra akkor a 22. ábrán látható kép tárulhat elénk. Először kiválasztjuk a módosítandó diákot a legördülő listából. Az első szám amit a tanuló neve előtt látunk az ID-ja, ezt követi a tanuló neve ezután pedig, hogy melyik osztály tagja.

Ha kiválasztottuk a tanulót adjuk meg az új nevet vagy írjuk be a régit és válasszuk ki az osztályt vagy adjuk meg a régi osztályát. Ezt követően kattintsunk a „Módosít” gombra. A rendszer értesítést küld egy felugró ablakban, hogy sikeres avagy sikertelen volt-e a módosítás pont, mint a 21. ábrán.



Tanuló módosítása

Kiválasztott tanuló:  
2 - Kis Pista - 9.b

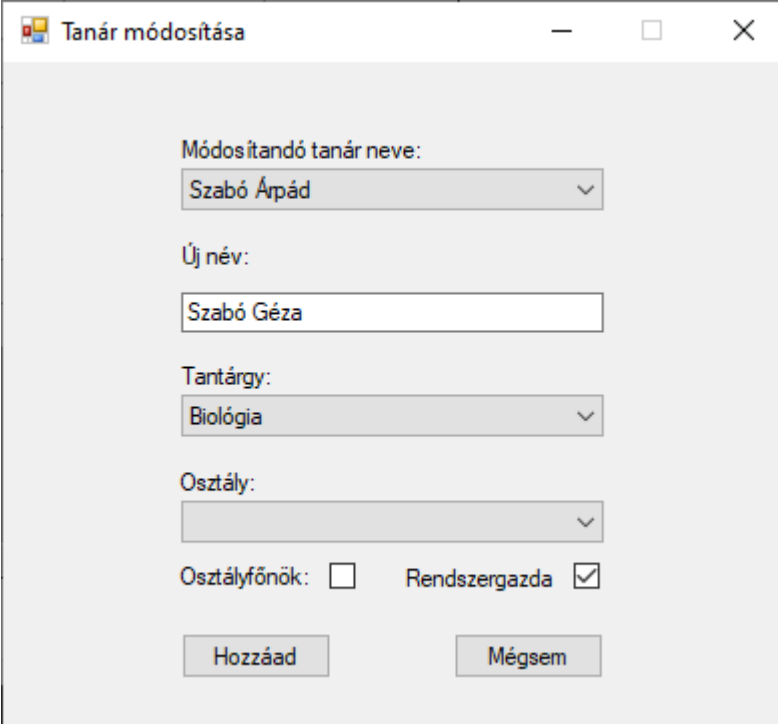
Teljes név:  
Kis István

Osztály:  
10.a

Módosít Mégsem

22. ábra: Tanuló módosítása rendszergazdai felületen.

Ezzel a Diákok füllel végeztünk is következzen a Tanárok fül. A tanárok tab nem sokban különbözik a diákokétól ugyanazok a lehetőségeink vannak, a feltöltésnél is szinte megegyezik a feltöltés menete, mint ha egy diákot töltenénk fel. Ugyanakkor a tanárok módosítása merőben eltérő egy diák módosításától. Az alap szinte megegyezik ugyanakkor néhány más adatot is szükséges megadnunk.



Tanár módosítása

Módosítandó tanár neve:  
Szabó Árpád

Új név:  
Szabó Géza

Tantárgy:  
Biológia

Osztály:

Osztályfőnök: ☐ Rendszergazda ☒

Hozzáad Mégsem

23. ábra: Tanár módosítása a rendszergazda felületen.



Ahogy a 23. ábrán látható a tanárok nevei ugyanúgy kerülnek kilistázásra, mint a diákok, ugyanúgy megtudjuk adni az új nevet vagy a régit, ugyanakkor az osztály helyett itt azt a tantárgyat tudjuk módosítani, amit a kiválasztott tanár tanít. Az osztályok listában csak akkor jelenik meg egy osztály ha az „Osztályfőnök” rubrika kipipálásra kerül. Lehetőségünk van, még megadni, hogy az adott tanárt szeretnénk-e rendszergazdaként könyvelni.

Az osztályok Tab-on ugyan ezekkel a módszerekkel lehet feltölteni az adatbázisba új osztályt illetve módosítani.

Ezt követi a regisztrációs fül, amelyen felhasználókat hozhatunk létre már létező személyekhez legyen ez tanár vagy diák. Ki kell választanunk, hogy diáknak avagy tanárnak szeretnénk felhasználót létrehozni. Ezt követően a legördülő listából kiválasztjuk, hogy mely személynek szeretnénk felhasználót létrehozni. Megadjuk a felhasználó nevet és a jelszót, egyelőre manuális későbbiekben a jelszó generált lesz és majd az adott felhasználó megfogja tudni változtatni. Ezt követően kiválasztjuk ha tanárról van szó, hogy szeretnénk-e rendszergazda jogosultságokat adni. Ha készen vagyunk akkor a Regisztrál gomb megnyomása után kapunk értesítést a sikeres vagy sikertelen regisztrációról. Ezeket láthatjuk a 24. ábrán.

Diákok Tanárok Osztályok Regisztrálás

Tanár: ☒ Diák: ☐

Név:

Felhasználónév:

Jelszó:

Rendszergazda: ☐

Regisztrál Mégsem

24. ábra: Regisztrációs felület.

Most, hogy bemutatásra került a rendszergazdai felület, következzen a tanári felület. Hasonlóan ahogyan a rendszergazdai felületet, ezt is a bejelentkezés után érhetjük el és a 25. ábrán látható kép fogad minket. Kettő darab tab közül választhatunk jelenleg:

- Osztályzatok
- Hiányzások

Vegyük alapul az osztályzatok, mint ahogy jelenleg a 25. ábrán látható. Itt kiválasztjuk, hogy mely osztálynak szeretnénk osztályzatot rögzíteni. Ezt követően kiválaszthatjuk a tanulót. A tantárgyak legördülő listában csak a bejelentkezett tanár tantárgyai jelennek meg. Egyelőre tanáronként egy, későbbiekben egy tanárnak több tantárgya is lehet a rendszerben. A tantárgy kiválasztása után válasszuk ki a kívánt osztályzatot és kattintsunk a rögzítés gombra. Ha helyesen kiválasztottunk mindent akkor üzenetet kapunk a rendszertől, hogy sikeresen rögzítettük az osztályzatot.

25. ábra: Tanári felület

A következő menüpont a hiányzások, ahol a tanulóknak tudunk rögzíteni mulasztást. Ezt egyszerűen megtehetjük a 26. ábra alapján. Kiválasztjuk az osztályzat rögzítéshez hasonlóan az osztályt, majd az osztályból a tanulót. Ezt követően a dátum választóval megadjuk a kívánt dátumot és a rögzítés gombra kattintva visszajelzést kapunk a rendszertől, hogy sikeres volt-e a rögzítés vagy nem.

The screenshot shows a web application window titled 'Tanári felület'. It has two tabs: 'Osztályzatok' and 'Hiányzások', with the latter being active. Inside the active tab, there are three dropdown menus. The first is labeled 'Osztály:' and the second 'Tanuló:'. Below them is a date selector showing '2021. április 7., szerda'. At the bottom of the form is a button labeled 'Rögzít'.

26. ábra: Tanári felületen a hiányzások menüpont.

## 8. ÖSSZEGZÉS

A záródolgozatom témájául egy iskolai digitális naplót választottam. Igyekeztem a legkorszerűbb megoldásokat használni továbbá törekedtem a felhasználóbarát megvalósítására. Törekedtem továbbá arra, hogy a felhasználók úgy érezzék, hogy nem a megszokott oldalt használják, illetve a program használatával a digitális naplók egy újfajta irányt vehetnek. Ugyanakkor a legtöbb megszokott funkciót igyekeztem beleépíteni, hogy ne szenvedjenek hiányt sem a diákok, sem a tanárok és a rendszergazdák se. Úgy gondolom, hogy a felhasználó dokumentációval sikeresen elsajátíthatják a rendszer működését.

Ugyanakkor további fejlesztések szükségesek még a rendszerben, ilyen például a weboldalon a tanári válaszadás az üzenetekre, a táblázatok bővítése illetve egy konkrét

menüpont létrehozásra, ahol a diákok az összes jegyüket megtekinthetik hónapokra és tantárgyakra lebontva. Ezeket elkészítése tervben van, későbbiekben beépítésre kerül.

A záradolgozat készítése közben sikerült akár a kutató munkával akár az iskolában tanultakkal olyan tudást elsajátítanom amit biztosan hasznosítani tudok majd a későbbi munkáim során. Megtanultam, hogy milyen egy konkrét projektet megtervezni és megvalósítani. Milyen kihívásokkal kell szembenézni, és milyen teljesen egyedül dolgozni. Rájöttem, hogy sokkal jobb egy csapat tagja lenni ahol a feladatokat fel lehet osztani ezért célszerűbb specializálódni egy adott szakterületre. Továbbá sajnos arra is rá kellett jönnöm, hogy a dizájn nem az erősségem ezért sajnos nem lett valami szemet gyönyörködtető a rendszer ugyanakkor ezzel tudatosulhatott bennem, hogy ezen területen is fejlesztenem kell magam.

Összeségében egy remek kihívásként éltem meg és úgy gondolom, hogy sikerült az általam kitűzött célok többségét megvalósítanom.

## IRODALOM JEGYZÉK

Dr. Szepesné Stiftinger M.(2010.07.27.). Rendszertervezés 5., IR követelménymodell.

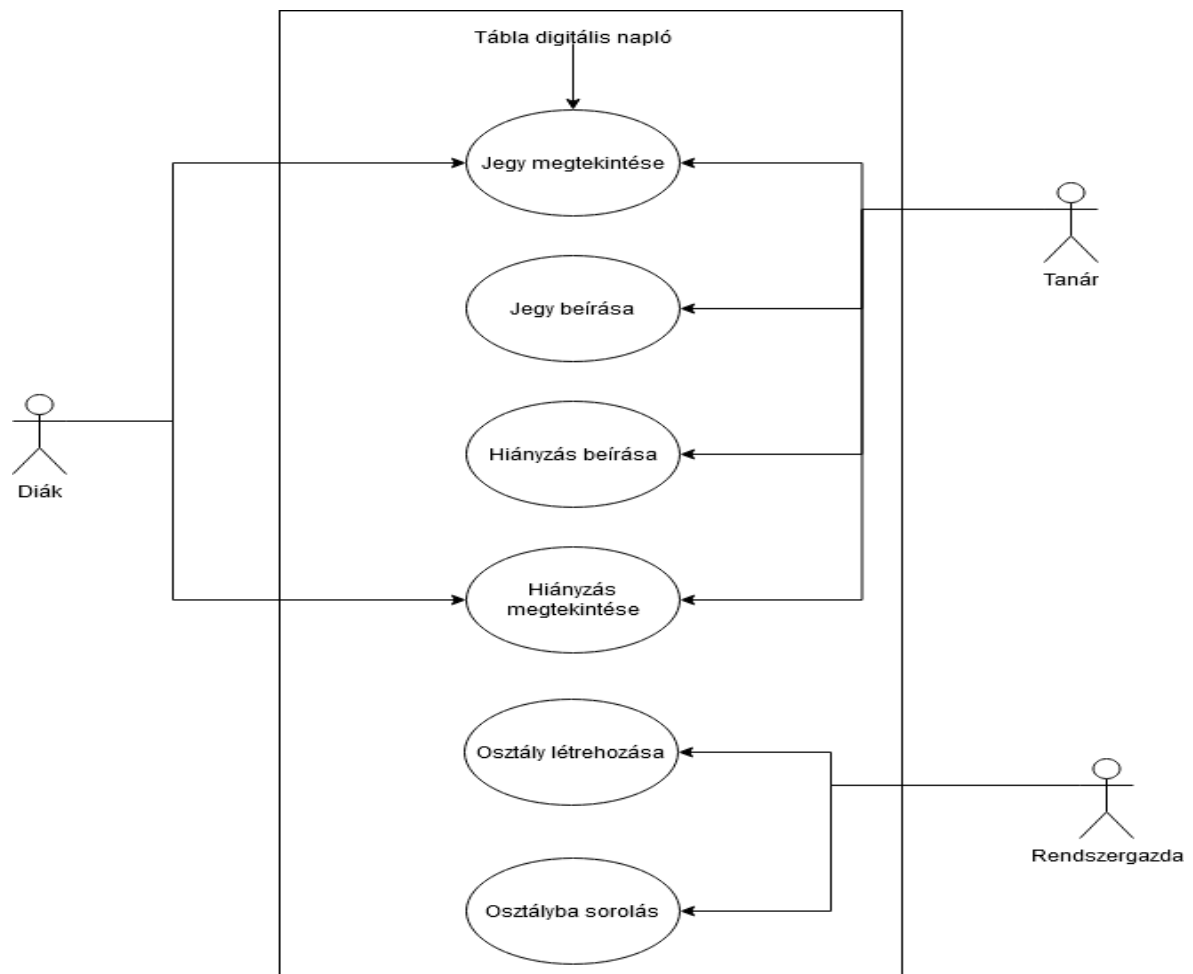
Letöltés időpontja: 2021.02.04., forrás: Digitális tankönyvtár:

[https://regi.tankonyvtar.hu/hu/tartalom/tamop425/0027\\_RSZ5/ch01s03.html](https://regi.tankonyvtar.hu/hu/tartalom/tamop425/0027_RSZ5/ch01s03.html)

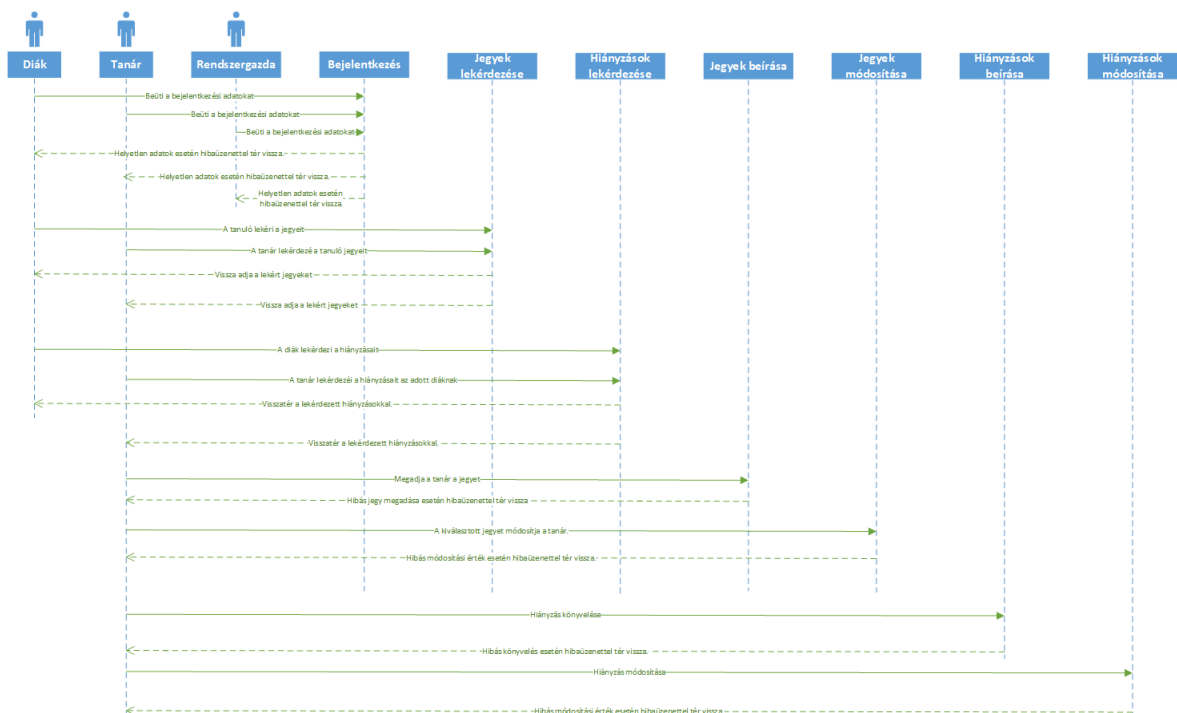
Wikipédia(2019. január 13). Unified Modeling Language. In Wikipédia, a szabad enciklopédia. Letöltés időpontja: 2021.02.04., forrás:

[https://hu.wikipedia.org/wiki/Unified\\_Modeling\\_Language](https://hu.wikipedia.org/wiki/Unified_Modeling_Language).

## MELLÉKLETEK

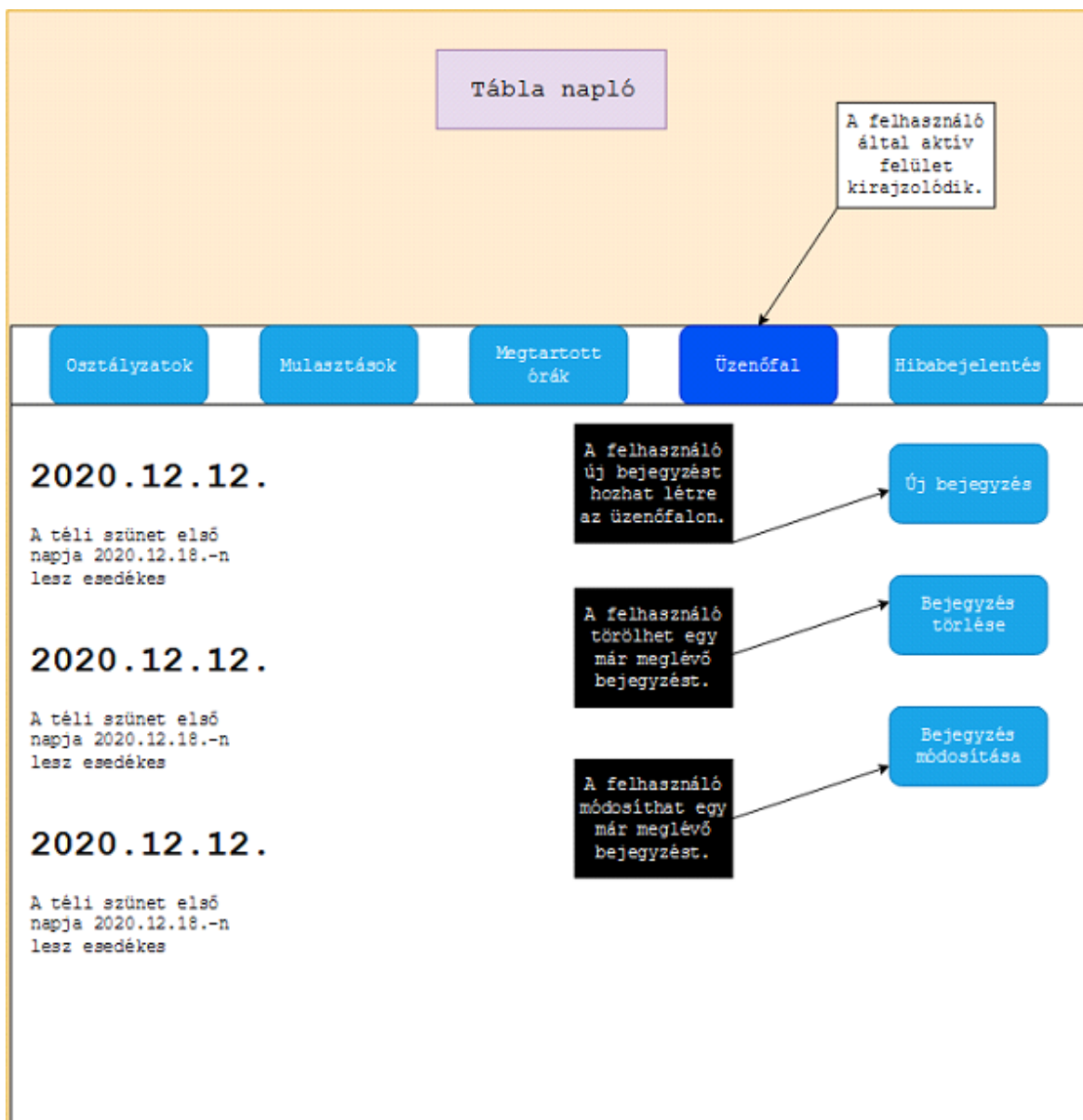


1. melléklet: Az asztali alkalmazás Use-Case diagramja.

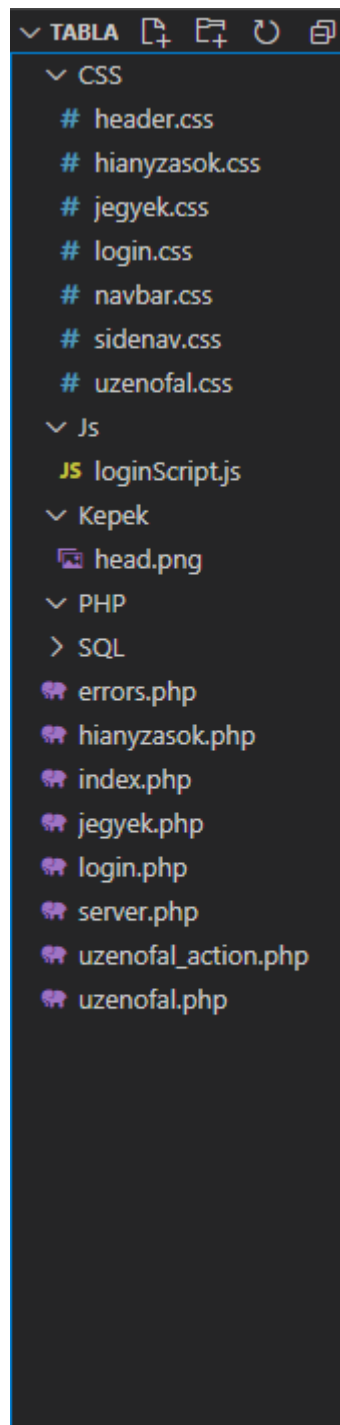


2. melléklet: Az asztali alkalmazás Üzleti-Szekvencia diagramja.





3. melléklet: A weboldal egy oldalának a tervezete.



4. melléklet: Fájlkiterjesztés szerinti rendezés webfejlesztés közben.

-- Tanárok --

-- Tanárok --

-- Tanárok --

Kerek Ferenc

Kis Géza

Szabó Árpád

Kell végre egy teszt

Másik teszt

Harmadik teszt

5. melléklet: Tanár kiválasztása a weboldalon üzenet küldéshez.

## HALLGATÓI NYILATKOZAT

Alulírott

**Török Bendegúz Krisztián** a Szegedi SZC Vasvári Pál Gazdasági és Informatikai Technikum hallgatója kijelentem, hogy a **Iskolai Digitális Napló** című záródolgozat a saját munkám.

Kelt:

2021. április 07.

---

aláírás