

**SAPIENTIA ERDÉLYI MAGYAR TUDOMÁNYEGYETEM
MAROSVÁSÁRHELYI KAR,
INFORMATIKA SZAK**



**SAPIENTIA
ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM**

Data Delivery: A jegyek kezelésének egyszerűsítése az egyetemi hallgatók számára

DIPLOMADOLGOZAT

Témavezető:
Osztián Erika,
Egyetemi adjunktus
Osztián Pálma-Rozália,
Egyetemi tanársegéd

Végzős hallgató:
Bálint Lehel

2023

**UNIVERSITATEA SAPIENTIA DIN CLUJ-NAPOCA
FACULTATEA DE ȘTIINȚE TEHNICE ȘI UMANISTE,
SPECIALIZAREA INFORMATICĂ**



**UNIVERSITATEA
SAPIENTIA**

Data Delivery: Simplificarea managementului notelor pentru
studenții universitari

LUCRARE DE DIPLOMĂ

Coordonator științific:
Osztián Erika,
Lector universitar
Osztián Pálma-Rozália,
Asistent universitar

Absolvent:
Bálint Lehel

2023

**SAPIENTIA HUNGARIAN UNIVERSITY OF
TRANSYLVANIA**
FACULTY OF TECHNICAL AND HUMAN SCIENCES
COMPUTER SCIENCE SPECIALIZATION



SAPIENTIA
HUNGARIAN UNIVERSITY
OF TRANSYLVANIA

Data Delivery: Streamlining Grade Management for University
Students

BACHELOR THESIS

Scientific advisor:

Osztián Erika,
Lecturer

Osztián Pálma-Rozália,
Assistant professor

Student:

Bálint Lehel

2023

Declarație

Subsemnatul/a BALINT LEHEL, absolvent(ă) al/a specializării INFORMATICA, promoția 2013 cunoscând prevederile Legii Educației Naționale 1/2011 și a Codului de etică și deontologie profesională a Universității Sapientia cu privire la furt intelectual declar pe propria răspundere că prezenta lucrare de licență/proiect de diplomă/disertație se bazează pe activitatea personală, cercetarea/proiectarea este efectuată de mine, informațiile și datele preluate din literatura de specialitate sunt citate în mod corespunzător.

Localitatea, TÂRCHI MURES
Data: 06.13.2013

Absolvent

Semnătura... Balint

LUCRARE DE DIPLOMĂ

Coordonator științific:
Lect. univ. dr. Osztian Erika

Candidat: **Bálint Lehel**
Anul absolvirii: **2023**

a) Tema lucrării de licență:

Aplicarea livrării datelor pentru a monitoriza și analiza performanța academică a studenților prin intermediul aplicației ”Data Delivery”

b) Problemele principale tratate:

- Identificarea și analiza principalelor provocări și probleme în implementarea și utilizarea aplicației
- Examinarea metodelor de prelucrare a datelor și generare a statisticilor relevante pentru monitorizarea performanței academice
- Propunerea de soluții și îmbunătățiri pentru optimizarea funcționalității și interfeței aplicației, în vederea îmbunătățirii experienței utilizatorilor

c) Desene obligatorii:

-diagrama arhitecturii sistemului Data Delivery, inclusiv interacțiunea dintre aplicație API și interfața profesorului

-capturi de ecran al interfeței aplicației Data Delivery, care prezintă funcționalitățile principale și informații necesare

d) Softuri obligatorii:

A fost dezvoltată o aplicație numită Data Delivery, care permite gestionarea rezultatelor și datelor statistice ale studenților. Aplicația client, scrisă în Android, oferă posibilitatea studenților de a urmări și evalua progresul lor în diferite cursuri și ore. De asemenea, interfața web și Strapi oferă funcționalități pentru îndeplinirea sarcinilor administrative, cum ar fi înregistrarea notelor și înregistrarea profesorilor.

e) Bibliografia recomandată:

- [1] Strapi's Documentation (<https://docs.strapi.io/>)
- [2] MPAndroidChart Documentation (<https://weeklycoding.com/mpandroidchart-documentation>)
- [3] Orhani, S., Saramati, E., & Drini, L. (2022). Electronic school diary for statistical analysis of student progress. Brazilian Journal of Science, 1(3), 58-65.
- [4] Android Documentation (<https://developer.android.com/docs>)

f) Termene obligatorii de consultații: săptămânal, preponderent online

g) Locul și durata practiciei: Universitatea „Sapientia” din Cluj-Napoca,

Facultatea de Științe Tehnice și Umaniste din Târgu Mureș, laboratorul de informatică (sala 414)

Primit tema la data de: 10.06.2022.

Termen de predare: 20.06.2023.

Semnătura Director Departament

Semnătura coordonatorului

Semnătura responsabilului
programului de studiu

Semnătura candidatului
Bálint Lehel

Kivonat

A statisztikai adatok rendkívül fontos szerepet játszanak a mai modern, fejlődő világunkban, így az oktatásban is egyaránt, mivel segítenek megérteni, elemzni a diákok és tanárok teljesítményét, valamint az oktatási rendszereket is. Dolgozatom témája egy olyan rendszer implementálása, amely egyfajta online naplóként funkcionál egyetemi hallgatók számára, és egyben lehetőséget nyújt arra, hogy a hallgatók nyomon követhessék, elemzéseket végezhessenek, statisztikákat értelmezhessenek, saját, és társaik jegyein egyaránt.

Az elkészített alkalmazás neve „Data Delivery”, ami kizárolag Androidos rendszerek felületén elérhető mivel Android-Kotlin technológia segítségével íródott, a diák felhasználókat megcélozva. Az mobilalkalmazás egy Strapi.io API-al kommunikál az adatok elérése és kezelése érdekében. Ezen kívül kifejlesztettem .NET Core technológiával egy webalkalmazást amely a jegyek egyszerű menedzselése érdekében készült, kizárolag a tanár felhasználóknak.

Az elkészült alkalmazás hasznos lehet az egyetemi hallgatók számára a tanulmányi jegyek nyomon követésében és az elemzésükben ezáltal értékelni tudják saját teljesítményüket és könnyebben kidolgozhatnak személyre szabott tanulási stratégiákat. Az alkalmazás könnyen kezelhető felhasználói felülettel rendelkezik, és széles körű funkcionalitást kínál a diákok számára.

A dolgozatomban szeretném részletesen bemutatni az applikáció megtervezésének lépésein, a működési elveket. Továbbá ismertetni szeretném a felhasznált technológiákat, a megvalósított és a jövőbeli fejlesztési lehetőségeket.

Kulcsszavak: tanulmányi eredmények, statisztika, e-napló

Rezumat

Datele statistice joacă un rol extrem de important în lumea noastră modernă și în dezvoltare, inclusiv în domeniul educației, deoarece ajută la înțelegerea și analizarea performanței elevilor și a profesorilor, precum și a sistemelor de învățământ. Tema lucrării mele este implementarea unui sistem care funcționează ca un jurnal online pentru studenții universitari, oferindu-le posibilitatea de a-și urmări progresul, de a efectua analize, de a interpreta statistici și de a vizualiza propriile note și ale colegilor lor.

Numele aplicației dezvoltate este "Data Delivery", care este disponibilă exclusiv pe sisteme Android, deoarece a fost scrisă folosind tehnologia Android-Kotlin, adresându-se utilizatorilor studenți. Aplicația mobilă comunică cu un API Strapi.io pentru a accesa și gestiona datele. În plus, am dezvoltat o aplicație web folosind tehnologia .NET Core pentru a facilita gestionarea ușoară a notelor, exclusiv pentru utilizatorii profesori.

Aplicația finalizată poate fi utilă pentru studenții universitari în urmărirea performanțelor academice și analizarea progresului lor. Aceasta le permite să-și evaluateze propria performanță și să dezvolte strategii personalizate de învățare mai eficientă. Aplicația are o interfață prietenoasă pentru utilizatori și oferă o gamă largă de funcționalități pentru studenți.

În lucrarea mea, doresc să prezint în detaliu pașii implicați în proiectarea aplicației și să explic principiile sale de funcționare. De asemenea, voi descrie tehnologiile utilizate, funcțiile implementate și voi discuta potențialele oportunități de dezvoltare viitoare.

Cuvinte de cheie: rezultate academice, statistică, e-jurnal

Abstract

The statistical data plays an extremely important role in our modern, developing world, including in education, as it helps understand and analyze the performance of students, teachers, and educational systems. The topic of my paper is the implementation of an online log system for university students, named "Data Delivery." This system provides students with the opportunity to track their progress, perform analyses, interpret statistics, and view their own grades as well as those of their peers.

The "Data Delivery" application is exclusively available on Android systems, as it was developed using Android-Kotlin technology with student users as the target audience. The mobile application communicates with a Strapi.io API to access and manage the data. Additionally, a web application has been developed using .NET Core technology to facilitate easy grade management, exclusively for teacher users.

The completed application can be highly useful for university students in tracking their academic performance and analyzing their progress. This allows them to evaluate their own performance and develop personalized learning strategies more effectively. The application boasts a user-friendly interface and offers a wide range of functionality for students.

In my paper, I aim to present in detail the steps involved in designing the application and explain its operating principles. Furthermore, I will describe the technologies used, the implemented features, and discuss potential future development opportunities.

Keyword: study results, statistics, e-diary

Tartalomjegyzék

1. Bevezető	11
2. Elméleti megalapozás	13
2.1. Szakirodalmi áttekintő	13
2.2. Hasonló alkalmazások	14
3. A rendszer specifikációi	17
3.1. Felhasználói követelmények	17
3.2. Rendszerkövetelmények	19
3.2.1. Funkcionális követelmények	19
3.2.2. Nem-funkcionális követelmények	20
4. Tervezés	21
4.1. Architektúra	21
4.2. Használt technológiák és architektúrák	21
4.3. Menedzsment	23
4.3.1. Arculat megtervezése	23
4.3.2. Feladatok beosztása	24
4.3.3. Verziókövetés	24
4.4. Adatbázis és adatkezelés	25
4.5. Szekvencia diagramok	27
5. A szoftver bemutatása	30
5.1. Android alkalmazás	30
5.1.1. Bejelentkezés	30
5.1.2. Statisztika oldal	31
5.1.3. Értesítések oldal	32
5.1.4. Saját tanfolyamok	33
5.2. Web alkalmazás	35
5.2.1. Bejelentkezés és főoldal	35
5.2.2. Jegyek/Minősítések	36
5.2.3. Tanfolyamok/Diákok	36
6. Limitációk	39
Összefoglaló	40

Ábrák jegyzéke **42**

Irodalomjegyzék **43**

1. fejezet

Bevezető

Az egyetemi napló egy kommunikációs eszköz, amelyet az egyetemi oktatók, tan-székvezetők és az egyetemi vezetés hozott létre az egyetemi információk és diákokkal kapcsolatos adatok megosztására. Az elektronikus napló (e-napló) használata jelentősen elősegíti az egyetemi vezetést, a naplóban szereplő statisztikákkal foglalkozó oktatókat és adminisztrátorokat munkáját is. Az e-napló nagymértékben csökkenti tudja az egyetemi adminisztrációval kapcsolatos tevékenységekre szánt időt. Míg a hagyományos naplóban manuálisan rögzítik a jegyeket és az eredményeket, az e-napló előre meghatározott szolgáltatásokat kínál, amelyeket a rendszer automatikusan ellenőriz és kiszámol. Az elektronikus hozzáférés lényegesen gyorsabb, mint a hagyományos adminisztrációs megközelítés, így az oktatók több időt tudnak fordítani a diákokkal való munkára és az oktatási tevékenységek tervezésére, ahelyett, hogy a napló formális kitöltésével foglalkoznának.

A statisztikai adatok a tanulók számára is a rendkívül fontosak az egyetemi tanulmányok alkalmával. Ezek az adatok lehetővé teszik számukra, hogy igen összetett, széles körű képet kapjanak saját teljesítményükről, fejlődésükéről, ezen kívül az egyetemi élet egyéb aspektusairól, például a kurzusok tananyagok nehézségi szintjéről, tanárok stílusairól.

Az egyetemi statisztikai adatok révén a tanulók nyomon követhetik saját akadémiai eredményeiket, példának okáért a jegyeiket, valamint vizsgaeredményeiket. Ez segít nekik az erősségeik, és gyengeségeik azonosításában, és nem utolsó sorban abban is, hogy megtegyék a szükséges lépéseket a tanulásuk javításának érdekében. Az online napló felhasználói az adatok alapján képesek megérteni, mely területeken teljesítenek jobban, valamint hol van szükségük további erőfeszítésekre, valamint támogatásra. Ezzel együtt a tanulók részére az egyetemi statisztikai adatok lehetőséget nyújtanak a saját és más diákok teljesítményének összehasonlítására (a tanfolyamon szerezett átlagteljesítmény alapján). Ez segít nekik abban, hogy megtudják, hol állnak a többiekhez mérten, valamint motivációt ad nekik a fejlődésre, továbbá versengésre buzdítja őket.

Manapság a legtöbb ember el sem tudja képzelni a napját okostelefon nélkül, így van ez a diákokkal is. Az okostelefonok lehetővé teszik, hogy gyorsan hozzáférjünk szinte bár-milyen adathoz amire éppen szükségünk lehet. Ennek köszönhetően rengeteg mindenben megkönnyítik az emberek minden napjait. Feltevődhet akkor a kérdés hogyha már adott ez a lehetőség akkor miért ne használnánk ki az oktatásban is ennek előnyeit?

Mindig is kíváncsi voltam és szerettem volna egyetemi éveim alatt tudni, hogy mennyire vannak jó jegyeim, mennyire kéne többet tanuljak, hol helyezkedek el a szakon, félév közben mennyi esélyem van ösztöndíjas helyet elérni, mi vár rám a felsőbb éveken,

mennyire nehéz egyes tantárgyakat teljesíteni. De sajnálatos módon a GDPR szabályok és egy erre alkalmas platform hiánya miatt nem volt erre lehetőségem. Ezért is választottam egy ilyen alkalmazás megvalósítását amelyben a hozzáám hasonló diákok kérdéseikre választ kaphatnak, természetesen anonim módon a szabályok megsértése nélkül.

A *Data Delivery* alkalmazás egy ilyen megoldás, amely segíti a diákok jegyeinek követését és a tanulmányi statisztikák megjelenítését elektronikus formátumban a saját okostelefonunkon.

A *Data Delivery* egy Android-Kotlin alapú alkalmazás, amely lehetővé teszi az egyetemi hallgatók részére, hogy könnyedén nyomon kövessék a tanulmányi jegyeiket, ezen kívül elérjék a releváns statisztikákat. Az alkalmazás azzal a céllal került megtervezésre, hogy támogatást nyújtson a tanulási előrehaladás hatékonyabbá tételeben, valamint a tanulók személyre szabott támogatásában.

Dolgozatom célkitűzése, hogy részletesen bemutassam a „*Data Delivery*” alkalmazás tervezését, fejlesztését, valamint működését. Kiemelt figyelmet fordítok az alkalmazás architektúrájára, a felhasználói felület tervezésére, elméleti megalapozásra, valamint implementációjára és a funkcionálisok differenciált leírására.



1.1. ábra. Data Delivery Logo

2. fejezet

Elméleti megalapozás

Ebben a fejezetben szeretném bemutatni az alkalmazásom tervezését és megvalósítását megalapozó elméleti és kutatási hátteret. Áttekintjük a releváns szakirodalmakat és bemutatók néhány hasonló alkalmazást is, amelyekből ötleteket merítettem az alkalmazásom fejlesztése során.

2.1. Szakirodalmi áttekintő

Kutatásom során különböző kutatásokat és tanulmányokat vizsgáltam meg, amelyek a diákok teljesítményével foglalkoznak és az elektronikus naplók hatásait, előnyeit és hátrányait vizsgálják. Ennek következtében arra jutottam, hogy számos kutató érvel amellett, hogy az elektronikus napló jó hatással van az oktatásra. Az alábbiakban szeretnék összefoglalni néhány olyan olyan eredményt amit érdekesnek, kiemelkedőnek találtam.

- Senad Orhani is társai kutatásukban [OSD22] kimutatták, hogy egy jól megtervezett online napló a tanárok és diákok körében is szinte maximális elégedettséget és elfogadást váltott ki. Ez annak volt köszönhető, hogy a rengeteg előny mellett (pontos eredmények a számításokban, könnyű hozzáférés, az adminisztratív folyamatok automatizálása, a papír alapú dokumentumok használatának csökkentése stb.) mindenki néhány hátránnal (használati nehézségek, adatvesztés, személyeség csökkenése) kellett szembenézni, a régi formátumú naplóhoz viszonyítva.
- Ugyanakkor Sekione R. Jeremiah és Joel S. Mtebe PhD egyik kutatásukban [JMP18] pedig arra is rávilágítottak, hogy az elektronikus naplók nem csak a diákok és a tanárok körében rendkívül népszerűek hanem a szülők körében is. A kutatás szerint ez annak tudható be, hogy az e-napló fokozza a szülők részvételét a gyerekük iskolával kapcsolatos tevékenységeiben.
- Szépvölgyi Katalin könyvében [Szé16], azon a véleményen van, hogy egy elektronikus naplónak ugyanolya előnyei és hátrányai vannak mint az elektronikus dokumentumknak. Előnyei, hogy tartalmazza az összes kötelező és szükséges információt, archiválható, megosztható, távolról elérhető, gyors visszakereshetőséget biztosít az adattartalomban, támogatja a statisztikai adatszolgáltatást és a statisztikai trend kimutatását, környezetbarát, helytakarékos. Hátránynak viszont felhozható az, hogy nehezebb megvédeni az adatokat és a hitelesítés problémája is fent áll.

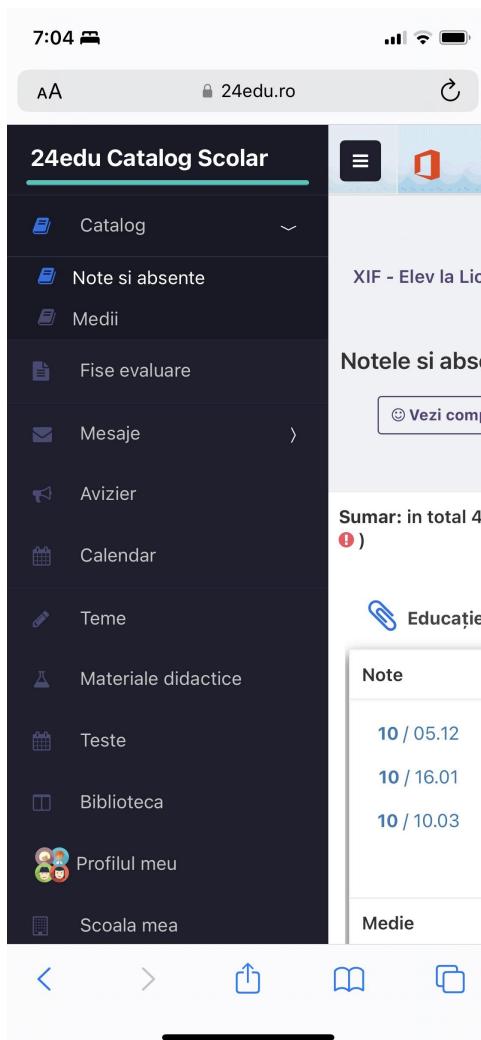
- Calmorin és Laurentina írása [Cal97] is rávilágít arra, hogy a statisztika nagyon fontos szereppel rendelkezik az oktatásban. A statisztikák segítségével tudjuk értékelni a hallgatók és a tanárok teljesítményét, valamit oktatási módszerek hatékonysága is könnyedén kiolvasható. Ezen kívül egészsges versenyhelyzetet idézhet elő a diákok között, ezzel is fokozva a teljesítményüket

2.2. Hasonló alkalmazások

A szoftver tervezése és fejlesztése előtt megvizsgáltam néhány már működő alkalmazást amelyek elektronikus naplók különböző funkcióit implementálják. Ezen alkalmazások áttekintése segített megismerni a már meglévő megoldásokat, ezeknek különböző előnyeit és hátrányait, valamint inspirációt nyertem a saját alkalmazásom tervezésére. Ezek közül szeretnék párat bemutatni:

- A **24edu Catalog Scolart** a romániai iskolák többsége használja manapság, rengeteg hasznos funkcióval rendelkezik (lásd 2.1 ábra). A diákok hiányzásait és jegyeit jeleníti meg valós időben a diákok és a szülők számára egyaránt (lásd 2.2 ábra). Emellett lehetőség van dátumokat megadni valamint házi feladatokat feladni. A sok hasznos funkció mellett negatív szempontnak hoznám fel azt, hogy csak a tanárok tudják a diákokat értékelni és a diákoknak nincs lehetőségük a tanárokról vagy akár tantárgyakról, tanítási módokról értékelést leadni. Ide sorolnám azt is, hogy a médián kívül a diákok számára semmiféle hasznos statisztika és adat nem jelenik meg amihez képest viszonyítani tudnák a saját teljesítményüket.
- Egy másik alkalmazás a **Neptun** rendszer amelyet felsőoktatási intézmények használnak köztük, a Sapientia, ahol tanulmányaimat végzem. A Neptunnak rengeteg olyan funkciója van amelyek a diákok, tanárok és adminisztrátorok életét is egyszerűbbé teszik. Lehet fizetéseket intézni, dokumentumokat kigenerálni névre szólóan, tanárokat értékelni, tantárgyakat felvenni, vizsgákra jelentkezni (lásd 2.3 ábra) és még sok más. Viszont személyes tapasztalataim alapján megjegyeznék párt negatívumot is. A kezelő felület nem valami átlátható sok időbe kerül megtalálni kezdő felhasználóként azt a funkciót amit használni szeretnénk, csak a végső jegyek tárolhatóak itt, a telefonos verzió pedig nehezen volt használható számomra.
- A **School eDiary** volt az az alkalmazás amelyik a leginkább elnyerte a tetszésem, mivel a rengeteg alap funkció (lásd 2.5 ábra) mellett amikkel egy ilyen alkalmazás rendelkezik sok újdonsággal és egy rendkívül letisztult és könnyen értelmezhető kezelőfelülettel is találkoztam. Ilyen például a buszkövető (lásd 2.4 ábra) amihez beépített térkép társul, a házi feladat feltöltési lehetőség, órarend, jegyzetek és egyebek. Fontos kiemelni, azonban hogy itt is csak minimális statisztikai mutatóval futottam össze és a diákoknak itt se volt lehetőségük arra, hogy visszajelzéseket tudjanak adni tapasztalataikról.

Annak köszönhetően, hogy kutatásom során rengeteg pozitív visszhangot véltem felfedezni az elektronikus naplók irányába, megalkottam a saját alkalmazásomat. Próbáltam a rengeteg alkalmazás közül a hasznos funkciókat és megoldásokat ötvözni és kizárnival azokat amelyek negatív hatást idézhetnek elő egy ilyen applikációban.



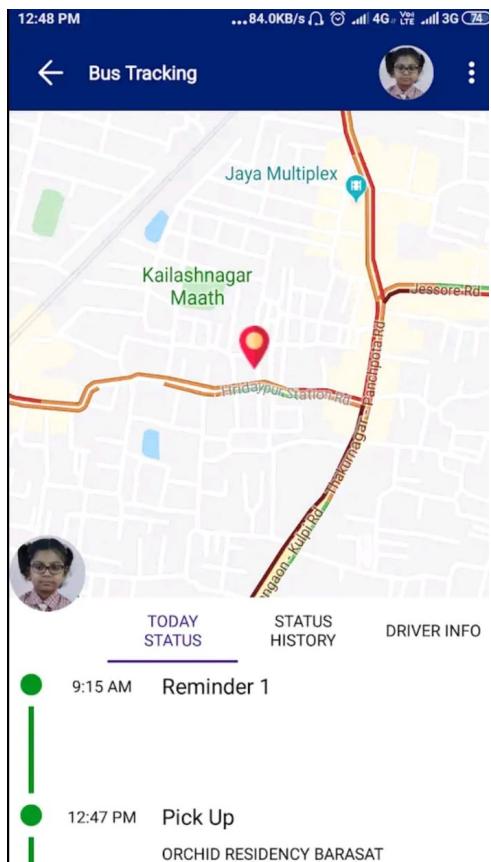
2.1. ábra. 24edu Catalog - menü

Subject	Note	Grade	Avg Grade
Educație artistică	NOTE 10/05.12 10/16.01 10/10.03 ABSENTE 21.11 19.12 10.02 21.04 28.04 19.05	MEDIE 10 (10.0)	MEDIE 10.0 ANUALA
Educație fizică	NOTE 9/07.10 10/20.01 9/14.03 8/02.05 10/13.06 ABSENTE 14.02	MEDIE 9 (9.2)	MEDIE 9.0 ANUALA
Geografie	NOTE 7/01.11 8/20.12 10/16.01 8/07.03 8/12.05 10/19.05 ABSENTE	MEDIE 9 (8.5)	MEDIE 9.0 ANUALA
Istorie	NOTE 9/06.10 8/06.12 9/06.02 9/04.05 ABSENTE 18.01 10.02 13.02 27.02 02.03	MEDIE 9 (8.5)	MEDIE 9.0 ANUALA

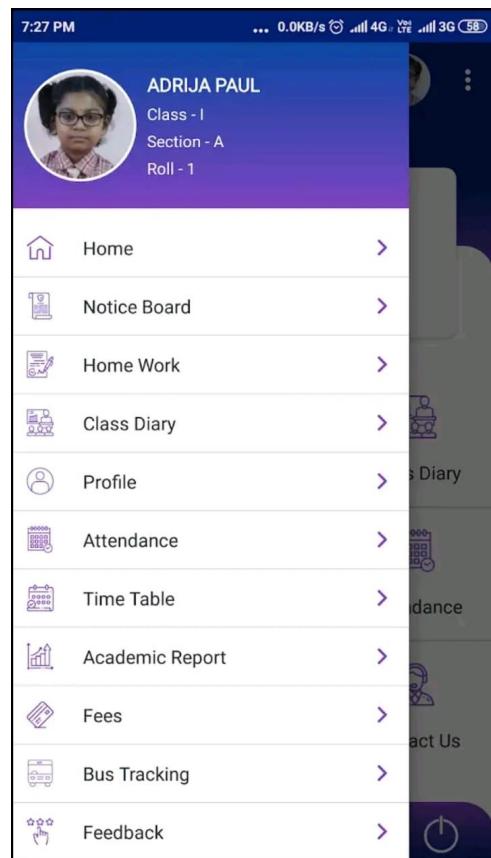
2.2. ábra. 24edu Catalog - jegyek

Tárgy	Tárgykód	Kurzus	Vizsgatípus	Vizsgajel.	Kezdés	Telephely	Termék	Oktatók	Limit (Kurzus limit)	Beszámít	Megjelent	Igazoltan távol	Eredmény	Leírás
Kriptológia és adatbázisok	MBEI0541	MBInfo_2020_6	Nincs megadva	Vizsga	2023. 05. 16. 9:00:00			Dr. Márton Gyöngyvér	28/	✓	✓		i	+
Modern technológiák (.NET)	MBME0301	MBInfo_2020_6	Nincs megadva	Vizsga	2023. 04. 24. 8:00:00			Dr. Jánosi-Rancz Katalin Tünde	43/	✓	✓		i	+
Mesteréjes intelligencia	MBEI0111	MBInfo_2020_6	Nincs megadva	Vizsga	2023. 05. 15. 10:00:00			Dr. Dávid László	36/	✓	✓		i	+
Szakdolgozatírás	MBLI0011	MBInfo_2020_6	Nincs megadva	Vizsga	2023. 04. 24. 8:00:00			Dr. Jánosi-Rancz Katalin Tünde	55/	✓	✓		i	+
Információ keresés	MBEI0591	MBInfo_2020_6	Nincs megadva	Vizsga	2023. 05. 18. 12:00:00			Dr. Idánzán David-Andrei	20/	✓	✓		i	+

2.3. ábra. Neptun - Felvett vizsgák



2.4. ábra. School eDiary - Busz követő

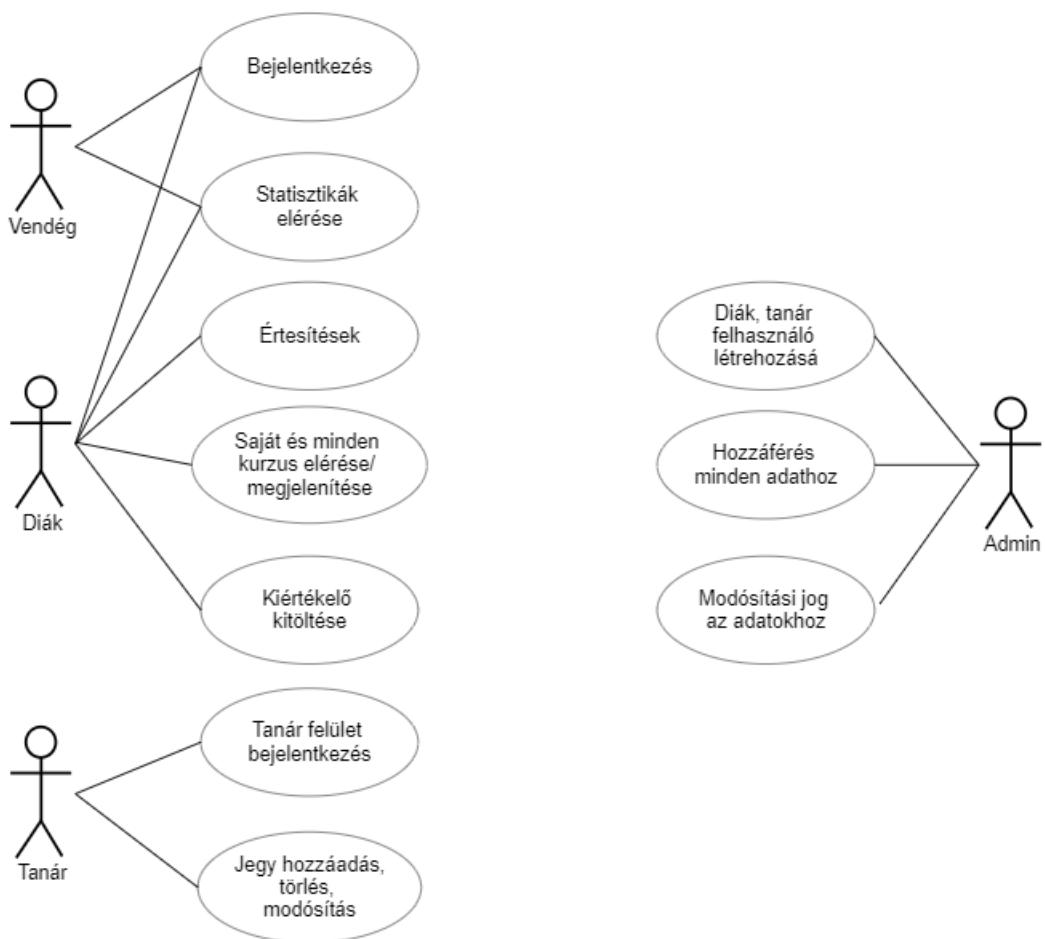


2.5. ábra. School eDiary - Menü

3. fejezet

A rendszer specifikációi

3.1. Felhasználói követelmények



3.1. ábra. Use Case Diagram

Az alkalmazáson belül a felhasználok többféle szerepkörbe tartozhatnak (vendég, diákok, tanár, admin), ezek minden különböző műveleteket végezhetnek, ahogyan az a használati eset diagramon is látható (lásd 3.1 ábra). Először is megneveznék két olyan műveletet amelyet **bárki** végezhet (lásd 3.1 ábra):

- **A bejelentkezés:** a felhasználó a saját egyedi kódjával (Neptun Id) és jelszó segítségével jelentkezhet be. Nincs regisztráció, csupán előre létrehozott felhasználók léteznek mivel, az adott egyetemen tanuló diákok közül más nem rendelkezhet felhasználóval
- **A főoldalra való belépés (statisztikák megtekintése):** itt minden felhasználó szabadon böngészheti az adatbázisban található jegyeket akár bejelentkezés nélkül is, különböző szűrő opciókat használva. Ezen kívül lehetőség nyílik arra is, hogy összehasonlítsokat végezzen. A statisztikák és adatokat anonim módon jelennek meg, név és egyéb azonosító nélkül.

A **diákok** ezeken kívül a következő lehetőségekkel találják szembe magukat:

- **Az értesítések:** a bejelentkezett diák meg tudja tekinteni a rendszerbe beírt jegyeit, a legfrissebb minősítéstől a legrégebbiig az összeset sorrendben.
- **Saját tanfolyamok:** minden diák a saját kurzusait látja, kiválasztáskor pedig megjelenik egy részletes nézet a tantárgyhoz kapcsolódó jegyekről és statisztikai mutatókról
- **Minden tanfolyam:** az egyetemen található összes kurzus jelenik meg, a tanfolyam kiválasztásakor pedig különböző diagramok és statisztikai mutatók
- **Kiértékelő:** a diák akkor találkozhat ezzel az oldallal ha elvégzett egy tanfolyamot (rendelkezik végső jeggyel) és meg szeretné azt tekinteni, a megtekintés előtt ki kell értékelnie az adott kurzust

A **tanároknak** egy teljesen új webes felület lett létrehozva, ōk az android applikációnak csak a vendég funkcióit tudják használni. A webes felületen pedig a következő műveletek állnak rendelkezésükre:

- **A bejelentkezés:** a tanár email cím és jelszó segítségével jelentkezhet be. A mobilos applikációhoz hasonlóan itt is előre létrehozott felhasználók vannak, mivel szigorúan csak az egyetemen tanító tanárok számára elérhető a bejelentkezés.
- **Jegyek módosítása:** ezen az oldalon megjelenik az összes jegy amelyet az adott tanár adott. Ezeket mohosítani és törölni is tudja.
- **Jegy leírása:** a tanár kiválasztja az adott kurzust és diákokat, ezt követően pedig megadja a szükséges információkat és a jegy bekerül az adatbázisba. Ezen kívül rendelkezésre áll egy automatikus végső jegy számítható opció amely a tanár döntése szerint felülírható vagy elfogadható.

Az **admin** felhasználók a Strapi.io kezelőfelületét tudják használ aniz adatok menedzselése érdekében:

- **Felhasználók kezelése:** az admin felhasználók számára adott a lehetőség hogy diákok és tanár felhasználókat hozzanak létre, töröljönek vagy módosítsanak. Ezt csak ők tehetik meg, ezért nem szerepel regisztráció funkció sem a diákok sem a tanár felületén.
- **Adatok hozzáférése és módosítása:** az admin típusú felhasználók értelemben szerűen hozzáférnek rendszerben szereplő összes adathoz és ezeket menedzselni tudják.

3.2. Rendszerkövetelmények

3.2.1. Funkcionális követelmények

- Bejelentkezés:
 - Diák:
 - * A Neptun rendszerhez hasonlóan minden diáknak rendelkezik egy egyedi kód-dal amely a bejelentkezéshez szükséges, a jelszó pedig ugyanez a kód összefűzve az adott diáknak születési évével. Ezekkel az adatokkal történik a bejelentkezés.
 - * Bejelentkezés nélkül csak a statisztika oldalhoz lehet hozzáférni amely mindenki számára elérhető anonim adatokat tartalmaz
 - * Bejelentkezéskor a Strapi API-hoz küld egy kérést az alkalmazás, hogy az ellenőrizze a megadott e-mail cím és jelszó páros helyességét, illetve, ha a válasznak az alkalmazás azt kapja vissza, hogy sikeres volt a bejelentkezés akkor tovább fog navigálni az értesítések oldalra. Ellenkező esetben újra meg kell adni az adatokat a bejelentkezéshez
 - Tanár:
 - * A felhasználók létrehozásakor minden felhasználónak az egyetemi email cím és jelszó kombinációja szerepel az adatbázisban, így a felhasználók ezekkel az alapértelmezett adatokkal jelentkezhetnek be.
 - * Bejelentkezés nélkül nincs hozzáférés egyik oldalhoz sem
 - * Bejelentkezéskor a Strapi API-hoz küld egy kérést az alkalmazás, hogy az ellenőrizze le, hogy a megadott e-mail cím és jelszó páros helyes-e, illetve, ha a válasznak az alkalmazás azt kapja vissza, hogy sikeres volt a bejelentkezés akkor tovább fog a főoldalra
- Kijelentkezés: ebben az esetben a felhasználók be kell legyenek jelentkezve és a kijelentkezés menüpontot kell használják a navigációban. Amelyet kiválasztva megjelenik a „Logout” ablak és ha ezt elfogadja akkor megerősíti, hogy valóban ki akar jelentkezni.
- A tanárok csak a saját tanítványaik jegyeit menedzselhetik és láthatják. A jegyek menedzselése során a meglévő adatok hozzáadódnak, illetve frissülnek a Strapi adatbázisban.

3.2.2. Nem-funkcionális követelmények

- Az **Android** applikáció használata lehetséges bármilyen Android készüléken amely rendelkezik internet kapcsolattal és legalább 6.0 operációs rendszerrel.
- A **webapplikáció**: használható bármely eszközön amelyre bármilyen féle böngésző van telepítve és rendelkezik internet kapcsolattal.
- Fejlesztési követelmények:
 - Az Android rendszer esetében az Android Studio nevű fejlesztői környezet.
 - A webapplikáció esetén szükséges a Visual Studio nevű fejlesztői környezet.
 - Ezenkívül mindenki szüksége van a további eszközökre: Github verziókövető rendszer, Strapi.io az adatok tárolása és kezelésére, Visual Studio Code fejlesztői környezet a Strapi.io szerver futtatásának érdekében.
- Külső követelmények:
 - Az adatok biztonságos tárolása a Strapi.io API által amely biztosítja, hogy mindenki csak az általa megtekinthető adatokhoz férjen hozzá.
 - Anonimitás megőrzése, az egyéni jogok védelme érdekében

4. fejezet

Tervezés

4.1. Architektúra

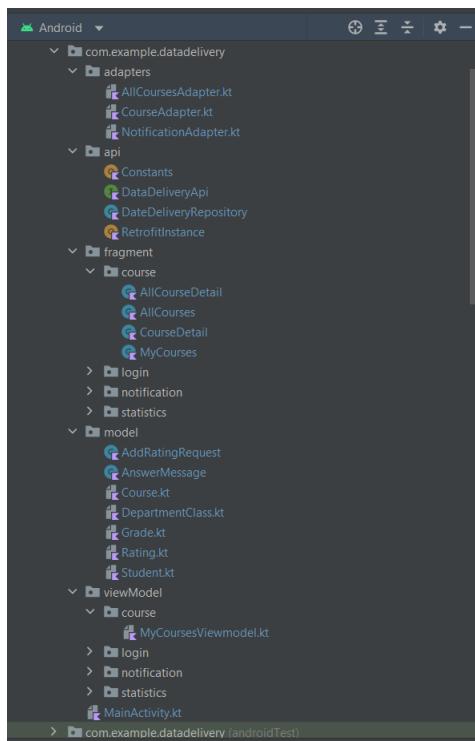


4.1. ábra. A rendszer architektúrája

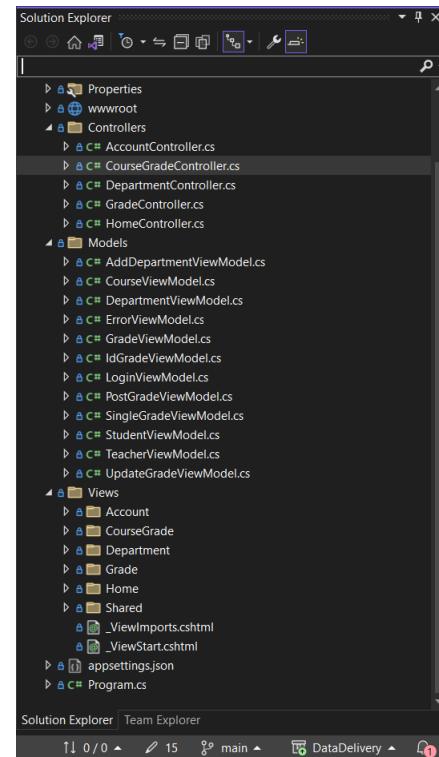
A 4.1 ábrán látható a jelenlegi szoftver egésze egy Android alkalmazás és egy Web alkalmazás amely a tanár felület funkcióit implementálja. A Android applikációm egy egyszerű de ugyanakkor gyakran használatos MVVM(Model-View-ViewModel) architektúrára épül (lásd 4.2 ábra) amely a Retrofit könyvtárcsomagon keresztül kommunikál a Strapi.io API-al. A tanárfelület egy .Net Core technológiával készített MVC (Model-View-Controller) (lásd 4.3 ábra) architektúrára épülő webalkalmazás, amely szintén a Strapi.io API-val kommunikál HTTP kérésekkel, a HttpClient osztály segítségével.

4.2. Használt technológiák és architektúrák

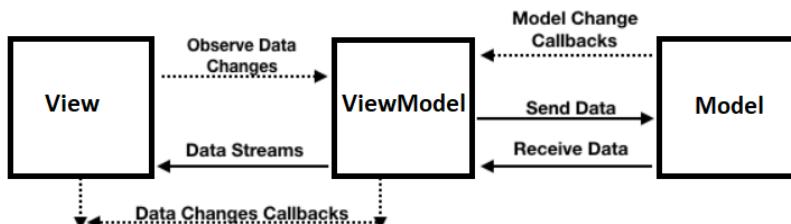
A szoftverem mobilos felületét **Android** operációs rendszerrel fejlesztettem mivel hatalmas felhasználó bázissal rendelkezik, nagyon elterjedt mobil operációs rendszer, véleményem szerint a egyik legmegfelelőbb mód, hogy minél több felhasználóhoz eljusson az applikáció. Programozási nyelvként a **Kotlin** használtam, mivel nagyon könnyű benne tömör és érthető kódot írni és a Java-val is könnyen integrálható Az Android-Kotlin applikációm fejlesztése során az **MVVM** tervezési mintát (design pattern) követtem, mivel könnyen karbantartható, átlátható kódot eredményez és más projektekben is használtam már. Ahogy a 4.4 ábrán is látható az MVVM (Model-View-View Model) három fő komponenset különít el egymástól és kommunikációt hoz létre közöttük. A model tárolja az adatokat, a view felelős a felhasználó felületért és a ViewModel pedig összeköti ezt a kettőt és továbbítja az adatokat.



4.2. ábra. Android Applikáció Architekturája



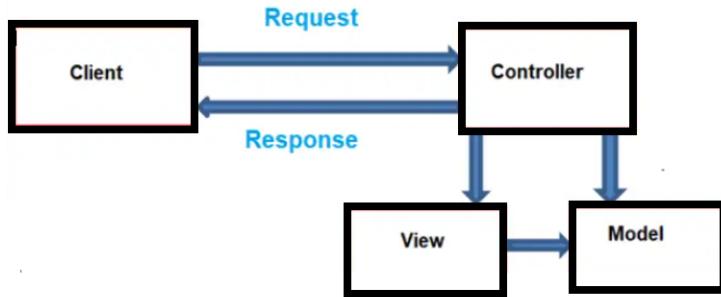
4.3. ábra. Web Alkalmazás Architektúrája



4.4. ábra. MVVM architektúra

A webes alkalmazás készítése során a **.Net Core** keretrendszeret használtam, mivel többféle operációs rendszeren futtatható, magas teljesítménnyel rendelkezik és biztonságos egyben. A fejlesztés során ebben az esetben az **.Net Core** tervezési mintát követtem. Az MVC (Model-View-Controller) az MVVM-hez hasonló tervezési minta (lásd 4.5 ábra), a model és view rétegek hasonló szerepeket töltenek, viszont az MVC esetében a Controller a HTTP kéréseket (GET,POST,UPDATE stb.) kezeli, közben pedig összeköti a másik kettő réteget.

Az adatok kezelésére a **Strapi** egy nyílt forráskódú, fejlett tartalomkezelő rendszer (CMS- Content Management System), amely ingyenes, korlátozások nélkül használható, ami a mai piacon rendkívül ritka. Egy fantasztikus eszköz a modern, gyors API-k készítéséhez, amelyek valós időben frissülnek. Hátterében egy objektum-relációs adatbázis-kezelő rendszer (PostgreSQL) van. A Strapi segítségével rendíkvül egyszerűen és dinamikusan sikerült létrehoznom az alkalmazásomhoz szükséges API-t amellyel minden két alkalmazásom tud kommunikálni.



4.5. ábra. MVC architektúra

4.3. Menedzsment

4.3.1. Arculat megtervezése

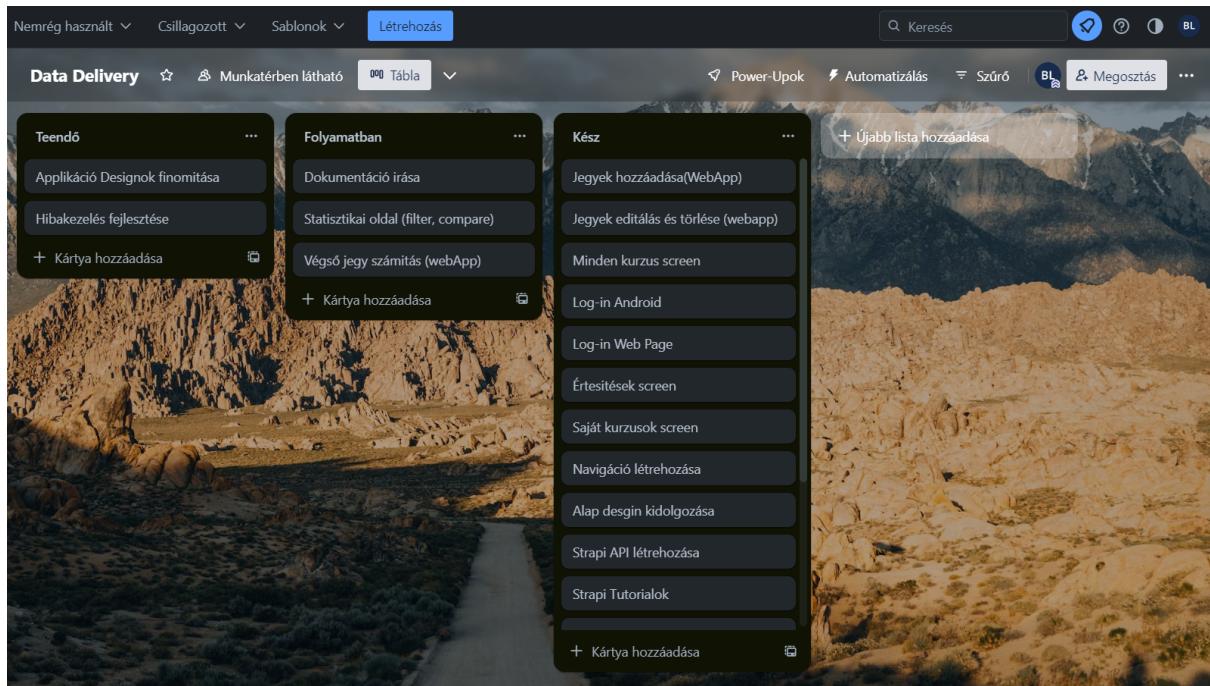
Az applikáció „keretvázát” (wireframe) a LucidChart nevezetű weboldal segítségével sikerült megterveznem, ahogy az a 4.6 ábrán is látható. Ez a terv egy nagyon kezdetleges változat, egy kiinduló pont volt az applikáció fejlesztésében. A fejlesztés során az arculat rengeteget változott viszont ez a terv végig irányadó volt számomra az alap funkcionálitások terén.



4.6. ábra. Arculat-1.változat

4.3.2. Feladatok beosztása

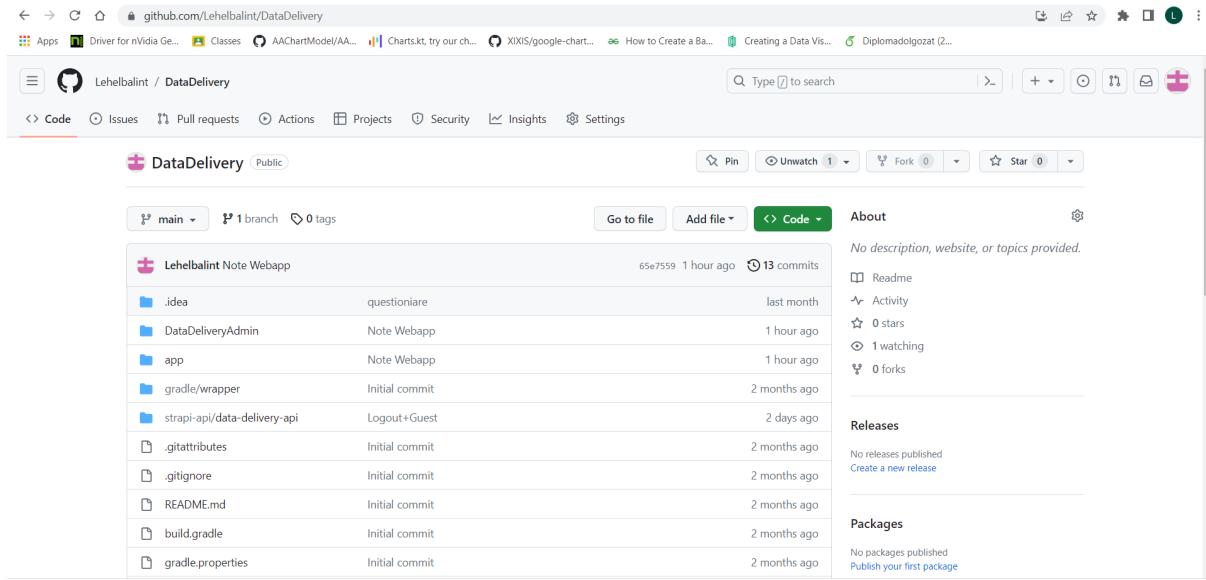
A rendszerem fejlesztése során igyekeztem a teendőimet minél kisebb részfeladatokra felosztani annak érdekében hogy minél jobban tudjam a haladásom követni. A feladatok nyomon követésére a Trello teendőlista-alkalmazást használtam. A trello alkalmazásban az alapvető stratégiát követtem: To do, In Progress, Done (lásd 4.7 ábra).



4.7. ábra. Teendőlista

4.3.3. Verziókövetés

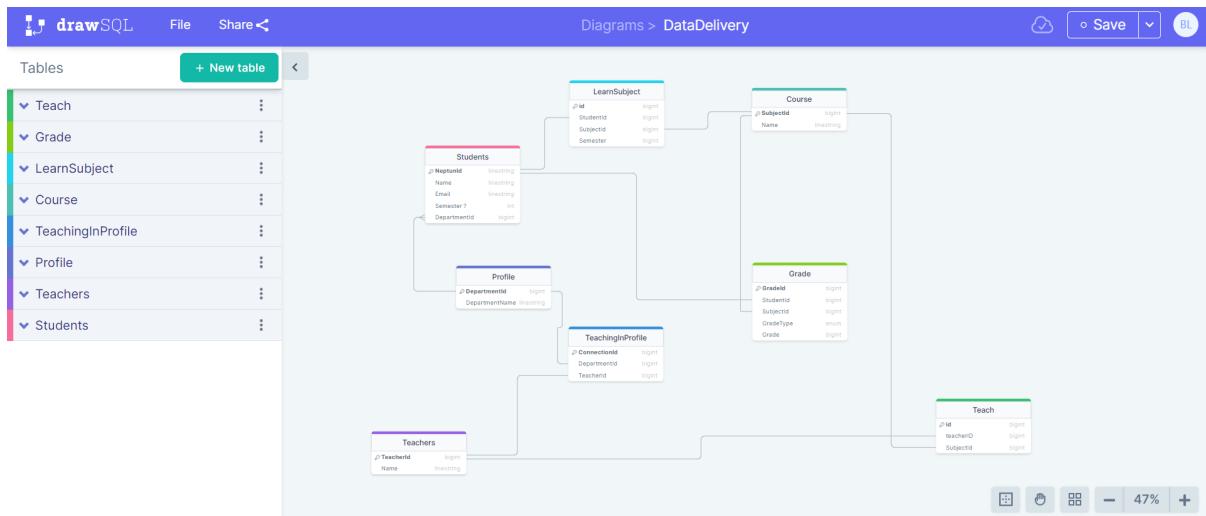
Az alkalmazás fejlesztése során verziókövetéshez a GitHub-ot használtam (lásd 4.8 ábra). A GitHub rendszer használata nagyon hasznos tud lenni még akkor is hogyha önálló szoftverfejlesztést végzünk. Mivel ha az alkalmazás állapotát bármiör feltöltjük akkor később visszatudjuk állítani bármikor ezt a működőképes állapotot, ha esetleg bármi gond történne a fejlesztés során és szükségünk lenne erre. Ezen kívül pedig egy biztonsági mentés forma is, a fejlesztés pedig nem lesz eszközkhöz kötött, mivel internet segítségével bárhonnan elérhető lesz.



4.8. ábra. Github Repository

4.4. Adatbázis és adatkezelés

A rendszer adatainak kezeléséhez a Strapi.io nevű oldalt használtam, amely egy nyílt forráskódú adatkezelő és fejlesztői rendszer. Az adatok tárolására egy SQL alapú adatbázist használtam, a PostgreSQL-t. A PostgreSQL egy relációs adatbázis kezelő rendszer amely rendkívül megbízható, hatékonyan kezeli a nagy adathalmazokat és nyílt forráskódal rendelkezik. Az adatbázis megtervezését, a drawSQL nevezetű oldal segítségével sikerült kivitelezni (lásd 4.9 ábra). A PostgreSQL egyik előnye, hogy nem szükséges N-N-es kapcsolatok esetén külön táblák létrehozása mivel a kapcsolat tárolható az egyed attribútumaként (lásd 4.11 és 4.10 ábra).



4.9. ábra. DrawSQL-Adatbázis terv

The screenshot shows the Strapi Content-Type Builder interface. On the left, a sidebar navigation includes 'Content Manager', 'Content-Type Builder' (which is selected and highlighted in blue), 'Media Library', 'Plugins', 'Marketplace', and 'Settings'. The main area is titled 'Grade' and contains the following information:

- Content-Type Builder**
- COLLECTION TYPES**: Course, Department, Grade (selected), Rating, Student, Teacher, User.
- SINGLE TYPES**: Create new single type.
- COMPONENTS**: Create new component.
- Grade** collection type details:

NAME	TYPE
grade	Number
percentage	Number
final	Boolean
date	Date
course	Relation with Course
student	Relation with Student
teacher	Relation with Teacher

4.10. ábra. Strapi-Grade tábla

The screenshot shows the Strapi Content-Type Builder interface. On the left, a sidebar navigation includes 'Content Manager', 'Content-Type Builder' (selected), 'Media Library', 'Plugins', 'Marketplace', and 'Settings'. The main area is titled 'Course' and contains the following information:

- Content-Type Builder**
- COLLECTION TYPES**: Course (selected), Department, Grade, Rating, Student, Teacher, User.
- SINGLE TYPES**: Create new single type.
- COMPONENTS**: Create new component.
- Course** collection type details:
 - Edit Teachers** (Modal): Refers to a Collection Type. It shows a relationship diagram where 'Course' (Field name: teachers) has a many-to-many relationship with 'Teacher' (Field name: courses). The diagram shows nodes for 'Course' and 'Teacher' with connecting arrows indicating the relationship.

4.11. ábra. Strapi- Reláció létrehozása

Az adatbázist öt főbb tábla alkotja:

- A **Student** tábla tárolja az egyetemen tanuló diákokat és ezek adatait: neptun kód, név, email, jelszó. Ezen kívül három kapcsolata van: a szak táblával (department) minden diáknak van egy szaka, a jegy táblával (grade) egy diáknak több jegye lehet, a kurzus táblával (course) minden diáknak több kurzusa lehet.
- A **Teacher** tábla tárolja az egyemen tanító tanárokat és adataikat: név, email, jelszó. A tanárok két kapcsolattal rendelkeznek: a kurzus táblával (course) minden tanár több kurzust taníthat és egy kurzushoz akár több tanár is tartozhat, a grade (jegy táblával) egy tanár több jegyet adhat.
- A **Grade** tábla tárolja az adatbázisban szereplő jegyeket és a jegyekről részletes információkat: jegy (1-10), százalék (mennyi ér az adott jegy), típus (végső vagy

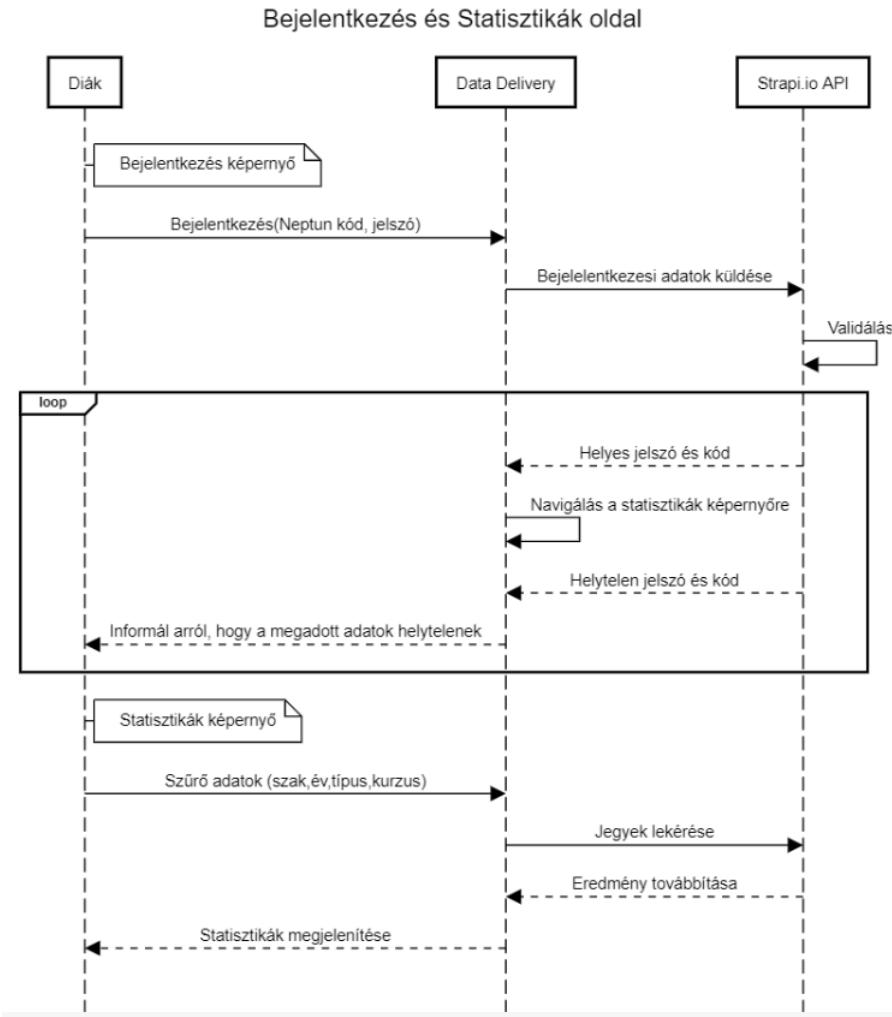
parciális), dátum. minden jegyhez tartozik három reláció: egy jegynek van egy kurzusa amihez tartozik, egy diák akihez tartozik, és egy tanár aki adta az adott jegyet.

- A **Course** tábla tárolja az egyetemi kurzusokat, minden kurzus nevét és két relációt: a tanár (teacher) táblával minden kurzusnak van egy vagy több tanára, a jegy (grades) táblával egy kurzushoz több jegy tartozhat.
- A **Rating** tábla tárolja azokat a válaszokat amelyeket a diákok megadnak a kérdőívekben, minden kérdés rendelkezik egy mezővel. Ezen kívül van két relációja minden kérdőívnak: a diák táblával(student) amely megadja a diákat aki a kérdőívet kitöltötte, a kurzus (course táblával) amely tárolja, hogy melyik kurzusra vonatkozik a kérdőív

A Strapi.io biztosítja minden adatbázishoz a CRUD (Create, Read, Update, Delete) műveleteket az adatokhoz. Ezen kívül a Strapi.io által biztosított API-al segítségével lehet ezeket az adatokat lekérdezni, szűrni, rendezni. Az adatvédelem és biztonság érdek és biztonság számos beépített funkció által biztosított a Strapi.io által. Lehetőséget biztosít a felhasználók regisztrációjára, bejelentkezésére és a tokenek kezelésére. Ezenkívül a Strapi.io rendelkezik jogosultsági funkcióval, lehetővé téve az adatokhoz való hozzáférés szabályozását. Az alkalmazásban különböző szerepkörököt definiál automatikusan (pl. adminisztrátor, szerkesztő, olvasó stb.), és ezekhez különböző jogosultságokat lehet hozzárendelni, ezáltal megszabva, hogy ki mihez férhet hozzá.

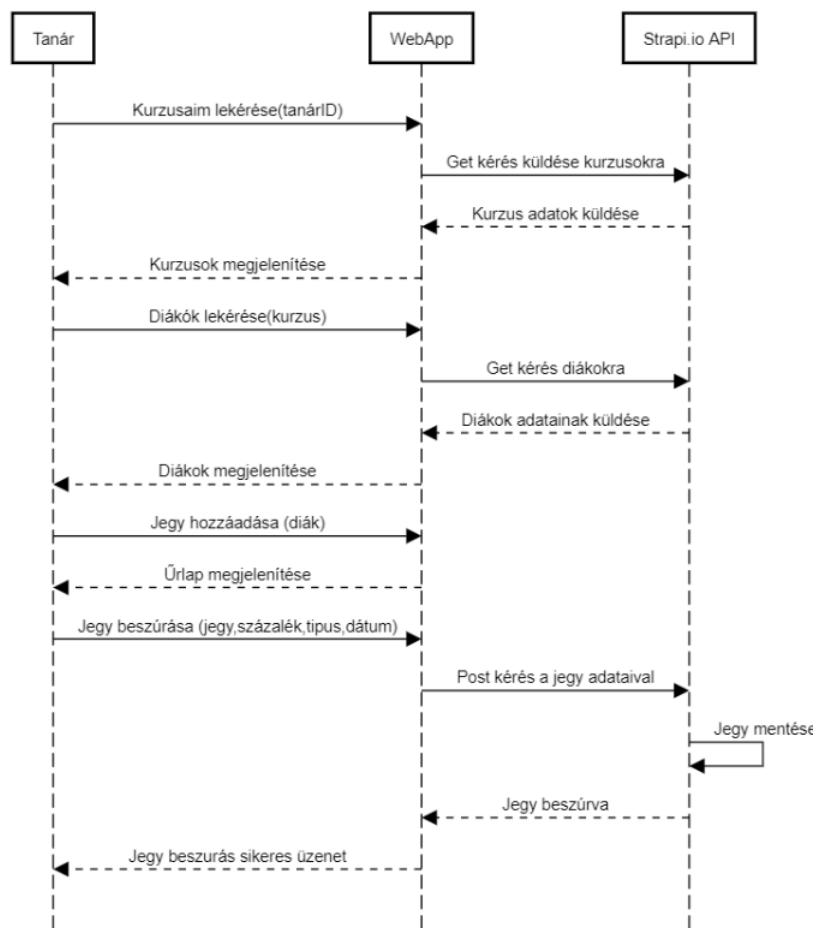
4.5. Szekvencia diagramok

A 4.12 ábrán látható szekvencia diagram megadja bejelentkezési folyamatot majd a statisztikák lekérését a mobilos alkalmazásban. A felhasználó diák megnyitva az alkalmazást, megadja a bejelentkezéshez szükséges adatokat, a neptun kódot és a jelszavát. Az alkalmazás továbbítja a POST kérést az API felé, az API pedig visszaadja a választ. Ha helytelen a jelszó akkor az alkalmazás tudatja ezt a felhasználóval és újra megjeleníti az űrlapot. Ha a jelszó helyes akkor megjelenik a statisztikák képernyő. Ezen a képernyőn a diák megadhat négy különböző szűrőt (szak, év, típus, kurzus), és a filter gombot kiválasztva tudatja az alkalmazással. Az alkalmazás GET kéréssel lekéri a jegyeket az API-tól, majd miután választ kapott kiválasztja a kért adatokat és megjeleníti azokat.



4.12. ábra. Bejelentkezés - Statisztikák lekérése

A következő 4.13 ábra bemutatja, hogy egy tanár felhasználó hogyan tud jegyet adni egy diáknak. A Tanár kiválasztja a tanfolyamok oldalt, ezzel együtt a web applikáció automatikusan GET kérést küld az API felé annak érdekében, hogy megkapja a kurzusokat. Választ kapva kiválasztja az adott tanárhoz tartozó kurzusokat és azokat megjeleníti a felhasználónak. Amikor a tanár egy tanfolyamot kiválaszt az applikáció újra egy GET kérést küld az API felé amely a diákokat kéri le. Miután megkapja a választ hasonló módon a diákokat is megjeleníti. Ha a tanár kiválasztja a jegy hozzáadás opciónt egy adott diáknál akkor az oldal megjelenít egy űrlapot ahol bekéri a jegy adatait. A tanár kitölți az űrlapot ennek következtében az oldal egy POST kérést küld az API-nak ami elmenti a jegyet és választ ad az applikációnak. Az applikáció pedig a választ megjeleníti a felhasználónak.



4.13. ábra. Jegy hozzáadás

5. fejezet

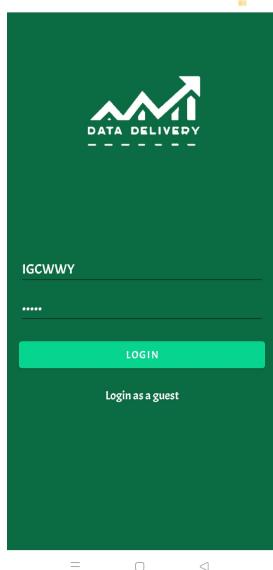
A szoftver bemutatása

A következő fejezetben szeretném bemutatni az általam készített szoftvert (Android alkalmazás és Web alkalmazás). A bemutatás során képernyőképek segítségével szeretném bemutatni a felhasználói felület és a különböző funkciókat, úgy mobil alkalmazást, mint a webes felületet.

5.1. Android alkalmazás

5.1.1. Bejelentkezés

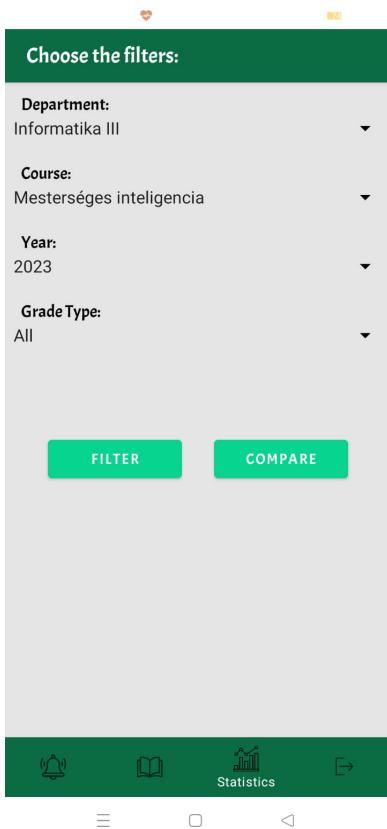
Az alkalmazás megnyitásakor a kezdőképernyővel találjuk szembe magunkat (lásd 5.1 ábra), ahol két művelet elvégzésre van lehetőségünk. Az első a bejelentkezés, amely a diákok kód és a jelszó kombinációval történik meg, a "Login" gombra kattintva .A másik lehetőség pedig a vendékként való belépés az applikációba. Ehhez a lehetőségéhez nincs szükség semmiféle személyes adatra, ezt bárki elérheti, míg a másik lehetőség csak a diákok számára elérhető.



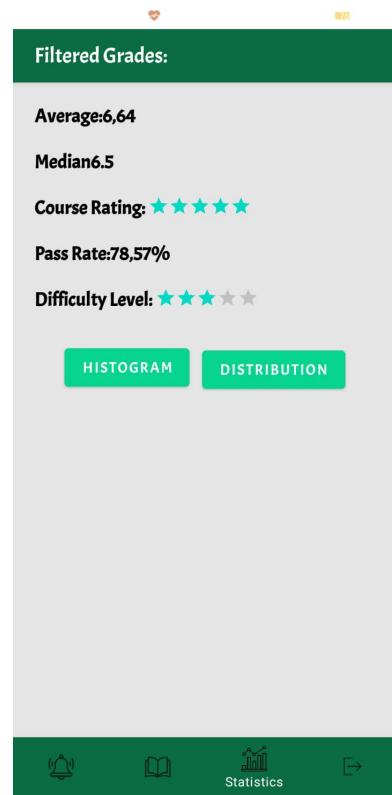
5.1. ábra. Bejelentkezés

5.1.2. Statisztika oldal

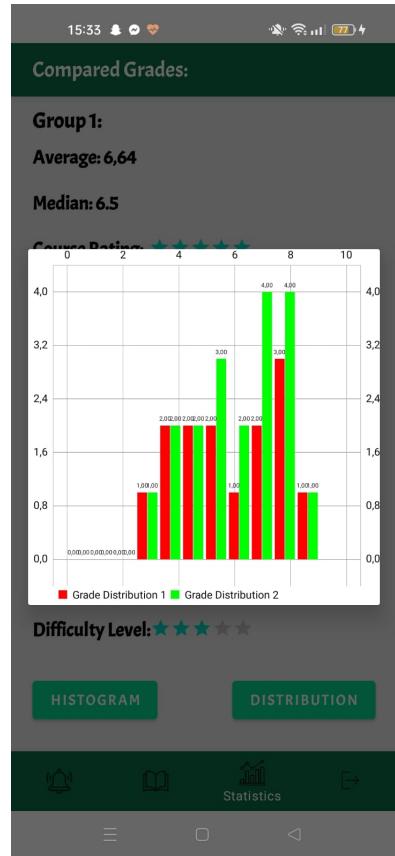
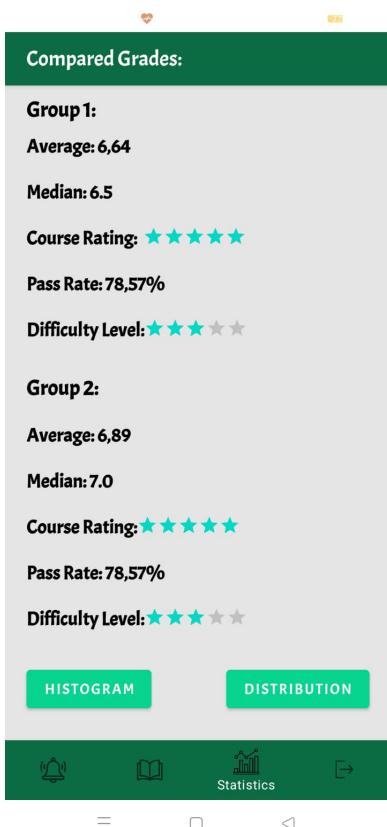
Abban az esetben hogyha vendégként jelentkezünk be, az alkalmazás automatikusan a "Statisztika" oldalra navigálja a felhasználót (lásd 5.2 ábra), ez az egy oldal ami mindenki számára elérhető. Ezen oldalon lehetőségünk nyílik különböző szűrő opciókat (szak, év, kurzus, jegytípus) megadni azokra a jegyekre amelyekből a statisztikai elemzéseket szeretnénk látni. Ezután szintén két választásunk lehet, az első opció a "filter" gombra kattintás amely a kiválasztott filtereket alkalmazva kigenerálja és megjeleníti a statisztikai mutatókat: átlag, medián, nehézségi mutató (ez a kurzusra/kurzusokra vonatkozik), átmenési arány, hisztogram, eloszlás diagram (lásd 5.3 ábra). A második opció a "compare" gomb, amelyre kattintva egy újabb szűrő kerül megjelenítésre. Ennek kitöltésével lehetőségünk nyílik arra, hogy két jegyhalmazt hasonlítsunk össze a fent megemlített mutatok alapján (lásd 5.5 és 5.4 ábra).



5.2. ábra. Szűrő



5.3. ábra. Szűrt adatok

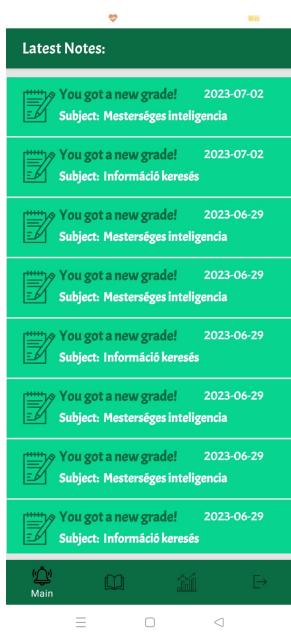


5.4. ábra. Hasonlító szűrő

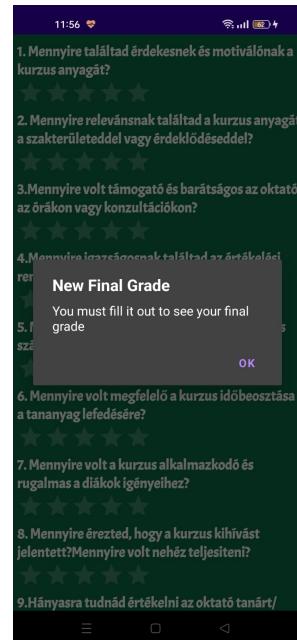
5.5. ábra. Összehasonlítás

5.1.3. Értesítések oldal

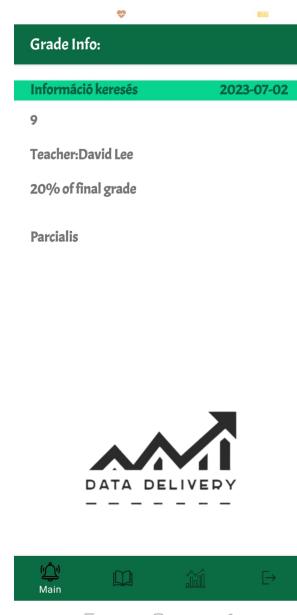
Abban az esetben ha a felhasználó sikeresen bejelentkezett az alkalmazásba, a "Notification" oldra (lásd 5.6 ábra) navigálódik. Ezen az oldalon értesítést kap arról hogyha egy új jegy került ne a rendszerbe. Abban az esetben hogyha ez a jegy, végső jegy és a diákok még nem töltöttek ki azzal a tantárggyal kapcsolatos értékelő kérdőívet, akkor nem tekintheti meg a jegyet hanem egy kérdőív fogadja (lásd 5.7 ábra). Ezen az oldalon a tantárgy értékelésével kapcsolatos kérdésekre kell válaszolni a felhasználónak, 1-től 5-ig pontozva, a válaszok beküldése után átkerül arra az oldalra ahová alapjáraton kerülne hogyha a kérdőív már ki lenne töltve. Ezen az oldalon meg tudja tekinteni az adott jegy részletes leírását, ahogy az a 5.8 ábrán is látható.



5.6. ábra. Értesítések



5.7. ábra. Kérdőív



5.8. ábra. Részletes jegyleírás

5.1.4. Saját tanfolyamok

Ezen az oldalon a felhasználó megtudja tekinteni a saját kurzusait (lásd 5.9 ábra) amelyeken részt vesz vagy már részt vett és teljesítette. Rákattintva egy kurzusra átnavigál az alkalmazás egy újabb oldalra ahol a kiválasztott kurzusról kap különböző információkat és statisztikákat (lásd 5.10 ábra). Ilyen például az összes jegye a tantárgyból, melyik jegy mennyit ér, a jegyek eloszlását a szakon ahol ő is van stb. Visszalépve a "My Courses" oldalra lehetőség nyílik az összes kurzus megtekintésére is (lásd 5.11 ábra). Itt kiválaszt-

va a bármely kurzust, különböző általános statisztikák fognak megjelenni a felhasználó számára, ezen kívül pedig itt találhatóak a kiértékelések eredményei is (lásd 5.12 ábra)

My Courses:

Click here to watch all courses!

- ▶ **Android**
Sarah Johnson
- ▶ **Információ keresés**
David Lee
- ▶ **Mesterséges intelligencia**
David Lee, Benjamin Williams
- ▶ **KAB**
Isabella Martinez

My Courses

Course Detail:

Mesterséges intelligencia

Your grades: 6-20%,
4-30%,
8-25%,
8-25%

Final Grade: 7

YOUR GRADES HISTOGRAM

DEPARTMENT GRADES DISTRIBUTION

Your final grade is better than 66,67% of other students

My Courses

5.9. ábra. Saját kurzusaim

5.10. ábra. Kurzus infók

All Courses:

- ▶ **Android**
Sarah Johnson
- ▶ **Mesterséges intelligencia**
David Lee, Benjamin Williams
- ▶ **KAB**
Isabella Martinez
- ▶ **Információ keresés**
David Lee
- ▶ **C++ Alapok**
David Lee
- ▶ **Adatstrukturák**
Benjamin Williams
- ▶ **Adatbázisok I**
Sarah Johnson, Emma Johnson
- ▶ **Adatbázisok II**
Isabella Martinez
- ▶ **Operációs rendszerek**
David Lee

My Courses

Course Detail:

Mesterséges intelligencia

Average: 7.5
Median: 8.0
Pass Rate: 75.0

Difficulty level: ★★★★★
Interest level: ★★★★★
Teacher friendly level: ★★★★★
Rating system: ★★★★★
Teacher Rating: ★★★★★
Course Rating: ★★★★★

My Courses

5.11. ábra. Összes kurzus

5.12. ábra. Tanfolyam adatai

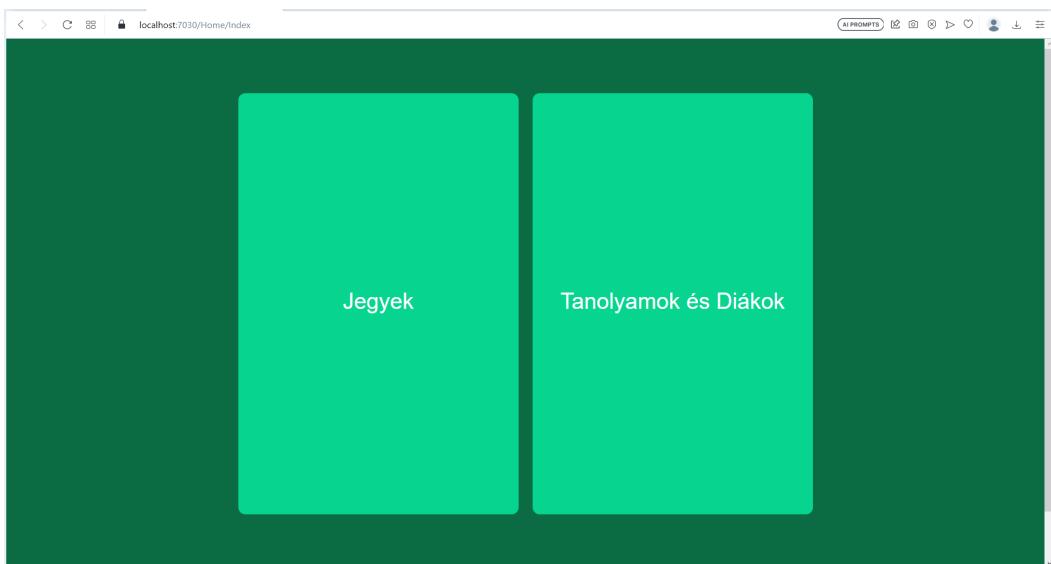
5.2. Web alkalmazás

5.2.1. Bejelentkezés és főoldal

A web alkalmazás elindításakor a bejelentkezés oldal (lásd 5.13 ábra) jelenik meg legelőször ami egy email és egy jelszó kombinációt vár a felhasználótól aki egy egyetemi tanár, mivel másnak nincs hozzáférése ehhez az oldalhoz. Ha sikeres a bejelentkezés akkor egy főoldal jelenik meg amely 2 kártyából áll, mint ahogyan az 5.14 ábrán is látható, ez a két kártya a két főműveletet jelenti amit a tanár felhasználó végezhet.

The screenshot shows a dark-themed login form. At the top center is the title 'Bejelentkezés'. Below it are two input fields: 'Email:' containing 'davidlee@sapientia.ro' and 'Jelszó:' containing '.....'. A green button at the bottom right is labeled 'Bejelentkezés'.

5.13. ábra. Bejelentkezés



5.14. ábra. Fő menü - Kártyák

5.2.2. Jegyek/Minősítések

Ha a "Jegyek" kártyára kattintunk rá akkor a weboldal megjeleníti az összes jegyet amit bejelentkezett tanár adott, rendezve őket időrendi sorrendbe (lásd 5.15 ábra). Lehetőség nyílik a tanár számára, hogy rákeressen jegyekre diákkód és diáknév szerint egyaránt. Az "Edit" gombra kattintva pedig a tanár megtudja változtatni a jegy adatait. A "Delete" gombbal pedig értelemszerűen kitörlődik az adott jegy.

Jegy	Százalék	Dátum	Tanfolyam neve	Diák kódja	Diák neve	
8	30	2023-07-02	Mesterséges intelligencia	AEFAEGE	Christopher Patel	Modósítás Törölés
6	30	2023-07-01	Operációs rendszerek	AEFAEGE	Christopher Patel	Modósítás Törölés
7	100	2023-07-02	Mesterséges intelligencia	IGCWY	Bálint Lehel	Modósítás Törölés
9	20	2023-07-02	Információ keresés	IGCWY	Bálint Lehel	Modósítás Törölés
6	30	2023-06-17	Mesterséges intelligencia	IGCWY2	Elizabeth Rodriguez	Modósítás Törölés
5	30	2023-06-22	Mesterséges intelligencia	IGCWY2	Elizabeth Rodriguez	Modósítás Törölés
8	25	2023-06-29	Mesterséges intelligencia	IGCWY	Bálint Lehel	Modósítás Törölés
8	25	2023-06-29	Mesterséges intelligencia	IGCWY	Bálint Lehel	Modósítás Törölés
4	40	2023-06-29	Információ keresés	IGCWY	Bálint Lehel	Modósítás Törölés
4	30	2023-06-29	Mesterséges intelligencia	IGCWY	Bálint Lehel	Modósítás Törölés
6	20	2023-06-29	Mesterséges intelligencia	IGCWY	Bálint Lehel	Modósítás Törölés
6	10	2023-06-29	Információ keresés	IGCWY	Bálint Lehel	Modósítás Törölés
7	10	2023-06-29	Információ keresés	IGCWY	Bálint Lehel	Modósítás Törölés

5.15. ábra. Tanár által adott jegyek

5.2.3. Tanfolyamok/Diákok

A másik kártya a "Tanfolyamok és Diákok" kártya amelyre kattintva megjelenik az összes kurzus amelyet az adott tanár tanít (lásd 5.16 ábra). Ha kiválaszt egy kurzust megjelennek a diákok akik ezt a kurzust tanulják (lásd 5.17 ábra). Egy diádot kiválasztva pedig jegyet tud adni az adott diáknak (lásd 5.18 ábra) vagy kiszámíthatja végső jegyet (lásd 5.19 ábra) az adott diáknak hogyha megfelelő jegyszámmal rendelkezik a diák.

The screenshot shows a web-based application interface with a sidebar containing various icons. The main content area has a title "Kurzusok". Below it is a table with three columns: "Kurzus név", "Tanárok", and "Kurzus résztvevői". The table contains four rows of data.

Kurzus név	Tanárok	Kurzus résztvevői
Mesterséges intelligencia	David Lee Benjamin Williams	Kurzus résztvevői
Információ keresés	David Lee	Kurzus résztvevői
C++ Alapok	David Lee	Kurzus résztvevői
Operációs rendszerek	David Lee	Kurzus résztvevői

5.16. ábra. Tanár tanfolyamai

The screenshot shows a web-based application interface with a sidebar containing various icons. The main content area has a title "CourseStudents/2". Below it is a table with four columns: "Neptun kód", "Név", "Email", and "Szak". The table contains five rows of data. Each row includes a link labeled "Jegy hozzáadása" and a button labeled "Végző jegy számítás".

Neptun kód	Név	Email	Szak
IGCWWY	Bálint Lehel	balint.lehel@student.ms.sapientia.ro	Informatika III
IGCWWY2	Elizabeth Rodriguez	eli@sapi.ro	Informatika III
AEFAEGER	Christopher Patel	chris@sapi.ro	SzamitasTechnika IV
HTSRS	Emily Nguyen	emi@sapi.ro	Informatika II
Hello	David Miller	aabc@gmail.com	Informatika III

5.17. ábra. Tanfolyamot végző diákok

The screenshot shows a web browser window with the URL `localhost:7030/CourseGrade/AddGrade`. The title bar reads "DataDeliveryAdmin". The main content area has a heading "Jegy hozzáadás". It contains four input fields: "Jegy" (grade) with value "0", "Százalék" (percentage) with value "0", "Dátum" (date) with value "01/01/0001", and "Megjegyzés" (comment). A blue "Küldés" (Send) button is at the bottom.

5.18. ábra. Űrlap jegy adáshoz

The screenshot shows a web browser window with the URL `localhost:7030/CourseGrade/AddFinal`. The title bar reads "DataDeliveryAdmin". The main content area has a heading "Diák jegyei" and a table:

Jegy	Százalék	Dátum
9	40	2023-06-08
5	30	2023-06-22
6	30	2023-06-17

Final grade:
A diákok végső jegye: 6.9

Tanár által adott végső jegy:

Megjegyzés:

A blue "Küldés" (Send) button is at the bottom.

5.19. ábra. Végső jegy számítás

6. fejezet

Limitációk

Az alábbiakban szeretném bemutatni, azokat a korlátozásokat amelyek felmerültek a rendszer fejlesztése során és hatással lehetnek a szoftver működésére és a felhasználói élményre. Ezek lehetnek olyan funkciók vagy megoldások amelyeket nem sikerült megvalósítani vagy nem úgy működnek ahogy az kellene:

- Az alkalmazás internetkapcsolat segítségével kéri le az adatokat az API segítségével. Ennek következtében nem működőképes internetkapcsolat hiányában.
- Jelenleg a rendszer egy egyetem diákjainak az adataira korlátozódik. Nagyobb felhasználói szám estén a teljesítmény csökkenhet.
- Az rendszer jelenleg nem rendelkezik teljeskörű hibakezeléssel így különböző hibák léphetnek fel a használat során.
- Az admin felület amely segítségével diákokat, tanárokat lehet hozzáadni a rendszerhez jelenleg csak a Strapi.io felületén lehetséges, nincs erre külön megoldás implementálva.
- Nincs lehetőség még a jelszó változtatásra vagy jelszó visszaigénylésre csupán a Strapi.io kezelői felületén keresztül.
- Mivel az adatokat API-n keresztül kapja az alkalmazás, ez néha az adatok lassú betöltését és frissítését eredményezheti az API válaszideje vagy a gyenge internetkapcsolat miatt.

Összefoglaló

Dolgozatom során egy olyan elektronikus napló megvalósításával foglalkoztam amely képes egy felsőoktatási intézményben az eredmények követésére és statisztikák megvalósítására valamint a jegyek menedzselésére. Egy rövid bevezető és elméleti megalapozás után bemutattam a szoftverhez tartozó követelményeket, majd a tervezés menetét. Ezután a használt technológiákat, architektúrákkal foglalkoztam, és végül pedig bemutattam a szoftver működését és a felhasználói felületet. Összességében a projektet sikeresnek tekinthetem mivel sikerült megvalósítanom egy működő rendszert főbb részleteit amely képes ellátni egy egyetemi napló szerepét, és egyben statisztikai adatok kimutatására. A megvalósítás fő részletei a következők voltak:

- Egy letisztult, könnyen kezelhető felhasználói felület ami mindenki számára gond nélkül használható legyen, hogy a felhasználók könnyedén tudjanak navigálni a meglévő funkciók között.
- Az alkalmazás által nyújtott funkciók helyes működése, különböző tesztek segítségével.
- Egy webalkalmazás megvalósítása a tanárfelület számára, mivel a tanároknak egy ilyen felületen egyszerűbb a jegyek menedzselése.
- Az adatok helyes és biztonságos tárolása.

Fejlesztési lehetőségek:

A rendszer fejlesztése során az alap funkciókat sikerült megvalósítanom de véleményem szerint mindig van ahová fejlődni, ezért megemlítenék néhány továbbfejlesztési lehetőséget a jövőre nézve. Ilyen például a multiplatformra való terjedés a tanár és a diáksorban esetén egyaránt, a kettő esetleges egyesítése. Egy "Super Admin" felület amely során lehetőség nyílik a felhasználók menedzselésére nem csak a Strapi kezelői felületéről. Esetlegesen órarend, jegyzetek, hírek opciók hozzáadása.

GitHub repository link: <https://github.com/Lehelbalint/DataDelivery>

Ábrák jegyzéke

1.1. Data Delivery Logo	12
2.1. 24edu Catalog - menü	15
2.2. 24edu Catalog - jegyek	15
2.3. Neptun - Felvett vizsgák	15
2.4. School eDiary - Busz követő	16
2.5. School eDiary - Menü	16
3.1. Use Case Diagram	17
4.1. A rendszer architektúrája	21
4.2. Android Applikáció Architekturája	22
4.3. Web Alkalmazás Architektúrája	22
4.4. MVVM architektúra	22
4.5. MVC architektúra	23
4.6. Arculat-1.változat	23
4.7. Teendőlista	24
4.8. Github Repository	25
4.9. DrawSQL-Adatbázis terv	25
4.10. Strapi-Grade tábla	26
4.11. Strapi- Reláció létrehozása	26
4.12. Bejelentkezés - Statisztikák lekérése	28
4.13. Jegy hozzáadás	29
5.1. Bejelentkezés	30
5.2. Szűrő	31
5.3. Szűrt adatok	31
5.4. Hasonlító szűrő	32
5.5. Összehasonlítás	32
5.6. Értesítések	33
5.7. Kérdőív	33
5.8. Részletes jegyleírás	33
5.9. Saját kurzusaim	34
5.10. Kurzus infók	34
5.11. Összes kurzus	34
5.12. Tanfolyam adatai	34
5.13. Bejelentkezés	35
5.14. Fő menü - Kártyák	35

5.15. Tanár által adott jegyek	36
5.16. Tanár tanfolyamai	37
5.17. Tanfolyamatot végző diákok	37
5.18. Śrlap jegy adáshoz	38
5.19. Végső jegy számítás	38

Irodalomjegyzék

- [Cal97] Laurentina Calmorin. *Statistics in Education and the Sciences*. Rex Bookstore, Inc., 1997.
- [JMP18] Sekione R Jeremiah and Joel S Mtebe PhD. Development of digital diary for enhanced parental school involvement in tanzania. *The African Journal of Information Systems*, 10(3):6, 2018.
- [OSD22] Senad Orhani, Emir Saramati, and Lulzim Drini. Electronic school diary for statistical analysis of student progress. *Brazilian Journal of Science*, 1(3):58–65, 2022.
- [Szé16] Katalin Szépvölgyi. ... gondjaink tesznek minket bátrak... 2016.