

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



SAPIENTIA
ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM

EduLane: Kurzusajánló a zsebedben

DIPLOMADOLGOZAT

Témavezetők:
Dr. Osztián Erika,
Egyetemi adjunktus
Osztián Pálma-Rozália,
Egyetemi tanársegéd

Végzős hallgató:
Gábos Alpár

2023

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



UNIVERSITATEA
SAPIENTIA

EduLane: Ghidul cursurilor în buzunarul tău

LUCRARE DE DIPLOMĂ

Coordonator științific:
Dr. Osztián Erika,
Lector universitar

Osztián Pálma-Rozália,
Asistent universitar

Absolvent:
Gábos Alpár

2023

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



SAPIENTIA
HUNGARIAN UNIVERSITY
OF TRANSYLVANIA

EduLane: Course Guide in Your Pocket

BACHELOR THESIS

Scientific advisor:
Dr. Osztián Erika,
Lecturer
Osztián Pálma-Rozália,
Assistant professor

Student:
Gábos Alpár

2023

LUCRARE DE DIPLOMĂ

Coordonator științific:
Lect. univ. dr. Osztian Erika

Candidat: **Gábos Alpár**
Anul absolvirii: **2023**

a) Tema lucrării de licență:

Proiectarea și dezvoltarea unei aplicații de management al cursurilor, care integrează conținutul furnizorilor de cursuri.

b) Problemele principale tratate:

- Realizarea unui studiu asupra aplicațiilor de educație prin cursuri online și identificarea oportunităților potențiale în acest domeniu.
- Examinarea API-urilor publice ale companiilor care oferă cursuri și evaluarea posibilității de a accesa conținutul prin intermediul acestora.
- Proiectarea și dezvoltarea unei aplicații software capabile să proceseze aceste cursuri, să le aducă într-o formă unificată și să le personalizeze în funcție de nevoile utilizatorului.
- Documentarea corespunzătoare a etapelor de proiectare a aplicațiilor.

c) Desene obligatorii:

- Schema bloc a sistemului
- Diagrame de proiectare pentru aplicația software realizată

d) Softuri obligatorii:

Proiectul implică dezvoltarea unei soluții complete pentru gestionarea cursurilor. Aplicația mobilă, construită în tehnologia Kotlin, oferă utilizatorilor posibilitatea de a vizualiza și procesa cursuri direct de pe dispozitivele lor mobile. Un API web, bazat pe tehnologia ASP.NET și oferit prin serviciile Azure, asigură conectivitatea și transferul datelor între aplicația mobilă, interfața web și server. Interfața web este creată specific profesorilor pentru desfășurarea activităților administrative asociate gestionării cursurilor. Această interfață web oferă funcționalități avansate de administrare.

e) Bibliografia recomandată:

- Wulf, J., Blohm, I., Leimeister, J. M., & Brenner, W. (2014). Massive open online courses. Business & Information Systems Engineering, 6, 111-114.
- Kay, J., Barg, M., Fekete, A., Greening, T., Hollands, O., Kingston, J. H., & Crawford, K. (2000). Problem-based learning for foundation computer science courses. Computer Science Education, 10(2), 109-128.
- Cetina, I., Goldbach, D., & Manea, N. (2018). Udemy: a case study in online education and training. Revista Economică, 70(3), 46-54.
- Angulo, M. A., & Aktunc, O. (2019, April). Using GitHub as a teaching tool for programming courses. In 2018 Gulf Southwest Section Conference.

f) Termene obligatorii de consultații: săptămânal, preponderent online

g) Locul și durata practicii: Universitatea „Sapientia” din Cluj-Napoca, Facultatea de Științe Tehnice și Umaniste din Târgu Mureș, laboratorul de informatică, sala 415
Primit tema la data de: 20.05.2022.

Termen de predare: 02.07.2023.

Semnătura Director Departament

Semnătura coordonatorului

Semnătura responsabilului
programului de studiu

Semnătura candidatului

Declarație

Subsemnatul/a GABOS ALPAR, absolvent(ă) al/a specializării INFORMATICA, promoția 2023.... cunoscând prevederile Legii Educației Naționale 1/2011 și a Codului de etică și deontologie profesională a Universității Sapienția 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, TARGU MURES

Data: 15.06.2023

Absolvent

Semnătura..... Gabos

Kivonat

Dolgozatom témája az online tanfolyamokkal történő tanulás egyszerűbbé tétele egy szoftver segítségével. A napjainkban használt modern technológiák segítségével szinte bármi megtanulható online tanfolyamok által. Ez az online tanulási forma egyre népszerűbbé válik, ezáltal napról napra több ilyen tanfolyam jelenik meg a piacon, ami magával hordozza, hogy egyre nehezebb kiválasztani a számunkra igazán relevánsat.

Az online tanfolyamok egyre több szolgáltató kínálatából elérhetőek, viszont ahhoz, hogy megtaláljuk a számunkra ideálisat, több tanulási platformon is körül kell néznünk, ami egy kifejezetten időigényes feladat lehet. Ebben az esetben fontos lehet az, hogy egy adott témában a lehető legtöbb forrásból tudjunk a lehető legrövidebb idő alatt megfelelő tanfolyamot találni.

Jelen dolgozatomban egy olyan szoftvert szeretnék bemutatni, amely ezt a tanfolyam keresési folyamatot képes könnyebbé tenni. Az szoftver létrejöttének fő célja tehát, hogy egy tanulni vágyó egyén a tanfolyamokat egy helyen láthassa, összevonva a korábban említett tanulási platformok különböző tartalmait. Emellett fontos, hogy a szoftver felhasználója képes legyen ezen tanfolyamokat elmenteni és ezáltal nyomon követni a fejlődési folyamatot.

A tervezés és fejlesztés során más hasonló alkalmazások tapasztalatait is figyelembe vettem, hogy új és innovatív funkciókat nyújtsak a felhasználók számára. Az alkalmazás struktúráját, a felhasznált technológiákat, valamint a tanfolyamok elérését és feldolgozását is részletesen bemutatom.

Végül a már elkészült szoftver bemutatására is sor kerül, amely a gyakorlatban is megmutatja, hogy milyen módon kerültek megoldásra korábban felvetett problémák, illetve, hogy milyen funkcionalitásokkal teszi ezt lehetővé a szoftver.

Kulcsszavak: tanfolyam-gyűjtemény, online tanulás, szoftver, technológia, tanulási platform

Rezumat

Tema tezei este ușurarea învățării prin intermediul cursurilor online utilizând un software. Cu ajutorul tehnologiilor moderne utilizate în prezent, aproape orice poate fi învățat prin intermediul cursurilor online. Aceste cursuri devin din ce în ce mai populare, ceea ce duce la apariția zilnică a tot mai multor astfel de cursuri pe piață, ceea ce înseamnă că devine tot mai dificil să alegem pe cel relevant pentru noi.

Cursurile online sunt disponibile din ce în ce mai multe furnizori de servicii, dar pentru a găsi cel ideal pentru noi, trebuie să căutăm în mai multe site-uri ale furnizorilor de cursuri, ceea ce poate fi o sarcină deosebit de consumatoare de timp. În acest caz, poate fi important să găsim un curs pe un anumit subiect din cât mai multe surse posibile în cel mai scurt timp posibil.

În teza mea actuală, doresc să prezint un software care să faciliteze procesul de căutare a cursurilor. Scopul principal al dezvoltării software-ului este, prin urmare, ca o persoană interesată de învățare să poată găsi cursurile într-un singur loc, adunând la un loc diversele cursuri menționate anterior, de pe diferite platforme de învățare. De asemenea, este important că utilizatorul software-ului să poată salva aceste cursuri și să urmărească procesul de dezvoltare.

Experiența altor aplicații similare este, de asemenea, luată în considerare în timpul proiectării și dezvoltării-am cumpărat pentru a oferi utilizatorilor funcții noi și inovații. Structura acestuia, tehnologiile utilizate, precum și accesul și procesarea cursurilor sunt prezentate în detaliu.

În final, software-ul finalizat va fi prezentat, demonstrând și în practică modul în care problemele ridicate anterior au fost rezolvate și funcționalitățile pe care le permite software-ul.

Cuvinte cheie: colecție de cursuri, învățare online, software, tehnologie, platformă de învățare

Abstract

The topic of my thesis is to simplify learning through online courses using software. With the help of modern technologies used today, almost anything can be learned through online courses. This form of online learning is becoming increasingly popular, resulting in a growing number of such courses available on the market, which makes it increasingly difficult to choose the ones that are truly relevant for us.

Online courses are available from an increasing number of providers. However, in order to find the ideal one for us, we have to explore multiple learning platforms, which can be a time-consuming task. In this case, it is important to be able to find the most suitable course from as many sources as possible in the shortest possible time, particularly on a specific topic.

In my thesis, I aim to present a software that can facilitate this course search process. The main purpose of developing the software is to allow individuals who want to learn to see the courses in one place, combining various courses from the mentioned learning platforms. Additionally, it is important that the software's users are able to save these courses and track their progress.

During the design and development process, I have also taken into account the experiences of other similar applications in order to provide new and innovative features to users. The thesis will provide a detailed presentation of the application's structure, the technologies used, and the access and processing of the courses.

Finally, the completed software will be presented, demonstrating in practice how previously identified problems were solved and what functionalities the software enables.

Keywords: course collection, online learning, software, technology, learning platform

Tartalomjegyzék

1. Bevezető	11
1.1. Célkitűzések	12
2. Elméleti megalapozás	13
2.1. Szakirodalmi áttekintés	13
2.2. Hasonló alkalmazások	14
3. Követelmény specifikáció	16
3.1. Felhasználói követelmények	16
3.2. Adminisztrátori követelmények	17
3.3. Rendszerkövetelmények	17
3.3.1. Funkcionális rendszerkövetelmények	18
3.3.2. Nem-funkcionális rendszerkövetelmények	18
4. Programok, technológiák bemutatása	19
4.1. Adatbázis	19
4.2. API	19
4.2.1. Publikus API-k	20
4.2.2. Saját API	20
4.3. Mobil applikáció	21
4.4. Webes felület	21
5. Fejlesztés folyamata	22
5.1. GitLab	22
6. Szoftver	25
6.1. A szoftver bemutatása	25
6.2. Architektúra	25
6.2.1. Mobil alkalmazás	27
6.2.2. Admin felület	29
6.2.3. Adatbázis	30
6.3. Lokalizáció	32
6.4. Diagramok	33
6.4.1. Osztálydiagramok	33
6.4.2. Use Case diagramok	35
6.4.3. Szekvencia diagramok	38
6.5. API Végpontok	40

7. EduLane - az alkalmazás működése	44
7.1. Mobil applikáció	44
7.1.1. Kezdőképernyő	45
7.1.2. Regisztráció	46
7.1.3. Bejelentkezés	47
7.1.4. Főoldal	48
7.1.5. Tanfolyamok keresése	48
7.1.6. Tanfolyamok kezelése	49
7.1.7. Tanfolyam megjelölése elvégzettként	50
7.1.8. Vendégként való belépés	50
7.1.9. Profil	50
7.2. Webes felület	51
7.2.1. Bejelentkezés	52
7.2.2. Felhasználói profil módosítása	52
7.2.3. Tanfolyam hozzáadása / törlése	52
7.2.4. Felhasználó hozzárendelése felhasználói csoporthoz	53
7.2.5. Tanfolyam hozzárendelése felhasználói csoporthoz	53
7.2.6. Tanfolyam módosítása	54
8. Limitációk és továbbfejlesztési lehetőségek	55
8.1. Limitációk	55
8.2. Továbbfejlesztési lehetőségek	55
8.2.1. Mesterséges intelligencia használata	55
8.2.2. Értékelések megosztása a felhasználók között	56
Összefoglaló	57
Ábrák jegyzéke	59
Táblázatok jegyzéke	60
Irodalomjegyzék	61

1. fejezet

Bevezető

Napjainkban a modern technológia térnyerésével és az internet globális elterjedésével gyakorlatilag végtelen információ áll a rendelkezésünkre. A fejlettebb társadalmakban már majdnem minden ember rendelkezik valamilyen okos eszközzel, ezen eszközökön keresztül pedig nagyon rövid időn belül elérhet minden olyan információt, amire szüksége van. Ezen információk birtokában felmerülhet a kérdés, hogy egyáltalán szükségünk van a tudásra, a fejlődésre így, hogy minden elérhető egy kattintással?

A válasz röviden az, hogy igen, szükségünk van rá. Bár már talán a lexikális tudást kevésbé értékelik annyira a társadalomban, mint évekkel ezelőtt, azért a mai napig szükség van a szakértelemre, a valós tudásra. Viszont az, hogy ezt a tudást hogyan szerezzük meg, az már kevésbé egyértelmű.

A 21. században a klasszikus értelemben vett tanulási folyamatokat egyre inkább kezdték átvenni a különböző online tanfolyamok. Ezen tanfolyamok legtöbbször videó anyaggal vannak ellátva és lépésről lépésre - a választott nehézség szerint - adják át az információt a tanulónak és segítenek kialakítani egy összképet az adott témáról.

Számos tanfolyam fellelhető már az interneten, léteznek erre szakosodott vállalkozások, amelyek minőségi és elismert oktatók által biztosítanak tanfolyamokat, ingyenes vagy fizetős módon. Viszont ezek mivel piaci alapon működnek, így próbálnak a lehető legjobb módon elkülönülni egymástól, ami nem könnyíti meg egy tanulni vágyó személy mindennapjait.

Mint egyetemi hallgató, kiemelten fontos számomra, hogy hatékonyan és releváns információkkal bővítssem a tudásom. Emiatt nagyon hasznosnak vélem azokat az online tanfolyamokat, amelyek képesek kiegészíteni az egyetemen megszerzett tudást, és olyan dolgokat mutatnak meg nekem, amikre esetleg az egyetemi órák anyaga és a laborfeladatok elvégzése mellett nem figyeltem fel. Ugyanakkor gyakran szembesülök azzal, hogy nagyon sok időt vesz el ezen tanfolyamok keresése, hiszen a különféle tanfolyam szolgáltatók weboldalait böngészve elég könnyen el tud tévedni az ember.

Épp ezért felmerült a kérdés, hogy mégis hogyan lehetne ezeket a tanfolyamokat egy fedél alá hozni? Hogyan lehetne az internet végtelen böngészése helyett egy helyről elérni mindent, amire egy felhasználónak szüksége van? Ezen kérdések megválaszolására szolgál az az ötlet, miszerint egy applikáció keretein belül el lehetne érni könnyedén ezeket a tanfolyamokat mindenféle kötöttség nélkül.

1.1. Célkitűzések

Ezen ötlet kiteljesedésére különböző célokat tűztem ki magam elé. A fő célkitűzéseim között szerepelt egy olyan alkalmazás létrehozása, ahol a felhasználó bátran böngészhet a tanfolyamok között, kiválaszthatja kedvenceit és követni tudja, hogy melyiket végezte már el. Viszont, ahogy a bevezető részben is említettem, az egyik legfontosabb az volt, hogy egy helyre hozzam össze ezeket a tanfolyamokat, amelyek különböző tanfolyam oldalakról származnak, illetve adjak azoknak egy olyan egységes formát, ami az alkalmazás felhasználója számára egyértelművé teszi, hogy mit is kell tudnia arról a tanfolyamról.

Számos alkalmazás van jelen a piacon, ami hasonló szerepet tölt be, ezek közül többet is kipróbáltam az applikáció elkészítése előtt. Ezen applikációk felhasználása közben igyekeztem a legjobb tulajdonságokat és a hiányosságokat is összegyűjteni, így az egyik fő célom az volt, hogy ezen tapasztalatokat is felhasználjam az alkalmazás elkészítése közben, ezáltal valami mást nyújtani, mint ami jelenleg a piacon elérhető.



1.1. ábra. Alkalmazás logo

Az alkalmazás tehát útvonalat kínál a tanulni vágyó felhasználónak, hogy elkezdhesse ezen folyamatot. Így a dolgozathoz készült szoftver az EduLane nevet kapta, amelyből az *Edu* rész az oktatásra, míg a *Lane* rész az ahhoz kijelölt útra utal.

A dolgozat további részében tehát ismertetem ezen alkalmazás fontosabb részeit, a felhasznált technológiákat, az alkalmazás stuktúráját, a fejlesztés folyamatát és a legfőbb funciókat. Továbbá szó lesz arról is, hogy milyen szempontokat vettem figyelembe a korábbi tapasztalataim alapján, vagy épp arról, hogy melyik oldalakról származnak a tanfolyamok, hogyan kerültek azok feldolgozásra és mik voltak a legfőbb nehézségek azok elérésében.

2. fejezet

Elméleti megalapozás

2.1. Szakirodalmi áttekintés

A bevezetőben már elmítésre került, hogy a mai modern társadalomban egyre elterjedtebbé válik a digitális úton történő tanulás. Ahogy Dr. Jochen Wulf és kutatótársai is megállapították [WBLB14], erre a fajta tanulásra kiváló lehetőséget nyújtanak a MOOC-ok (Massive Open Online Courses), amelyek már nagy jelentőséggel bírnak az oktatás területén. A MOOC-ok olyan online tanfolyamok, amelyeket nagyobb tömegek számára hoznak létre és tesznek elérhetővé. Nyitott hozzáféréssel rendelkeznek, így gyakorlatilag bárki csatlakozhat hozzájuk az interneten keresztül. A tananyag strukturált és interaktív módon van bemutatva, lehetőséget nyújtva a résztvevők számára a saját tempójukban történő tanulásra, valamint lehetővé teszik a közösségi együttműködést és az együttműködésen alapuló tanulást.

Bár a rohamos technológiai fejlődés kétségtelenül fókuszba helyezte ezen online tanfolyamokat, azért már több mint húsz évvel ezelőtt is foglalkoztak a tanfolyamok által jelentett kihívásokkal. Judy Kay és kutatótársai megvizsgálták [KBF⁺00], hogy a számítástechnikai alaptanfolyamok milyen különleges kihívásokat jelentenek mind a tanár, mind a tanuló számára. Ezen kihívások közé tartozik az a tény is, hogy több tanfolyamra is jellemző, hogy kifejezetten műszaki fókusszal rendelkeznek, továbbá szintén a kihívásokhoz sorolható, hogy az egyéni tanulásra is komoly befolyással vannak. Ugyanakkor az sem elhanyagolható tény, hogy az alapok kialakítása is szükséges számos területen, amelyek fontos jelentőséggel bírnak a számítástechnikai diplomások számára.

Az online tanulás megjelenése forradalmasította az oktatást, így sok platform jött létre, ami kifejezetten erre specializálódik. Ezek közül is van néhány, ami globális ismertségre tett szert. Iuliana Cetina és kutatótársai az online oktatás és képzés terén egy kiemelkedő szerepet játszó platformot, a [Udemy](#)-t vizsgálták meg a kutatásukban [CGM18], ahol annak előnyei és hátrányai feltérképezése mellett többet között összehasonlítást is végeztek a szegmens más szereplőivel is. Bár a Udemy kétségkívül az egyik legnagyobb ilyen platform, akadnak még hozzá hasonló platform óriások a piacon, mint a [Coursera](#), az [edX](#), vagy épp a [Udacity](#). Ezek mind különböző területeken és különböző minőségű és árú oktatást nyújtanak, így nehéz lenne közülük kiemelni egyet, ami minden igényt kielégítene egy tanulni vágyó egyén számára.

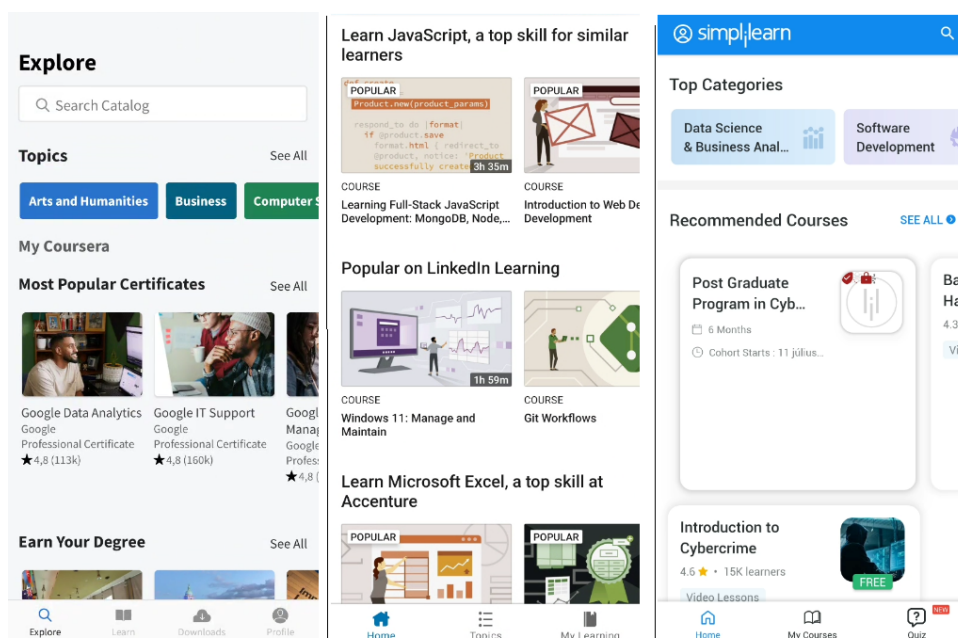
Ugyan a digitális oktatás alatt leggyakrabban a klasszikus online tanfolyamokra gondolunk, ezen felül léteznek még olyan platformok, amik ma már egyre inkább elenged-

hetetlenek az oktatás során. Ahogy Miguel A. Angulo és kutatótársa is rámutat [AA19], egyre gyakrabban használják a [GitHubot](#), mint egy webes kódkezelési és együttműködési platformot az oktatásban. A leggyakoribb felhasználási módja a GitHubnak az oktatásban a hallgatók programozási feladatainak begyűjtése, hiszen a hallgatók könnyen felölthetik és megoszthatják programozási feladataikat a GitHub repository-jukban. Nem utolsósorban pedig a tanulmány arra is utal, hogy a GitHubot az oktatásban a projekt alapú tanfolyamokban való együttműködési környezet létrehozására és a programozási tanfolyamok feladatkezelésének javítására is alkalmazzák.

Összességében elmondható, hogy a digitális oktatás és az online tanulás számos előnyt nyújt a modern társadalomban. A MOOC-ok és más online tanfolyamok lehetővé teszik az egyéni tempóban történő tanulást és a közösségi együttműködést. Emellett a különböző platformok, mint például a GitHub, további lehetőségeket kínálnak az oktatásban és a programozási feladatok kezelésében. A digitális oktatás folyamatosan fejlődik, és további innovációkat és lehetőségeket kínál a tanulás terén.

2.2. Hasonló alkalmazások

Ahogy a bevezetőben is említettem, az alkalmazás elkészítése előtt megnéztem, hogy mégis milyen alkalmazások vannak a piacon jelenleg, amik hasonló célt szolgálnak. Többet is kipróbáltam, ezek között pedig voltak fizetősek és ingyenesek is, viszont több időt az ingyenesekre szántam, így elsősorban azokra alapozok. Ezek közül is azt a hármat emelném ki, amelyek a legnagyobb inspirációt adták a saját alkalmazásomhoz. Mivel nem állítottam fel semmilyen sorrendet ezen alkalmazások között, így a felsorolás sorrendje kizárólag betűrend szerinti felsorolás - ez alapján az alkalmazások nevei a következők: Coursera, LinkedIn Learning és a Simplilearn.



2.1. ábra. Hasonló alkalmazások

Mindhárom alkalmazásban közös, hogy számos tanfolyam közül lehet választani már a főképernyőn (lásd 2.1 ábra) és ezek egymástól függetlenül teljesíthetők - vagyis az alkalmazás nem kötelezi a felhasználót arra, hogy egy tanfolyamot fejezzon be, mielőtt elkezdene egy másikat. Ezen alkalmazások sok lehetőséget biztosítanak egy felhasználó számára, ugyanakkor vannak hiányosságaik is, így ezen tapasztalatokat is felhasználva építettem fel a saját alkalmazásom.

A Coursera és a LinkedIn Learning alkalmazások első használatakor első lépésként szükséges a bejelentkezés - enélkül nem is használható az alkalmazás. Ezzel szemben a Simplilearn alkalmazása használható bejelentkezés nélkül is, így mindenképp egy szempont volt, hogy a saját alkalmazásom is használható legyen előzetes regisztráció nélkül is. Mindhárom alkalmazásnál megfigyelhető volt, hogy már a főoldalon megjelennek különböző tanfolyamok, néhol akár kategorizálva is - ez is abszolút pozitívumként kiemelendő. További nagyon hasznos funkcióként fellelhető a tanfolyamok elmentése, hogy a felhasználó képes legyen elraktározni a számára szimpatikus tanfolyamokat.

Természetesen ezen alkalmazásokat használva felmerültek hiányosságok is, amiket felhasználva jobbra tudnám tenni a saját alkalmazásom. Talán a legfontosabb ezek közül az, hogy ne legyenek lelimitálva a tanfolyamok egy tartalomgyártó termékeire. Bár egy ilyen tanfolyamokat biztosító vállalatnál minden bizonnyal nem engedhető meg, hogy más tanfolyamai is megjelenjenek, mégis ezt tartom talán az egyik legfontosabb szempontnak, hogy minél tágabb adathalmazból legyen képes az alkalmazás meríteni és tudjon a felhasználó választani, hogy épp melyiket szeretné.

3. fejezet

Követelmény specifikáció

Ebben a fejezetben az alkalmazás felhasználói, adminisztrátori és rendszerkövetelményei kerülnek részletes bemutatásra. Megjelennek azok a funkciók és szolgáltatások, amelyek biztosítják, hogy a felhasználó elvárásainak megfelelően az alkalmazás, illetve hogy a rendszer hatékonyan tudjon működni. Ezen felül felvázolom a rendszerrel szemben támasztott funkcionális és nem-funkcionális követelményeket, amelyek a felhasználói élményt, az alkalmazás minél jobb teljesítményét és maximális biztonságát érintik.

3.1. Felhasználói követelmények

A legfontosabb felhasználó követelmények a 3.1 táblázata lettek összefoglalva - bal oldalon a követelmény megnevezése, míg jobb oldalon annak részletezése látható.

Regisztráció	A felhasználónak lehetősége van regisztrálni az alkalmazásba. A regisztrációhoz szükséges adatok: - név, emailcím, jelszó, kor
Bejelentkezés	A felhasználónak lehetősége van bejelentkezni az alkalmazásba. A bejelentkezéshez szükséges adatok: - email - jelszó
Tartalom böngészése	A felhasználó megtekintheti a kilistázott tanfolyamokat, illetve azok részleteit tanfolyamokat kereshet címszó alapján Elmentheti a tanfolyamot a saját listájába Kedvencek közé helyezheti a tanfolyamot Megjelölheti, hogy elvégezte a tanfolyamot - esetenként szükséges elvégezni egy ellenőrző tesztet.
Tanfolyam értékelése	A felhasználó értékelheti az általa elvégzett tanfolyamot

3.1. táblázat. Felhasználói követelmények

3.2. Adminisztrátori követelmények

Az adminisztrátori követelmények mutatnak némi hasonlóságot a felhasználói követelményekkel, viszont itt elvárás, hogy a felhasználónál magasabb szintű cselekvések is végbevihetők legyenek. Ezen követelmények a 3.2 táblázatból olvashatóak ki.

Bejelentkezés	Az adminisztrátornak lehetősége van bejelentkezni az alkalmazásba A bejelentkezéshez szükséges adatok: <ul style="list-style-type: none">- email- jelszó- admin-key
Felhasználók kezelése	Az adminisztrátornak lehetősége van felhasználót hozzáadni Az adminisztrátornak lehetősége van a felhasználó adatait módosítani Továbbá lehetőséget biztosít a felhasználóhoz csatolt tanfolyamok megtekintéséhez, hozzáadásához és eltávolításához
Tanfolyamok kezelése	Az adminisztrátornak lehetősége van kilistázni a tanfolyamokat A meglévő tanfolyamokat módosíthatja / törölheti Lehetőség van új tanfolyamot hozzáadni
Felhasználói csoportok kezelése	Az adminisztrátor megtekintheti a kilistázott csoportokat, illetve azok részleteit Hozzárendelhet felhasználót a csoporthoz Tanfolyamokat rendelhet hozzá / törölhet a csoportokból

3.2. táblázat. Adminisztrátori követelmények

3.3. Rendszerkövetelmények

Egy alkalmazás tervezése és fejlesztése során elengedhetetlen a megfelelő rendszerkövetelmények meghatározása. Ezen követelmények az alkalmazás azon funkcionális és nem-funkcionális jellemzőit határozzák meg, amelyekre annak megfelelően kell reagálnia ahhoz, hogy hatékonyan és megbízhatóan működhessen.

Ahhoz, hogy ezen követelmények a lehető legjobban legyenek meghatározva, mindenképp szükséges a felhasználók és adminisztrátorok igényeit, valamint a rendszerrel támasztott elvárásokat is figyelembe venni. A funkcionális rendszerkövetelmények meghatározzák az alkalmazás felhasználói interakcióit és funkcionalitását, míg a nem-funkcionális rendszerkövetelmények az alkalmazás megbízhatóságát, teljesítményét és biztonságosságát célozzák. A következőkben tehát ezen funkcionális és nem-funkcionális rendszerkövetelmények kerülnek bemutatásra.

Ahhoz, hogy az alkalmazás zökkenőmentesen fusson a felhasználó eszközein, a következő minimum rendszerkövetelményeknek kell megfeleljen:

- Mobil alkalmazás:
 - Android 5+ (Lollipop), min. SDK 21
 - 60 - 70 MB szabad tárhely
 - Internet hozzáférés
 - 1 GB RAM
- Webes felület:
 - Webböngésző
 - Internet hozzáférés

3.3.1. Funkcionális rendszerkövetelmények

Az alkalmazással szemben felállított funkcionális rendszerkövetelmények közül az egyik legfontosabb, hogy támogatnia kell a felhasználók regisztrációját és bejelentkezését. Ez ad lehetőséget az alkalmazás felhasználónkénti személyre szabására. Ahogy az elméleti megalapozás során elhangzott, kiemelt helyen szerepel az is, hogy az alkalmazás lehetőséget biztosítson olyan felhasználóknak is, akik nem szeretnének regisztrálni vagy bejelentkezni.

A felhasználóknak lehetőségük kell legyen arra, hogy böngésszenek a kurzusok között, illetve, hogy kulcsszavak alapján keresni tudjanak ezekre. A kiválasztott tanfolyamok kezelését is biztosítani kell a felhasználó számára, ezalatt legfőképp a mentés, a kedvencek közé helyezés és az elvégzettek közé helyezést jelenti elsősorban. Az elvégzett tanfolyamok után a felhasználónak lehetőséget kell biztosítani arra, hogy értékelje az adott tanfolyamot.

Az adminisztrátoroknak hozzáférést kell biztosítani minden olyan adathoz, amit egy általános felhasználó nem érhet el. A felhasználók fiókjai módosíthatóak kell legyenek, illetve a tanfolyamokhoz köthető olyan tevékenységeket is el kell tudni végezni, mint az új példány hozzáadása, a meglévők módosítása vagy épp törlése.

3.3.2. Nem-funkcionális rendszerkövetelmények

A nem-funkcionális rendszerkövetelmények egyik legfontosabb alapja, hogy az alkalmazásnak felhasználóbarátnak és könnyen kezelhetőnek kell lennie. A felhasználók képesek kell legyenek használni az alkalmazást a lehető legegyszerűbben, segítség nélkül.

Továbbá fontos, hogy az alkalmazás gyorsan és megbízhatóan működjön, ez elengedhetetlen a megfelelő felhasználói élményhez. A felhasználók adatainak védelme és a biztonságos adatkezelés szintén egy kiemelt követelmény az alkalmazással szemben.

Az alkalmazás webes (admin) felülete pedig kompatibilitást kell biztosítson különböző webböngészőkkel és operációs rendszerekkel.

4. fejezet

Programok, technológiák bemutatása

A dolgozat ezen fejezetében bemutatásra kerülnek azok a technológiák és programok, amelyek a projekt részeként felhasználásra kerülnek. Az alkalmazás alapvetően négy különálló részre bontható szét technológiai szempontból, így ezek egyenkénti bemutatása részletes áttekintést nyújt a választott megoldásokról és technológiákról.

4.1. Adatbázis

Az adatbázis egy fontos részét képezi az alkalmazásnak, hiszen itt kerül eltárolásra minden olyan adat, ami nem nyerhető ki valamilyen külső adatbázis igénybevételével - erről az API alfejezetben lesz részletesebben szó. Az én választásom a Firebase Realtime Database lett, ami egy felhőalapú NoSQL adatbázis. Az egyik legfontosabb tulajdonsága, hogy nem táblákban tárolja az adatokat, mint egy SQL adatbázis, hanem JSON formátumban. Ez azt jelenti, hogy az adatokat hierarchikus struktúraként tárolja és kulcs-érték párok segítségével lehet hivatkozni az adatokra.

Ennek a technológiának nagy előnye, hogy valós időben szinkronizálja az adatokat a kapcsolódott kliensek között, ezzel felgyorsítva és leegyszerűsítve a kommunikációs feladatokat. Ugyanakkor a Realtime Database (valós idejű adatbázis) lehetővé teszi azt is, hogy a különböző platformon futó alkalmazások meg tudnak osztani egy valós idejű adatbázist és automatikusan frissítéseket tudnak kapni a legfrissebb adatokról.

Ami szintén a Firebase egyik kiemelt funkcionalitása, az a felhasználókezelés és autentikáció. A Firebase integrált felhasználókezelési rendszert és autentikációs megoldásokat kínál, amelyekkel könnyedén kezelhetők a felhasználói regisztrációk, bejelentkezések és hitelesítések az alkalmazásban.

4.2. API

Az EduLane alkalmazás részei közötti kommunikáció elengedhetetlen eleme az API (Application Programming Interface). Az alkalmazás *backend* része egy ASP.NET Web API, egy olyan keretrendszer, amely lehetővé teszi a RESTful webes szolgáltatások fejlesztését és kiszolgálását, továbbá az API műveletek definiálását, a routingot és az adatkezelést. A kód C# programozási nyelvben lett megírva, a Microsoft Visual Studio 2022

közösségi változatát vettem igénybe ennek létrehozásához, továbbá felhasználtam előre megírt csomagokat is:

- Firebase-hez köthető csomagok:
 - FirebaseAdmin
 - FirebaseAdminAuthentication
 - FirebaseAuthentication.net
 - FireSharp.Serialization.JsonNet
- Token alapú autentikációhoz:
 - Microsoft.AspNet.WebApi.Core
 - Microsoft.AspNetCore.Authentication
 - Microsoft.AspNetCore.Session
- JSON fájlok kezeléséhez:
 - Newtonsoft.Json

4.2.1. Publikus API-k

Mivel az alkalmazás több helyről gyűjti össze a kurzusokat, így különböző szolgáltatók publikus API-ja került felhasználásra és integrálásra. A következő listában ezen API-k szerepelnek:

- [Udemy Affiliate API v2.0](#)
- [YouTube Data API](#)
- [GitHub REST API](#)

4.2.2. Saját API

Ahogy az alfejezet elején is említettem, az alkalmazás saját API-ja végzi a kommunikációt az adatbázis és a mobil applikáció között, továbbá ez kommunikál a külső forrásokkal is - emellett számos más műveletet végez a háttérben, hogy a mobil app és a webes felület a lehetőleg hatékonyabban működhessen.

Viszont ahhoz, hogy ez technikailag is megvalósulhasson, mindenképp szükséges ennek az API-nak a publikálása. Ez a publikálás pedig a Microsoft Azure-on keresztül történik. Az Azure azon túl, hogy biztosítja az API endpoint URL-eket, minden szükséges biztonsági tanúsítvánnyal is rendelkezik. A Web API bármilyen módosítása esetén a publikálás a Visual Studio beépített publikáló mechanizmusán keresztül történik és azonnal megjelenik a *felhőben* is.

Az alkalmazás a következő, állandó API root endpointon keresztül kommunikál:

- [EduLane API](#)

4.3. Mobil applikáció

Az EduLane mobil alkalmazása Androidra lett létrehozva és Kotlin programozási nyelvben íródott. A Kotlin az újabb programozási nyelvek közé tartozik és az Androidra tervezett alkalmazások egyre gyakrabban készülnek ezt a nyelvet használva. Előnyei közé sorolandó, hogy kiválóan működik együtt a Java kódokkal, továbbá bizonyos kód-részek sokkal tömörebben megírhatóak, mint egy Java kód esetében, ezzel is gyorsítva a programozás folyamatát.

Fejlesztői környezetként az Android Studio-t használtam, amely amelelt, hogy gyors és hatékony buildelést valamint futtatást tesz lehetővé, kódkiegészítést is biztosít. A standard nyelvi komponensek mellett a következő csomagok kerültek felhasználásra az alkalmazás fejlesztése során:

- Retrofit: elérést biztosít az API végpontokhoz
- Coroutines: aszinkron hívásokat tud kezelni - az API hívások hatékony és hibamentes futásához szükséges
- Glide: URL alapján képeket tölt be az alkalmazásba
- Viewmodel: képernyők közötti közös adatmegosztásra alkalmas
- LiveData: valós időben lehet figyelni a benne lévő változó tartalmát - aszinkron folyamatokhoz használt

4.4. Webes felület

Az alkalmazás webes felülete az adminisztrátori feladatok elvégzésére lett létrehozva. Egy klasszikus egyoldalas (Single Page Application - SPA) weboldal formájában jött létre. A felhasznált programozási nyelvek a HTML az alapvető dizájnhoz, míg Javascript scriptek formájában szerepelnek a funkcionáltások - ezalatt az összes admin tevékenység értendő.

Az alapvető design az a *w3* nyílt forráskódú elemeket veszi igénybe. Ezen elemek kombinációja és azok testreszabása által jött létre a webes felület alapvető kinézete.

5. fejezet

Fejlesztés folyamata

Egy alkalmazás fejlesztése szempontjából kulcsfontosságú, hogy az egy előre megtervezett folyamat mentén történjen, a lehető legjobban elkülönítve egymástól a különböző részeket. A fejlesztési folyamat alatt ebben az esetben az a folyamat értendő, amely során az alap ötletből a már kész produktum megvalósul, vagyis minden, ami a két véglet között megvalósul - beleértve a tervezést is. Ebben a fejezetben tehát ezen fejlesztés folyamat kerül bemutatásra, hogy milyen alapelvek szerint történ a tervezés és fejlesztés, valamint, hogy ez hogyan került kivitelezésre.

5.1. GitLab

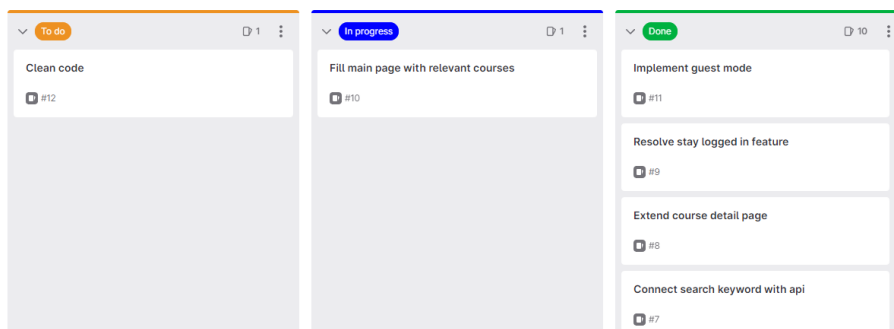
Ahhoz, hogy ez a fejlesztési folyamat zökkenőmentesen végbemehessen, mindenképp szükség volt egy olyan felületre, ahol a tervezés és a fejlesztési lépcsők könnyedén összehangolhatóak. Ennek megfelelően a GitLab szolgáltatását vettem igénybe, amely kiváló lehetőséget biztosít ezen folyamat elvégzésére.

Az alkalmazáshoz két GitLab repository készült, szétválasztva a mobi applikációt az API-tól. Mindkét repository ugyanolyan elvek alapján került felépítésre, ezzel meg a fő cél az volt, hogy az alkalmazás két fő részével egymástól teljesen függetlenül lehessen dolgozni. Ez a két repository biztonságos teret biztosított az alkalmazás tárolására, csak nekem és a felügyelő tanárnak volt hozzáférése ezekhez a tartalmakhoz a fejlesztés során. Azóta viszont a repository-k publikusak lettek, ami a következő linkeken érhetőek el:

- [Mobil app forráskód](#)
- [API forráskód](#)

A tervezési folyamatot és a megfelelő ütemterv tartását nagyban megkönnyítette a GitLab **feladatok (issues)** nyomon követése és hozzárendelése. Ezen belül a GitLab rendelkezis saját táblakezelő felülettel (boards), így egy letisztult vizuális felületen lehet követni, hogy épp hol tart az alkalmazás fejlesztése. A könnyebb áttekinthetőség érdekében három fő kategóriába soroltam a feladatokat a tervezés és kivitelezés során.

Ahogy az 5.1 ábrán is megfigyelhető, a három kategória a *To do*, az *In progress* és a *Done* neveket viseli. A *To do* (teendők) kategóriába kerülnek azok a feladatok, amik

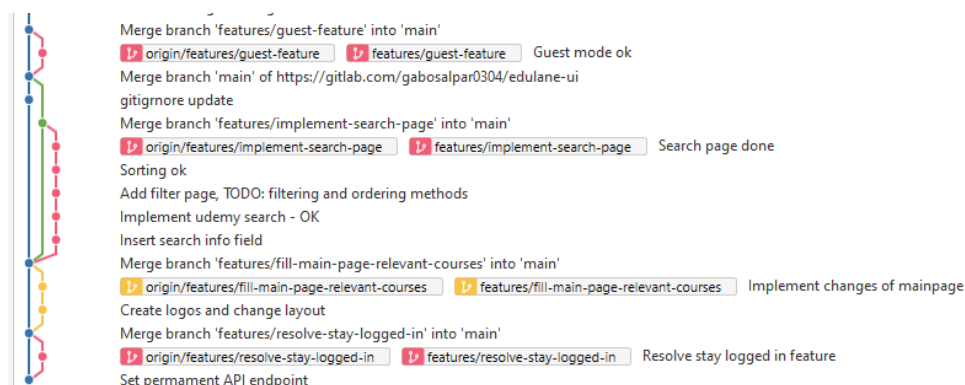


5.1. ábra. Példa GitLab táblákra

még csak tervezés alatt állnak, még nem kezdődött el a munka vele. Az *In progress* (folyamatban lévő) kategóriába pedig azok a feladatok kerülnek, amik már kikerültek a tervezési fázisból és jelenleg épp valamilyen formában azon megy a munka. A harmadik, *Done* (elkészült) kategóriába pedig értelemszerűen azok a feladatok kerülnek, amelyek már az előző két fázison végigmentek és nincs több dolog velük, vagyis befejezettnek tekinthetők.

A GitLab egyik leghasznosabb szolgáltatása, hogy lehetővé teszi a Git alapú **verziókezelést**. Ahogy korábban is említettem, a GitLab az alkalmazás első számú forráskód tárolója, így a forráskódon történő módosítások azonnal felkerülnek ide. Az előbekezdésben említett feladatok elvégzésére ún. ágakat (branch) hoztam létre, így minden módosítás a kódban, ami az adott feladathoz köthető, azon az ágon belül történik. Az verziókezelés három fő szálon fut: a main szál tartalmazza az alkalmazás legaktuálisabb működő verzióját. Innen kerül leválasztásra a *features* ágak, ide kerülnek a hozzáadott új funkcionalitások (pl. *features/implement-search-page*), valamint létezik a *fixes* kategória is, ide kerülnek be a nagyobb hibajavítások.

Ezen ágak közvetlenül összeköthetőek a táblákön szereplő feladatokkal, így néhány kattintással bármikor visszakövethetőek a módosítások. Ez a funkció azon túl, hogy rendszerezettebbé teszi a forráskód fejlődését repository-n belül, lehetőséget biztosít arra is, hogy egy komolyabb hiba esetén egyszerűen vissza lehessen állítani egy olyan korábbi verziót, ami még megfelelően működött.



5.2. ábra. Ágak (branchek) a Sourcetree-ben

A GitLab és a forráskód hatékony átjárhatóságához egy külső alkalmazást vettem igénybe, név szerint a Sourcetree alkalmazását. Ezzel az alkalmazással egy vizuális felületen keresztül könnyedén végrehajthatók az olyan alap műveletek, mint a *Commit*, a *Push* és a *Pull*, valamint a GitLab felületén pedig ezen ágak összefésülése a fő ággal (main branch).

6. fejezet

Szoftver

Ebben a fejezetben a szoftver részletes bemutatására kerül sor. Az elnevezés eredetének a magyarázatától kezdve a különböző fő komponensei létjogosultságának és a köztük lévő kapcsolat fontosságának az alátámasztása is kiemelt hangsúlyt fog kapni. Továbbá a szoftver váza és a funkcionalitások koncepciója különböző diagramok segítségével lesz ábrázolva, valamint az ezekhez szorosan kötődő API végpontok részletes bemutatására is sor kerül.

6.1. A szoftver bemutatása

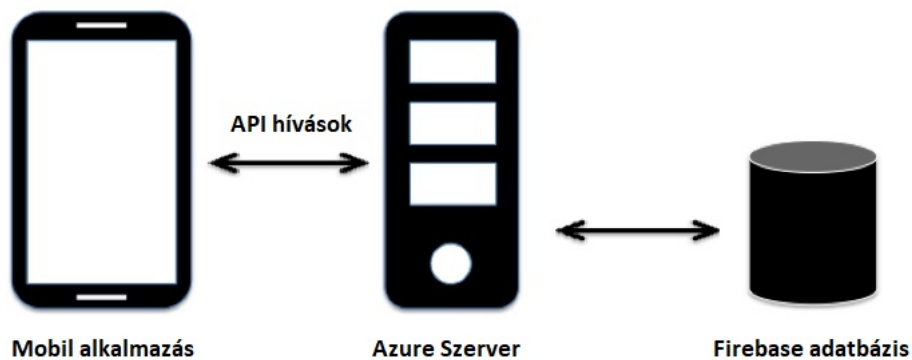
Ahogy a bevezetőben is említésre került, a projekt fő célja az volt, hogy tanulásra alkalmas tanfolyamok legyenek összegyűjtve, szolgáltatótól függetlenül. Mivel a különböző tanfolyam szolgáltatók egyenként eltérnek egymástól mind felhasználói felület, mind a kínált anyagok módszertana szerint, ezért kifejezetten komplikált és a felhasználók számára nem egyértelmű módon történtek volna ezen tanfolyamok feldolgozása.

Emiatt az alkalmazás csupán útvonalat kínál a tanulni vágyó felhasználónak, hogy elkezdhesse ezt a folyamatot. Részben erre utal az alkalmazás neve is, ami a [1.1](#) bevezető fejezetben részletesebben is bemutatásra került.

6.2. Architektúra

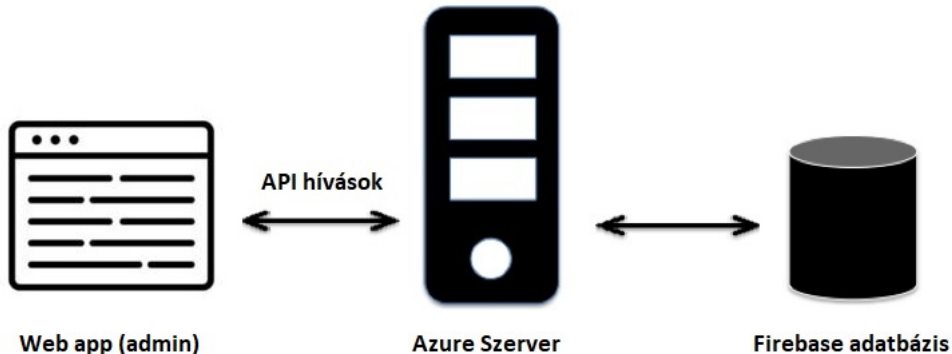
Ahhoz, hogy a felhasználó bárholnan elérje ezt a szolgáltatást, kézenfekvő volt egy mobil applikáció létrehozása, ami egy vizuális felületet biztosít a felhasználónak ezen tanfolyamok megtalálásához. Az app megfelelő működéséhez pedig elengedhetetlen egy olyan backend, ami mindig elérhető és azonnal biztosítani tudja az adatokat az alkalmazás felé, ehhez jött létre az API és az adatbázis is.

A szoftver tehát három fő komponensből tevődik össze, ezen komponensek elnevezésére a továbbiakban a mobil alkalmazás, API és adatbázis elnevezéseket fogom használni. Ezen felül létezik egy negyedik komponens is, az ún. admin felület, amelynek fontos kiégyezítő és szabályozó szerepe van az alkalmazás működésében.



6.1. ábra. Mobil alkalmazás architektúrája

Ahogy a 6.1-es ábrán is látható, az alkalmazás komponensei több lépcsős folyamatként kommunikálnak egymással. A mobil alkalmazásnak nincs közvetlen kapcsolata az adatbázissal, csak API hívásokat hajt végre. Ezen API hívásokat pedig az Azure szerver végzi el, ennek van kapcsolata az adatbázissal, és így képes a benne lévő adatokkal foglalkozni.

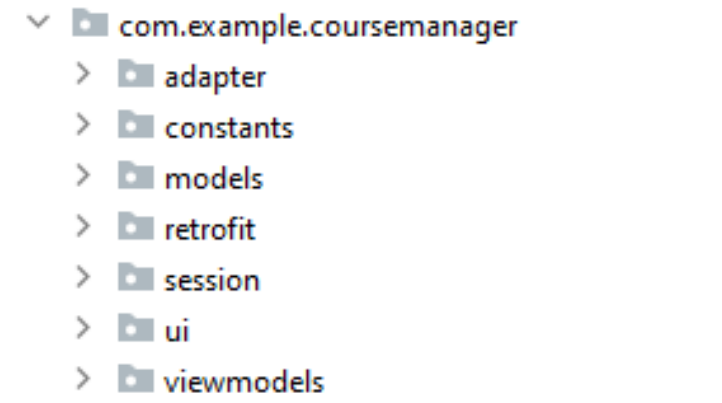


6.2. ábra. Web alkalmazás architektúrája

Hasonlóan a mobil alkalmazáshoz, a webes felület sem kommunikál közvetlenül az adatbázissal, ennek is kizárólag a szerverrel van kapcsolata, szintén az API hívások által. Továbbá fontos kiemelni, hogy a mobil alkalmazás és a webes felület teljesen elkülönül egymástól, vagyis itt sem jöhet létre közvetlen kapcsolat, csupán az adatbázisba tudnak módosításokat végezni (közvetetten, a szerver által) mindketten és azon módosítások csak egy újabb API hívás után válnak elérhetővé.

6.2.1. Mobil alkalmazás

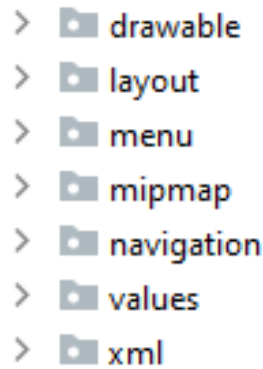
A mobil alkalmazás egy klasszikus Kotlin alkalmazás, amely tartalmazza felépítésben a szükséges komponenseket. Alapvetően két fő részre osztható fel a mobil alkalmazás architektúrája: a *java* mappában található meg minden olyan fájl, ami java alapú, jelen esetben *.kt* kiterjesztésű osztályokat tartalmaz. A másik oldalon pedig az *src* mappa tartalma szerepel, ez a mappa tartalma felel az alkalmazás kinézetéért.



6.3. ábra. A mobil app alapvető csomag struktúrája

A 6.3-as ábra a mobil alkalmazás osztályainak főbb csoportjait jeleníti meg. Ezen jól látható, hogy alapvetően hét kategóriára vannak bontva az osztályok, ezek pedig a következők szerint csoportosulnak:

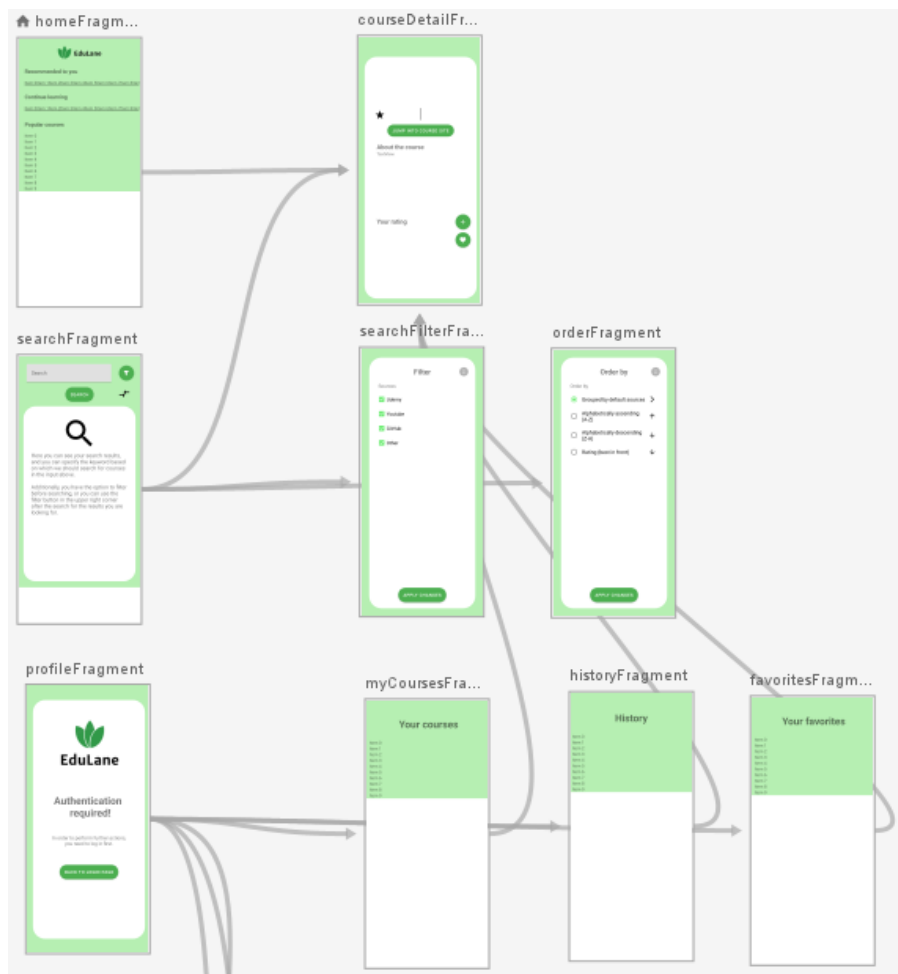
- Adapter: ebben a mappában helyezkednek el a kártyás megjelenítéshez szükséges adapterek - ezek felelnek az adatok megfelelő kitöltéséért egy kártya esetében.
- Constants: ahogy a nevében is szerepel, azok a fájlok kerülhetnek ide, amelyek konstans értékeket tárolnak. Ezek közül a legkézenfekvőbb az API endpointok stringjeinek tárolása, hiszen azok egy fix útvonalon érhetőek el.
- Models: ez a legnagyobb csoportja az osztályoknak. Ide kerül be az összes model, ami valahol az applikáció során felhasználásra kerül - az egyszerű data class-es is ide sorolandók.
- Retrofit: az API hívások kezeléséhez használt fájlok kerülnek ide - az endpointokat tartalmazó interface illetve maga a retrofit instance, amely singletonként szerepel.
- Session: ezek az osztályok felelősek az alkalmazás aktuális futási periódusához csatolt változókért - pl. token
- UI: ebben a mappában van az összes osztály, amelyik valamilyen képernyő felületért felelős - az activityk és a fragmentek osztályai kerülnek be ide
- Viewmodels: azok a Viewmodel típusú osztályok kerülnek be ide, amelyek a felhasználói felület komponensei közötti adatok hatékony átjárásáért felelősek.



6.4. ábra. XML fájlok struktúrája

A mobil applikáció vizuális felületét közvetlenül meghatározó XML fájlok is egy meghatározott struktúrákban szerepelnek. A 6.4-es ábrán szereplő struktúra a következőképpen épül fel:

- **Drawable:** ebbe a kategóriába kerül minden olyan kép vagy ikon, ami az alkalmazás vizuális felületein megtalálható.
- **Layout:** ezen mappa elemei szorosan kötődnek az előző struktúrendszer UI mappa fájljaihoz, hiszen minden fragment vagy activity vizuális felületét ezek határozzák meg.
- **Menu:** ide kerülnek a menü elemek, ezek határozzák meg a navigációs menük kinézetét
- **Mipmap:** a mobil alkalmazás telefonon megjelenő ikonjaiért felelős rész
- **Navigation:** az egyik legfontosabb csoport, ide kerülnek a navigációért felelős XML fájlok, vagyis ezek határozzák meg, hogy az adott képernyőről hová tud lépni a felhasználó - erre egy példa a 6.5-ös ábra.
- **Values:** ide kerülnek az alap stílust és a szövegeket meghatározó XML fájlok, ezek közül az egyik legjelentősebb a strings XML fájl, amely azonosítóhoz csatol megadott szöveget, így elég csak innen módosítani, ha épp valahol egy állandó szöveget át szeretnénk írni. Ugyanakkor ez teszi lehetővé azt is, hogy többnyelvű lehessen az app.
- **XML:** ide kerül az összes többi XML fájl, ami az előzőekhez nem illeszkedik be, ilyenek lehetnek a backup fájlok stb.



6.5. ábra. Navigáció példa

A fenti ábrán (6.5) a mobil alkalmazás az egyik saját belső navigációs gráfjának egy részlete látszik. A téglalapok jelölik a megadott fragmenteket - vagyis az oldalt, amit aktuálisan lát a felhasználó. Ezek a fragmentek feltöltött adatokat nem tartalmaznak, ezért egyfajta vázlatzerűen jelennek meg. A fölöttük lévő szöveg azok elnevezéseket takarja, míg a nyilak mutatják meg, hogy az adott oldalról hová tud navigálni felhasználó a szoftver használása közben.

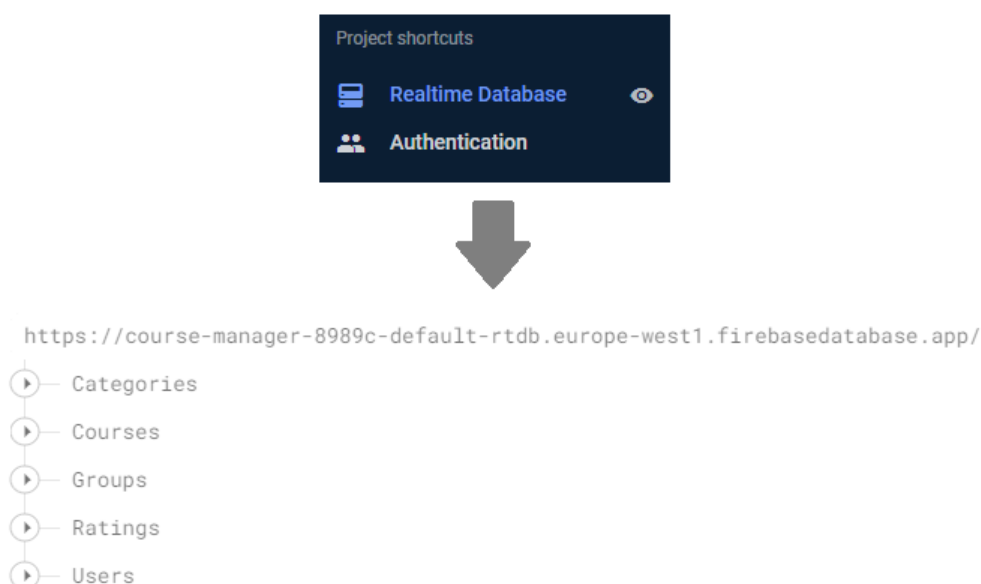
6.2.2. Admin felület

Az admin felület egy egyszerű SPA (Single Page Application), ami egyetlen oldalon képes elvégezni minden tevékenységet. Az admin jogosultságtól függően a webes felület limitálja az oldal láthatóságát. Minden tevékenység ugyanazon API segítségével történik, mint amit a mobil alkalmazás is használ, ez segíti elő, hogy dinamikusan tudjon megjeleníteni elemeket az oldal.

6.2.3. Adatbázis

Az alkalmazás adatbázisa a 4. fejezetben már technikailag bemutatásra került, így az adatbázisbn eltárolt adatok struktúrája fog a főszerepbe kerülni ebben az alfejezetben. Itt fontos kiemelni, hogy két szempontból kell vizsgálni az adatbázist: az eltárolt tanfolyamok - vagyis maga a tartalom -, és a felhasználók bejelentkezéséhez szükséges adatok két külön helyen vannak eltárolva.

Első lépésként tehát nézzük meg a Realtime Database-t, vagyis a felhő alapú adatbázis azon részét, ahol a konkrét tanfolyam adatok kerültek eltárolásra. Mivel a Firebase egy JSON alapú NoSQL adatbázis, így az adatok sem a klasszikus táblás formátumban vannak eltárolva. A 6.6 ábrán tekinthető meg a fő struktúrája az adatbázisnak, vagyis az a négy fő csoport ahová beletartoznak az adatok.



6.6. ábra. Adatbázis struktúra

A négy kategória egymástól teljesen elkülönül, viszont valamilyen szinten mégis kötődnek egymáshoz. Elsőként a Courses kategóriát emelném ki, ahol - ahogy a nevében is benne van - a tanfolyamok kerülnek eltárolásra. Itt a tanfolyam teljes objektumkjént kerül eltárolása, vagyis itt jelenik meg a név, a leírás, a szerző stb. Minden tanfolyam egyéni azonosítóval rendelkezik, amit a Firebase beépített azonosító generátora ad meg neki.

A Categories-ban kerülnek eltárolásra azok a kategóriák, amelyek közös besorolhatóak a tanfolyamok

A Groups kategórián belül találhatóak meg a felhasználói csoportok, aminek a tanfolyam hozzárendelése szempontjából van jelentősége, hiszen itt tárolódnak el, hogy ki melyik csoportban van, és mik vannak hozzárendelve a csoport tagokhoz.

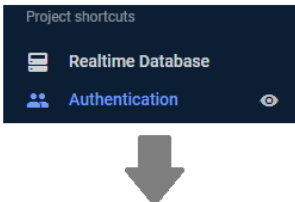
A Rating-en belül kerülnek eltárolásra a tanfolyamokhoz tartozó felhasználói értékelések. Ezek a Courses kategóriában lévő azonosítókhoz társítanak egy értékelés tömböt, vagyis ezen értékelés halmaz átlaga teszi ki egy adott tanfolyam átlag értékelését.






A Users az a kategória, ami az alkalmazás funkcionalitása szempontjából a legfontosabb adatokat tárolja. Azon túl, hogy az általános felhasználói adatok is itt kerülnek eltárolásra, itt szerepelnek az adott felhasználó tanfolyam történetei, tehát hogy melyek az aktív, a kedvenc és a már elvégzett tanfolyamai. Az objektumok egyedi azonosítói pedig megegyeznek a regisztrációkor a Firebase által kigenerált azonosítókkal. Az alábbi ábrán látható egy példa, hogy milyen adatokat is tartalmaz egy eltárolt felhasználó objektuma.



6.7. ábra. Eltárolt felhasználói adatok az adatbázisban

Az adatbázis két fő strukutrális eleme közül a második az autentikációhoz szükséges adatok eltárolása. Mivel a Firebase rendelkezik saját, beépített felhasználó kezeléssel, így az alkalmazáshz is ezt a funkcionalitást vettem igénybe. Egy email és jelszó kombináció alapján tárolja el a felhasználókat, amikhez generál egy egyedi azonosítót - ezek az azonosítók kerülnek be az előzőleg bemutatott Users kategóriába. Továbbá itt lehet módosításokat is elvégezni a felhasználóval, ilyen például a jelszó változtatás.



kd@gmail.com		Dec 30, 2022	Dec 30, 2022	BOOJFNqsMxTpH927YEDS8UMD...
emily@gmail.com		Dec 30, 2022	Dec 30, 2022	msC4TAtWoZ5afbMEaEqIqD5alq2
jrodriguez@gmail.com		Dec 30, 2022	Dec 30, 2022	z3NiIMIRQHU44IEwVvXmun0PY3L2
johnmith@email.com		Dec 30, 2022	Dec 30, 2022	E6gnZrrcU9cl3szVYfj9taP6Scz2
user@example.com		Dec 22, 2022	Jun 21, 2023	JaMzAHaNxrS9NIsGfW98tcqtau32

6.8. ábra. Felhasználói lista

A 6.8 ábrán pedig látható ezen felhasználók közül néhány, email címmel és egyedi azonosítóval lehet őket megkülönböztetni. Továbbá a levél ikon mutatja, hogy milyen típusú a regisztrált felhasználó, itt az összesnél a standard, email és jelszó kombinációval létrehozott profilok jelennek meg.

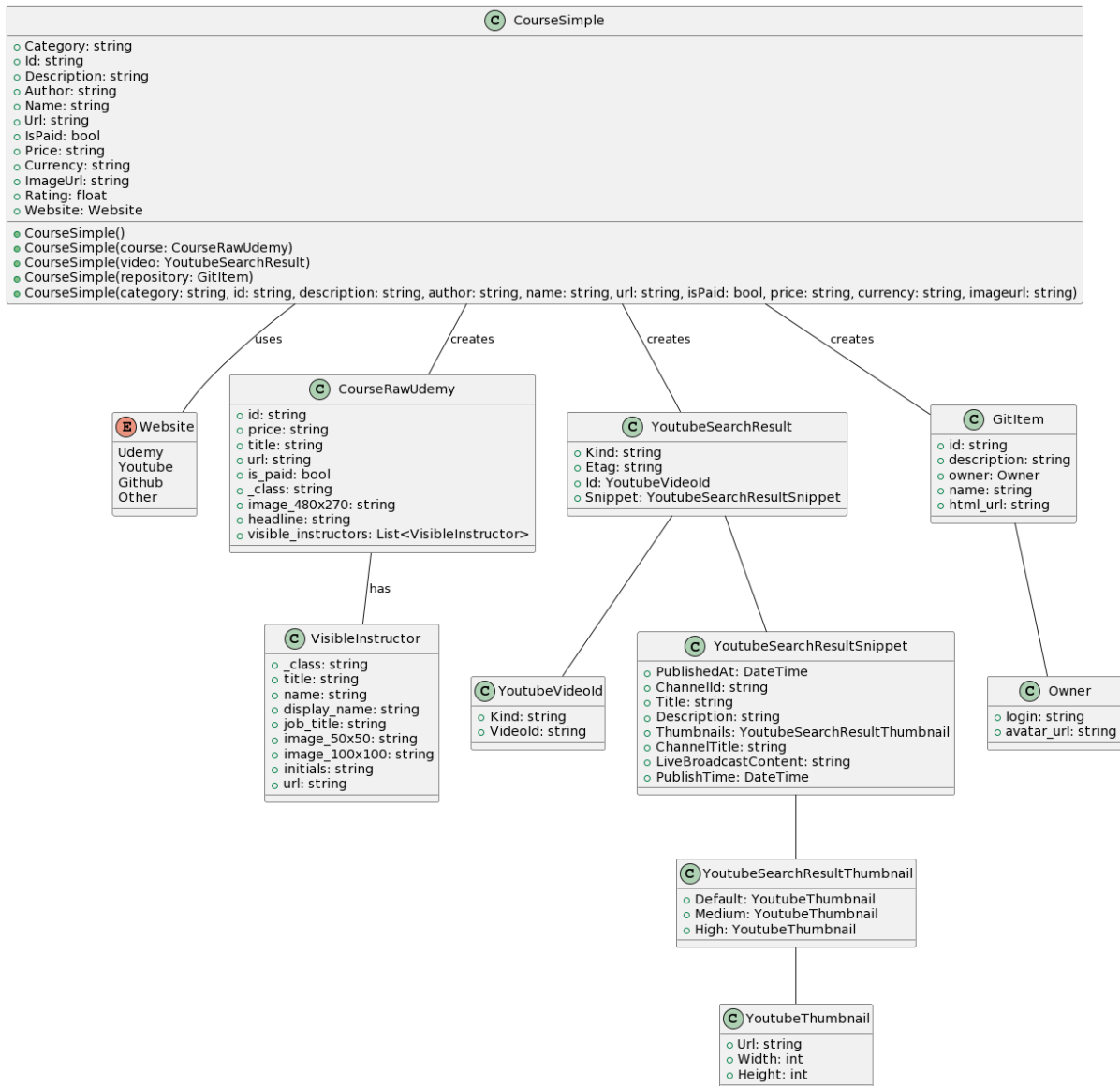
6.3. Lokalizáció

Az alkalmazás támogatja a lokalizációt is, ami lehetővé teszi, hogy az alkalmazás állandó szövegei kulcs-érték párok segítségével kicserélhetőek legyenek. Ezzel megnyitja a lehetőségét annak, hogy az alkalmazás több nyelvű legyen, egyetlen fájl módosításával (a korábban említett strings XML fájl teszi ezt lehetővé). Az alkalmazás egyelőre kizárólag az angol nyelvet támogatja.

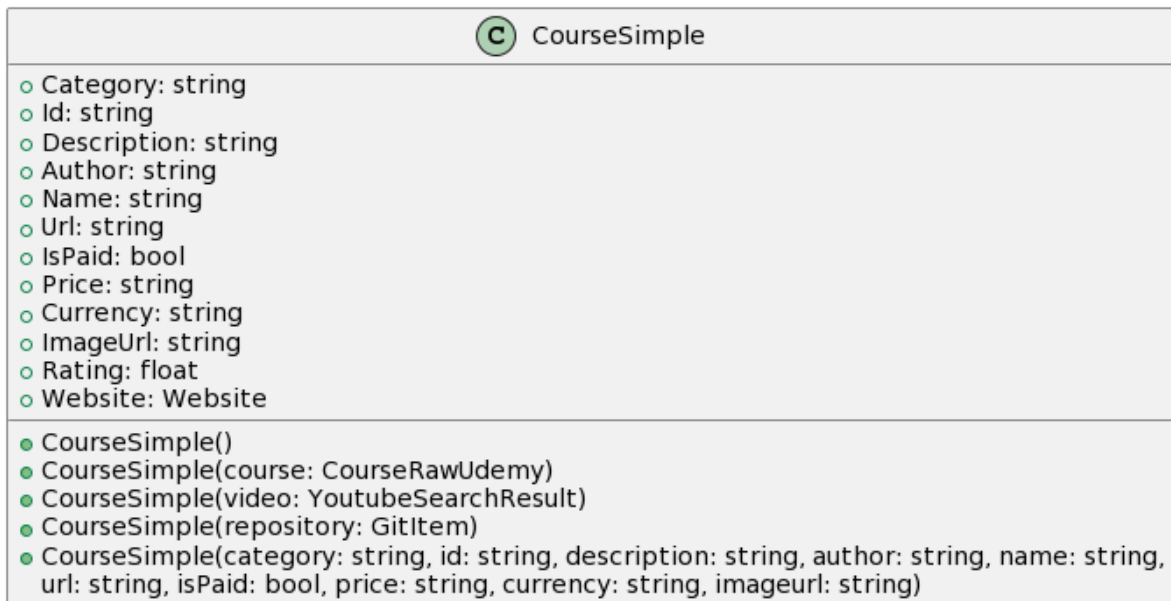
6.4. Diagramok

6.4.1. Osztálydiagramok

A következőkben néhány osztálydiagram kerül bemutatásra. Elsőként az API egyik alap osztálya, a tanfolyam osztály diagramja szerepel (6.9 ábra). Ez az osztály teszi lehetővé, hogy egy egységes formája legyen minden tanfolyamnak.

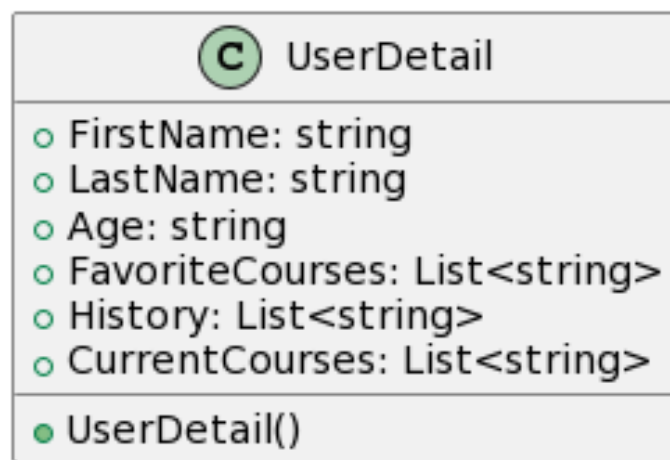


6.9. ábra. Tanfolyam osztálydiagram



6.10. ábra. Egyszerűsített tanfolyam osztálydiagram

Ahogy az a fenti ábrán is megfigyelhető, több konstruktora van az osztálynak, ezzel hidalva át azt a nehézséget, hogy minden tanfolyam szolgáltatótól más formátumban érkezik az adat az API-ról.

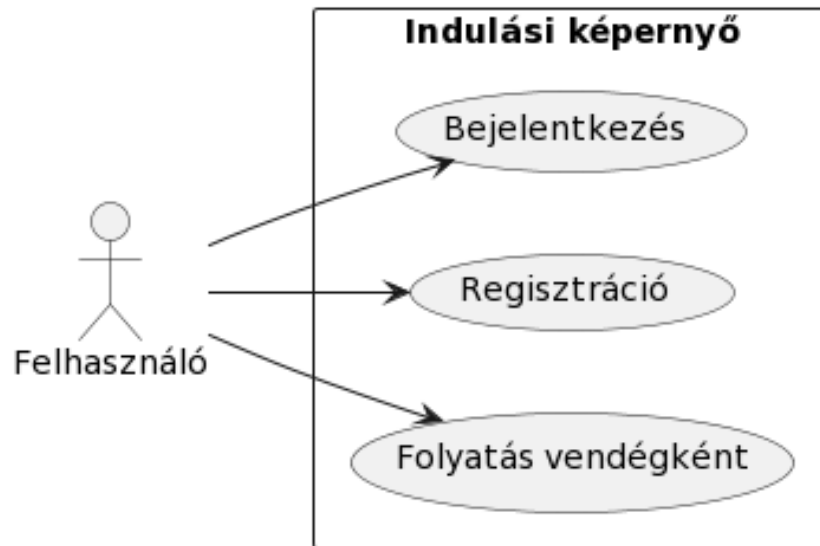


6.11. ábra. Felhasználói profil osztálydiagram

A 6.11 ábrán a felhasználók osztálydiagramja szerepel, ahol megfigyelhető, hogy a tanfolyamok beágyazva szerepelnek sima string listaként, tehát itt csak tanfolyam azonosítók lesznek eltárolva a hatékony tárolás végett - a részletek egyszerűen lekérhetők API hívással a szerverről.

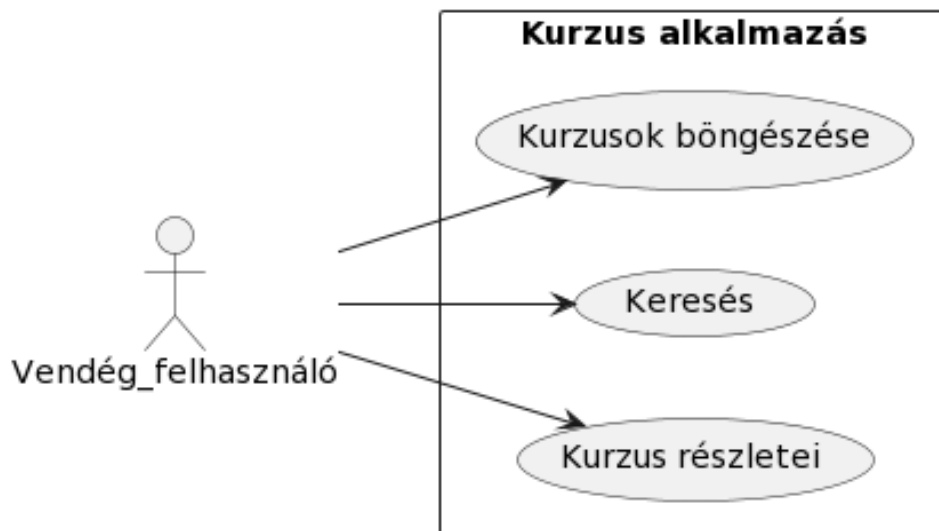
6.4.2. Use Case diagramok

Az alkalmazás indításakor egy felhasználónak egy indító képernyő segítségével a következő tevékenységeket végezheti el (6.12 ábra):



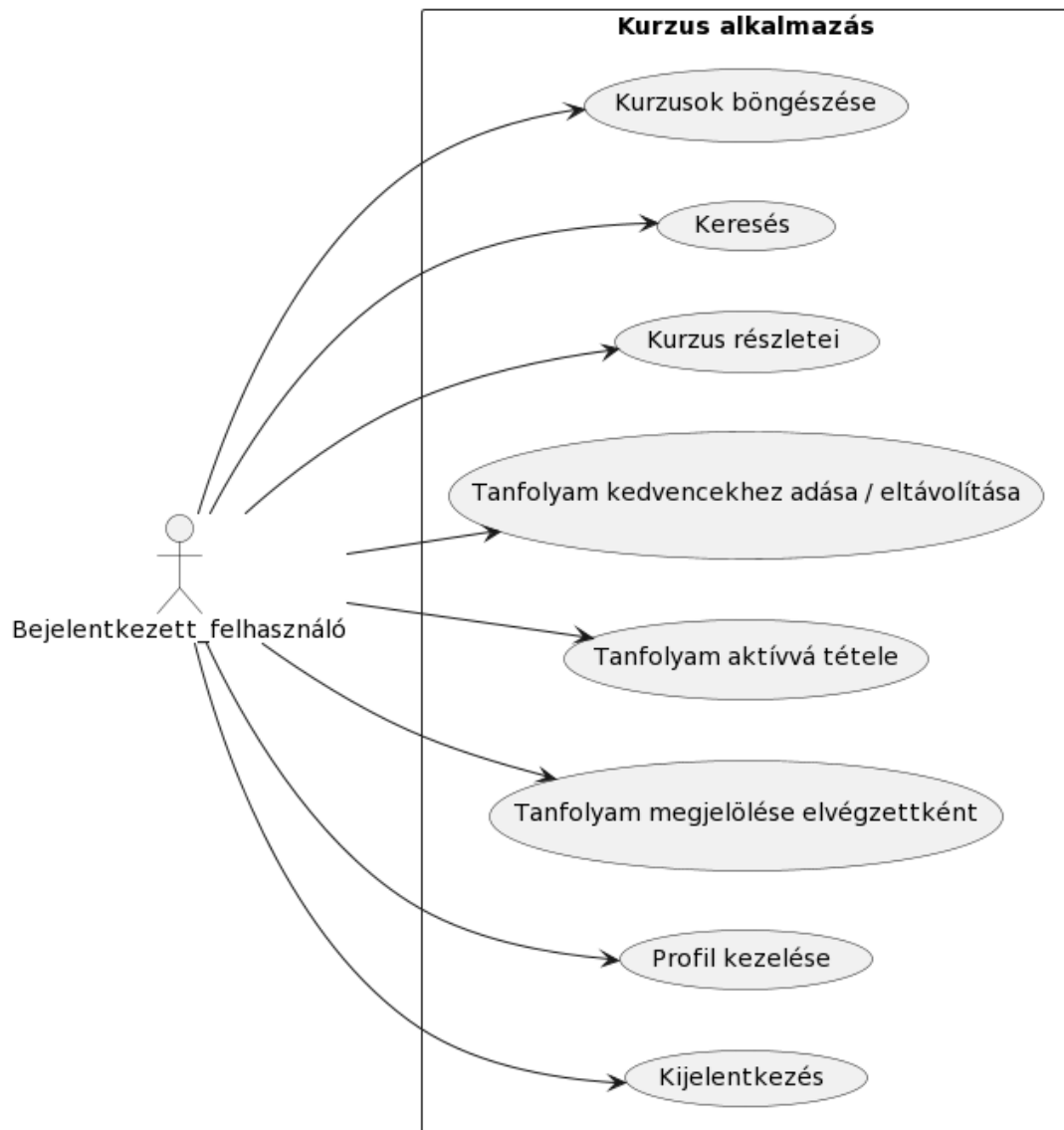
6.12. ábra. Kezdőképernyő az alkalmazás indulásakor

Az kezdőképernyő után a felhasználó nem köteles bejelentkezni, így ha a folytatás vendégként opciót választja, korlátozott lehetőségekkel tudja használni az alkalmazást (6.13):



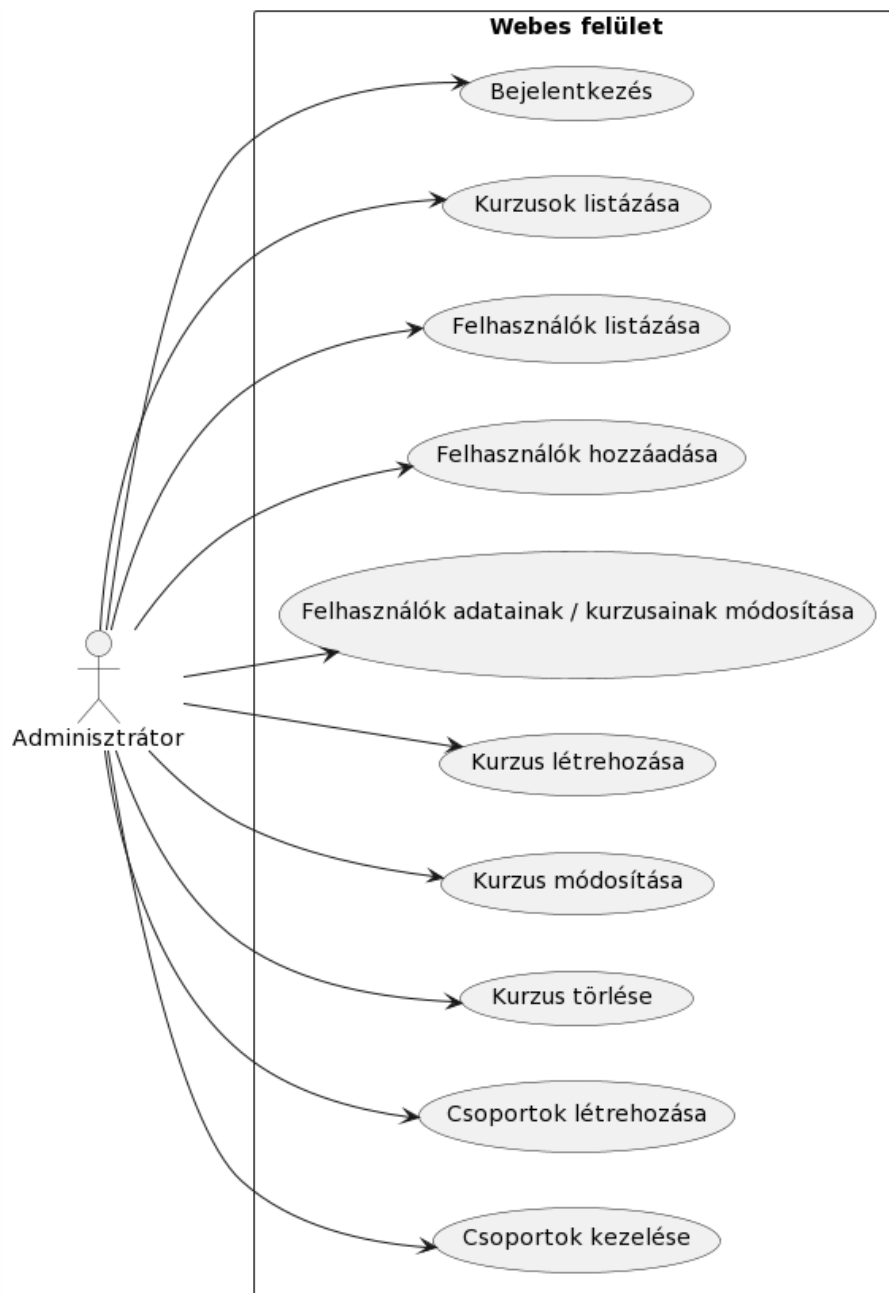
6.13. ábra. Vendég felhasználó

Amennyiben a felhasználó bejelentkezik (vagy a regisztráció után automatikusan bejelentkeztetésre kerül), már sokkal több lehetősége van az alkalmazáson belül, amelyeket a következő ábra szemléltet (6.14):



6.14. ábra. Bejelentkezett felhasználó

Ha az alkalmazásban admin tevékenységet szeretnénk végrehajtani, akkor a webes felületet kell igénybe venni, ahol kizárólag adminként lehet bejelentkezni és emelett még számos feladat végrehajtható (6.15):

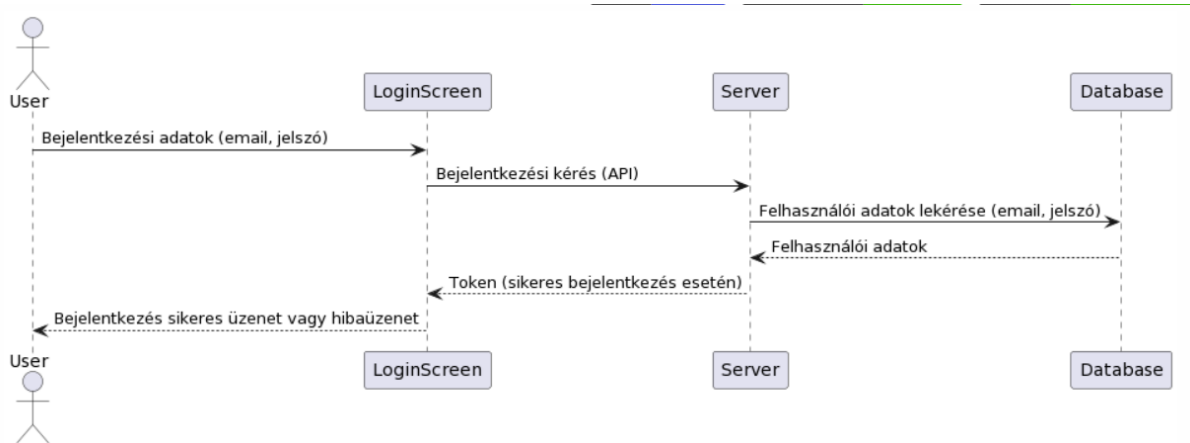


6.15. ábra. Webes felület (adminisztrátor)

6.4.3. Szekvencia diagramok

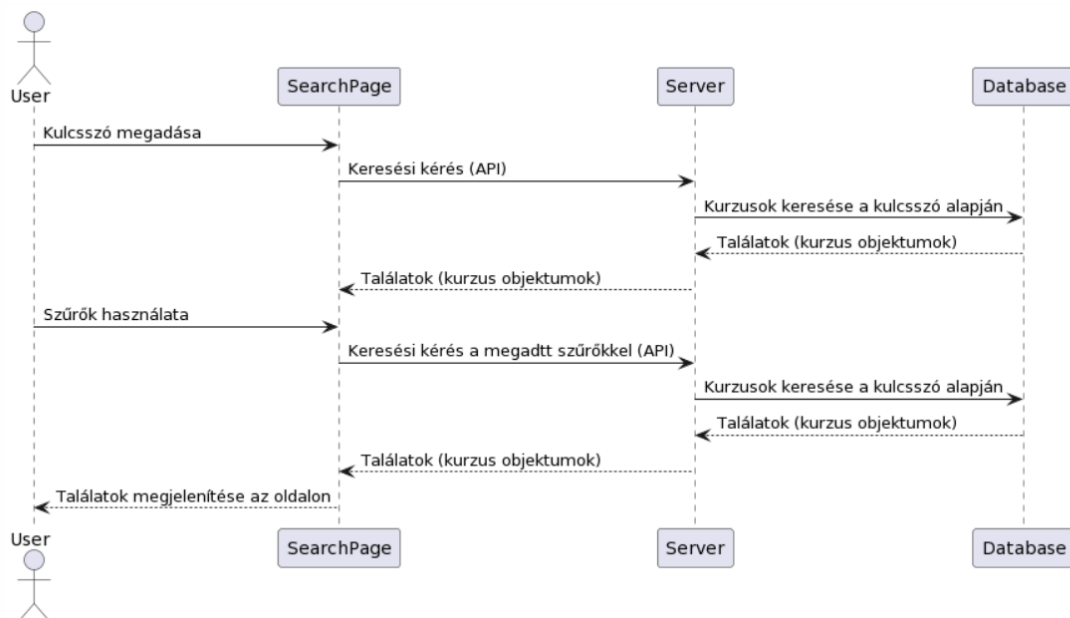
Az alkalmazás folyamatai több lépcsős rendszereken keresztül működnek, így a különböző folyamatok különböző bonyolultsággal futnak le. Ennek prezentálására elkészítettem pár folyamat szekvencia diagramját, amelyek lépésről lépésre megmutatják ezen folyamatok működését.

Elsőként az egyik legkézenfekvőbb folyamat, a bejelentkezés folyamata kerül bemutatásra egy szekvencia diagram segítségével (6.16):



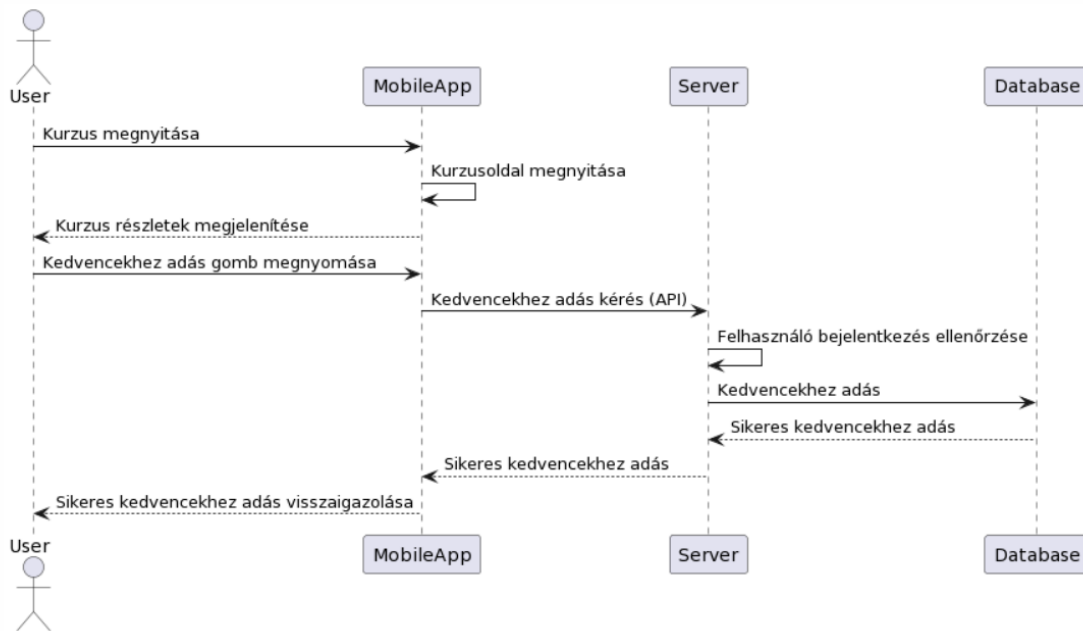
6.16. ábra. Bejelentkezés szekvencia diagram

A keresési folyamat egy minden felhasználó számára elérhető lehetőség, így a következő lépésben az kerül bemutatásra (6.17):



6.17. ábra. Keresés szekvencia diagram

Végül egy bejelentkezett felhasználók számára exklúzív folyamatot választottam, ami már valamennyivel bonyolultabb a korábban bemutatottaknál. Ez a folyamat a tanfolyamok kezeléséhez köthető, még hozzá egy, a felhasználó által kiválasztott és megnyitott tanfolyam kedvencekhez történő adását jelenti, ami a következő diagramon lesz szemléltetve (6.18):



6.18. ábra. Kedvencekhez adás szekvencia diagram

Hasonló módon működik az aktív tanfolyamokhoz való mentés és az elvégzett tanfolyamok közé helyezés folyamata is, csak az annak megfelelő adatbázis listába kerül bele.

6.5. API Végpontok

Az API végpontok a következő linkhez hozzáférhetővé működnek:

[EduLane Server Root URL](#)

- GET /api/course/getAll
 - Funkcionalitás: Az összes tanfolyam lekérdezése
 - Visszaadja az adatbázisban szereplő összes tanfolyam objektumot
- GET /api/course/get/courseId
 - Funkcionalitás: Egy adott tanfolyam lekérdezése az azonosító alapján
 - Adott tanfolyam azonosító alapján visszaadja a hozzá tartozó tanfolyam objektumot
- POST /api/course/insert
 - Funkcionalitás: Új tanfolyam beszúrása az adatbázisba
 - Szükséges bemenet (body): CourseSimple objektum
- PUT /api/course/update/courseId
 - Funkcionalitás: Egy adott tanfolyam frissítése az azonosító alapján
 - Bemenet (body): Tanfolyam objektum
- DELETE /api/course/delete/courseId
 - Funkcionalitás: Egy adott tanfolyam törlése az azonosító alapján
 - (Kizárólag az adatbázisban elmentett tanfolyamokra értendő)
- GET /api/course/check/courseName
 - Funkcionalitás: Egy adott tanfolyam létezésének ellenőrzése a név alapján
 - Válasz: boolean érték
- POST /api/course/insertIfNotExists
 - Funkcionalitás: Új tanfolyam beszúrása, ha nem létezik
 - Bemenet (body): Tanfolyam objektum
 - A külsős tanfolyamok esetében van szerepe, elmenti a saját adatbázisba azt, amit egy felhasználó egyszer igénybe vesz
- POST /api/auth/register
 - Funkcionalitás: Felhasználó regisztrációja
 - Bemenet (body): AuthCreation objektum (felhasználó adatai)
 - Sikeres regisztráció esetén visszatér a hozzá tartozó token-t

- POST /api/auth/adminLogin
 - Funkcionalitás: Adminisztrátor bejelentkezése
 - Bemenet (body): LoginDataAdmin (email, jelszó, kulcs)
 - Sikeres bejelentkezés esetén token-t térít vissza
- POST /api/auth/login
 - Funkcionalitás: Felhasználó bejelentkezése
 - Bemenet (body): LoginData (email, jelszó)
 - Sikeres bejelentkezés esetén token-t térít vissza
- GET /api/user/getAll
 - Funkcionalitás: Az összes felhasználó lekérdezése - token szükséges hozzá
 - Visszatéríti az összes felhasználót egy listaként
- GET /api/user/getAllList
 - Funkcionalitás: Az összes felhasználó listájának lekérdezése - token szükséges hozzá
 - Visszatéríti az összes felhasználóhoz tartozó tanfolyamokat egy azonosító listaként
- GET /api/user/getById
 - Funkcionalitás: Egy adott felhasználó lekérdezése az azonosító alapján - token szükséges hozzá
 - Visszatérít egy felhasználó objektumot, amennyiben az szerepel a listában
- POST /api/user/courses/current/insert
 - Funkcionalitás: Aktuális tanfolyam beszúrása a felhasználóhoz - token szükséges hozzá
 - Hozzáad egy tanfolyamt (ID) az adott azonosítójú felhasználó elmentett aktuális tanfolyamai közé
- POST /api/user/courses/history/insert
 - Funkcionalitás: Előzmény tanfolyam beszúrása a felhasználóhoz - token szükséges hozzá
 - Hozzáad egy tanfolyamt (ID) az adott azonosítójú felhasználó előzmény tanfolyamai közé
- POST /api/user/courses/favorites/insert
 - Funkcionalitás: Kedvenc tanfolyam beszúrása a felhasználóhoz - token szükséges hozzá

- Hozzáad egy tanfolyamot (ID) az adott azonosítójú felhasználó kedvenc tanfolyamai közé
- POST /api/user/courses/history/delete
 - Funkcionalitás: Előzmény tanfolyam törlése a felhasználótól - token szükséges hozzá
 - Sikeres végrehajtás esetén visszatéríti a tanfolyam azonosítót
- POST /api/user/courses/current/delete
 - Funkcionalitás: Aktuális tanfolyam törlése a felhasználótól - token szükséges hozzá
 - Sikeres végrehajtás esetén visszatéríti a tanfolyam azonosítót
- POST /api/user/courses/favorites/delete
 - Funkcionalitás: Kedvenc tanfolyam törlése a felhasználótól - token szükséges hozzá
 - Sikeres végrehajtás esetén visszatéríti a tanfolyam azonosítót
- GET /api/user/courses/favorites/get
 - Funkcionalitás: A felhasználó kedvenc tanfolyamainak lekérdezése - token szükséges hozzá
 - Visszatéríti a felhasználó kedvenc tanfolyamait (ID) listában
- GET /api/user/courses/history/get
 - Funkcionalitás: A felhasználó előzmény tanfolyamainak lekérdezése - token szükséges hozzá
 - Visszatéríti a felhasználó előzmény tanfolyamait (ID) listában
- GET /api/user/courses/current/get
 - Funkcionalitás: A felhasználó aktuális tanfolyamainak lekérdezése - token szükséges hozzá
 - Visszatéríti a felhasználó aktuális tanfolyamait (ID) listában
- POST /api/user/courses/current/move
 - Funkcionalitás: Aktuális tanfolyam áthelyezése az előzményekbe - token szükséges hozzá
 - Áthelyezi a kiválasztott tanfolyamot a felhasználó aktív tanfolyamai közül a felhasználó előzményeibe
- GET /api/udemy/search
 - Funkcionalitás: Keresés az Udemy platformon

- Kulcsszó alapján keres a UdeMy adatbázisában
- A visszatérített érték az egységes tanfolyam objektum listára lesz konvertálva
- GET /api/youtube/search
 - Funkcionalitás: Keresés a YouTube platformon
 - Kulcsszó alapján keres a Youtube adatbázisában
 - A visszatérített érték az egységes tanfolyam objektum listára lesz konvertálva
- GET /api/github/search
 - Funkcionalitás: Keresés a GitHub platformon
 - Kulcsszó alapján keres a GitHub adatbázisában
 - A visszatérített érték az egységes tanfolyam objektum listára lesz konvertálva
- GET /api/groups/get
 - Funkcionalitás: Az összes csoport lekérdezése
 - Lekéri az adatbázisban található összes csoportot
- GET /api/groups/get/userId
 - Funkcionalitás: Csoportok lekérdezése felhasználó azonosító alapján
 - Lekéri a felhasználóhoz tartozó csoportokat az adott felhasználó azonosítója alapján
- POST /api/groups/insert
 - Funkcionalitás: Csoport beszúrása
 - Beszúrja a megadott csoportot az adatbázisba
- POST /api/groups/delete
 - Funkcionalitás: Csoport törlése
 - Törli a megadott csoportot az adatbázisból
- GET /api/rating/get/courseId
 - Funkcionalitás: Értékelés lekérdezése tanfolyam azonosító alapján
 - Lekéri az adott tanfolyamhoz tartozó értékelést a tanfolyam azonosítója alapján
- POST /api/rating/insert
 - Funkcionalitás: Értékelés beszúrása tanfolyam azonosító alapján
 - Beszúrja az adott tanfolyamhoz a megadott értékelést a tanfolyam azonosítója alapján

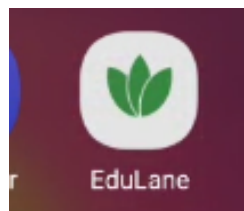
7. fejezet

EduLane - az alkalmazás működése

Ebben a fejezetben az EduLane alkalmazás kerül bemutatásra. A két fő komponens - a mobil applikáció és a webes felület - felhasználói felületei funkcionalitás és oldalak szerint elkülönítve lesznek bemutatva, képernyőfelvételek segítségével felhasználva.

7.1. Mobil applikáció

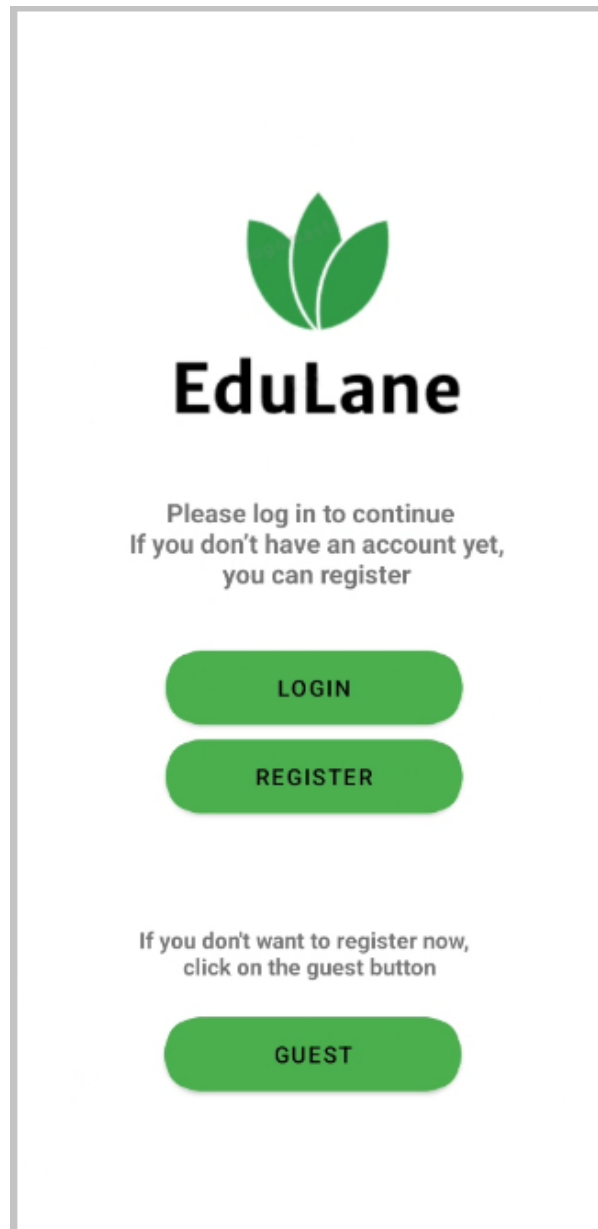
Elsőként tehát nézzük meg a mobil alkalmazást, aminek minden funkcionalitása egy regisztráció után igénybe vehető (ellentétben a webes felülettel).



7.1. ábra. Alkalmazás ikonja a készülék menüjében

7.1.1. Kezdőképernyő

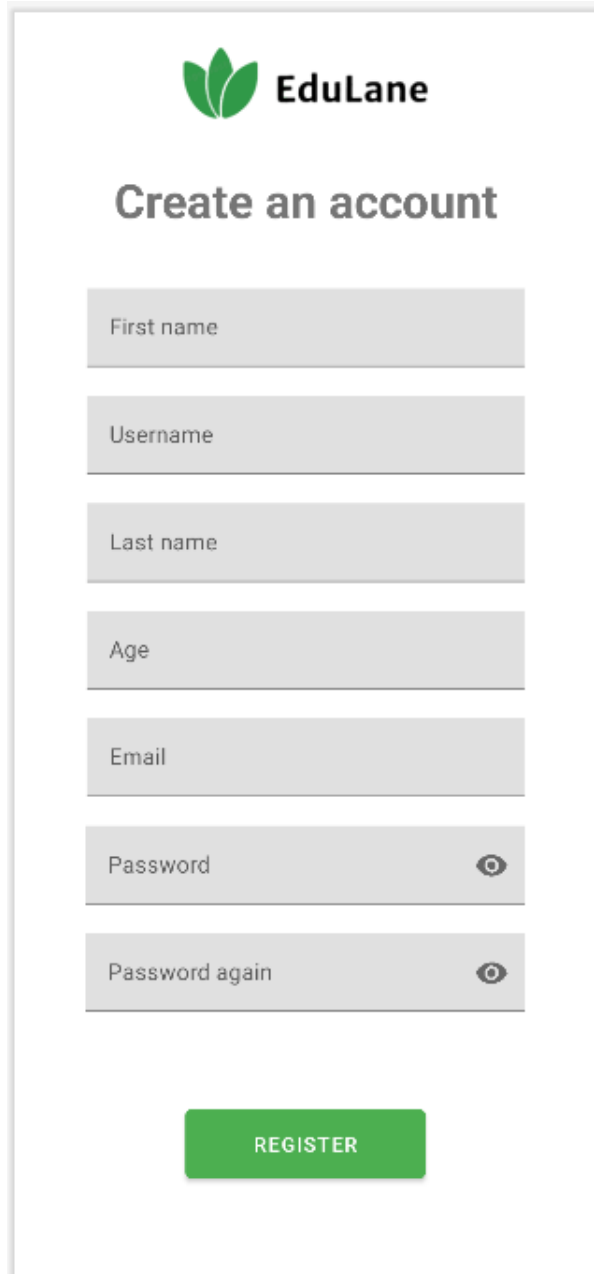
Az alkalmazás első indulásakor egy üdvözlő felület köszönti a felhasználót, ahol lehetősége van három opció közül választani: Regisztráció, Bejelentkezés és Folytatás vendégként. Az első két opció lenyomása után az alkalmazás a nevükkel járó tevékenység oldalára navigál, míg a harmadik lehetőség azonnal a főoldalra küldi a felhasználót, viszont ott korlátozott lehetőségei lesznek az alkalmazás használatára.



7.2. ábra. Kezdőképernyő

7.1.2. Regisztráció

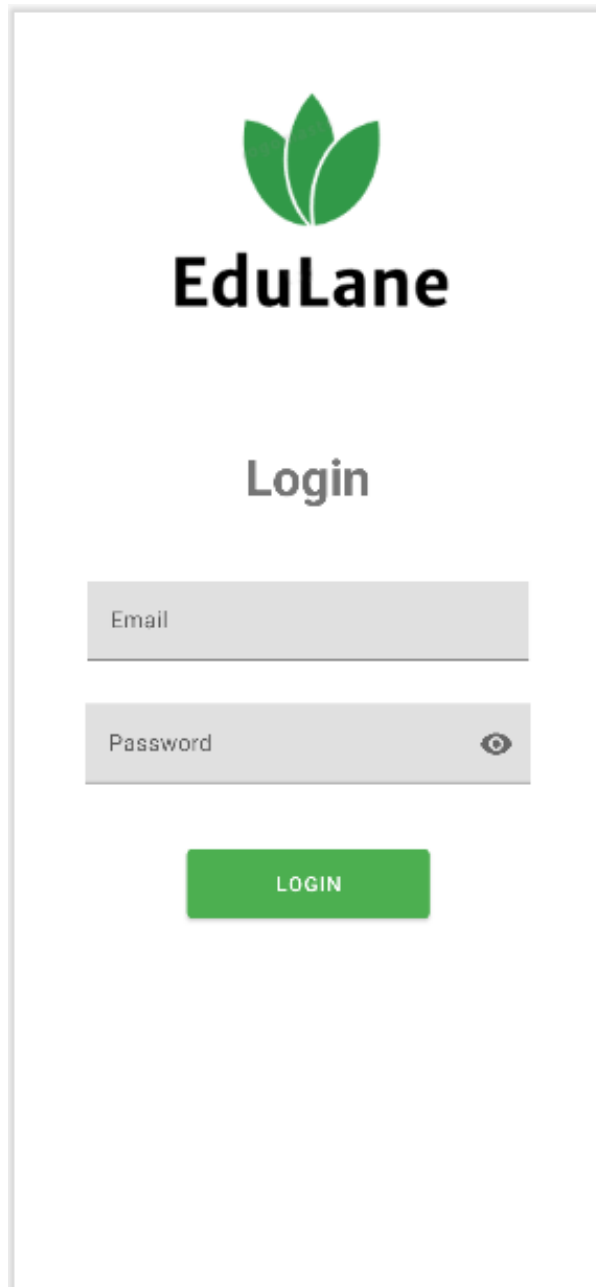
A regisztrációs felületen egy egyszerű űrlap fogad, itt tudjuk kitölteni a regisztrációs szükséges adatokat. Ha mindent jól adtunk meg, az alkalmazás tovább navigál minket a megfelelő helyre.

The image shows a registration form for EduLane. At the top, there is a green leaf logo followed by the text "EduLane". Below this, the heading "Create an account" is displayed in a bold, dark grey font. The form consists of several input fields stacked vertically: "First name", "Username", "Last name", "Age", "Email", "Password", and "Password again". Each field is a light grey rectangle with its label inside. The "Password" and "Password again" fields have a small eye icon on the right side, indicating a toggle for password visibility. At the bottom of the form, there is a green rectangular button with the word "REGISTER" in white, uppercase letters.

7.3. ábra. Regisztrációs oldal

7.1.3. Bejelentkezés

Ezen az oldalon tudjuk megadni a bejelentkezési adatainkat, amit a mobil alkalmazás elküld a szervernek és amennyiben jó adatokat adtunk meg, máris a főképernyőn találjuk magunkat. Téves adatok megadásánál a megfelelő hibaüzenet ugrik fel, ami után újra megpróbálhatjuk a bejelentkezést.

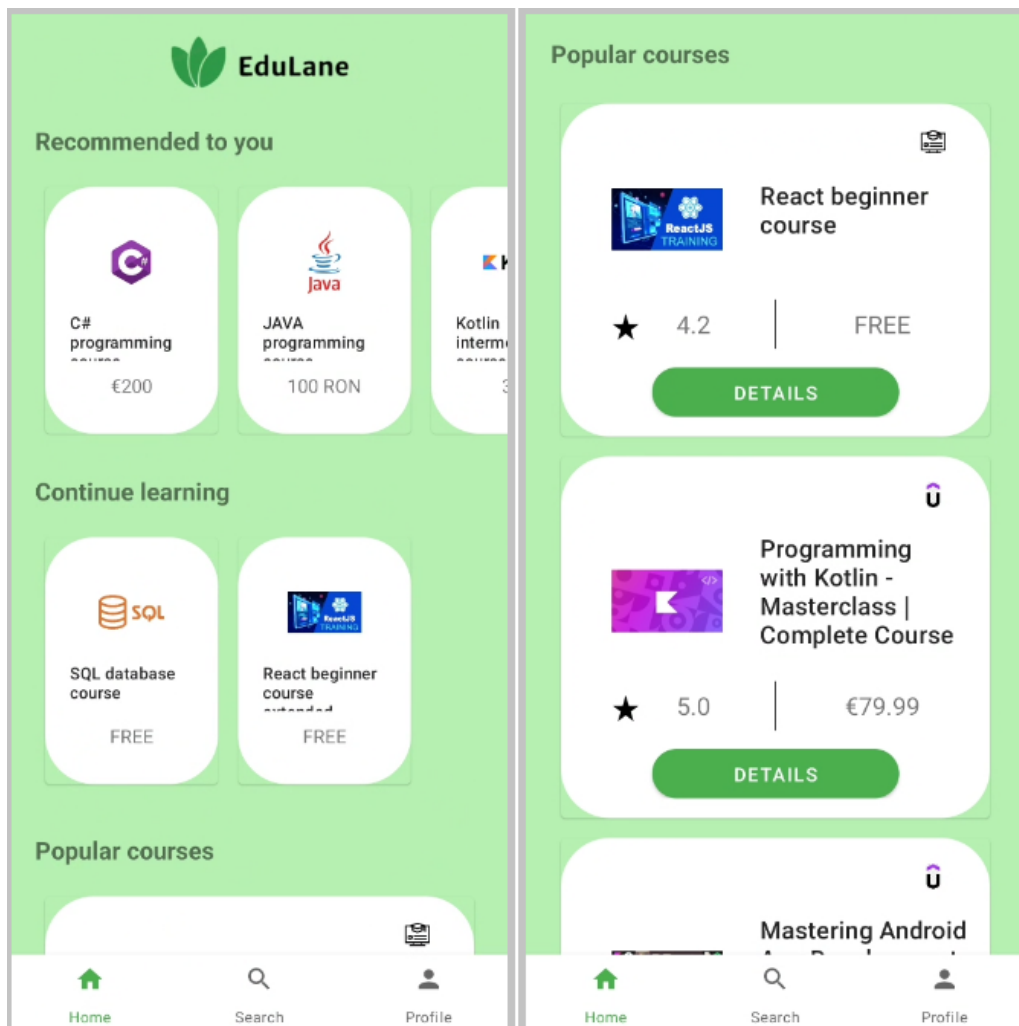
The image shows a mobile application login screen for 'EduLane'. At the top center is a green logo consisting of three leaves. Below the logo, the text 'EduLane' is displayed in a large, bold, black font. Underneath the name, the word 'Login' is centered in a smaller, bold, black font. There are two input fields: the first is labeled 'Email' and the second is labeled 'Password'. The 'Password' field has a small eye icon to its right, indicating a toggle for password visibility. Below these fields is a green rectangular button with the word 'LOGIN' in white, uppercase letters.

7.4. ábra. Bejelentkezés oldal

7.1.4. Főoldal

A főoldal az elsődleges képernyő, ami bejelentkezés vagy vendégként való belépés után fogadja a felhasználót. Itt a felhasználói profiltól függően különböző csoportokban jelennek meg a tanfolyamok, viszont ami közös, hogy mindig a legnépszerűbb tanfolyamok is megjelennek itt. Továbbá fontos megemlíteni, hogy ha nem jelentkezett ki a felhasználó az app bezárása után, automatikusan átugorja a bejelentkezést az alkalmazás és azonnal használhatók a funkcionalitások.

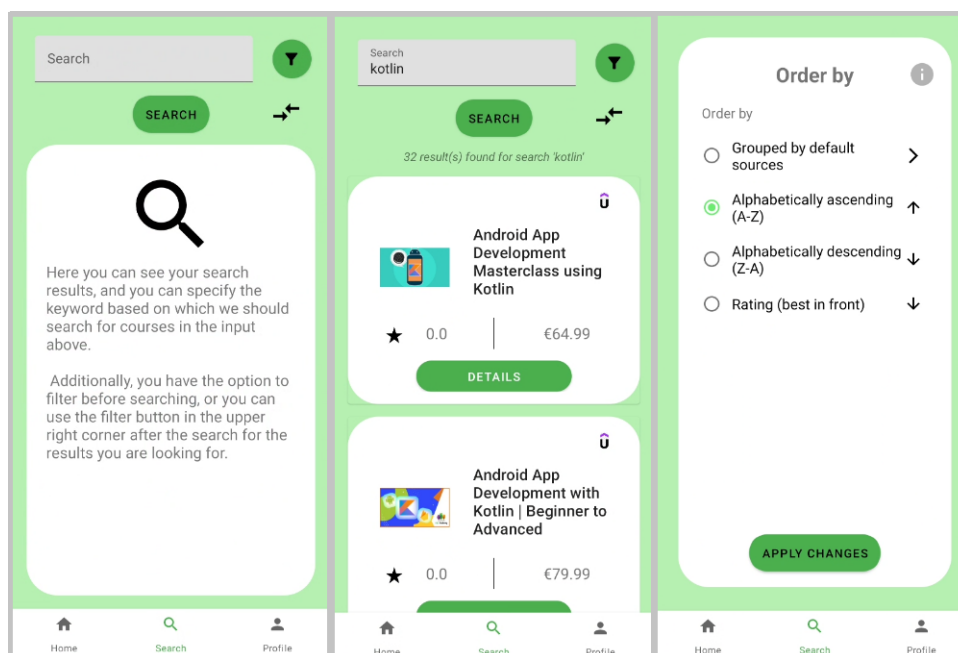
Továbbá itt találkozhat először a felhasználó a navigációs sávval, ami a képernyő alján szerepel. Innen tud ugrni más felületekre egy gombnyomás által.



7.5. ábra. Főoldal

7.1.5. Tanfolyamok keresése

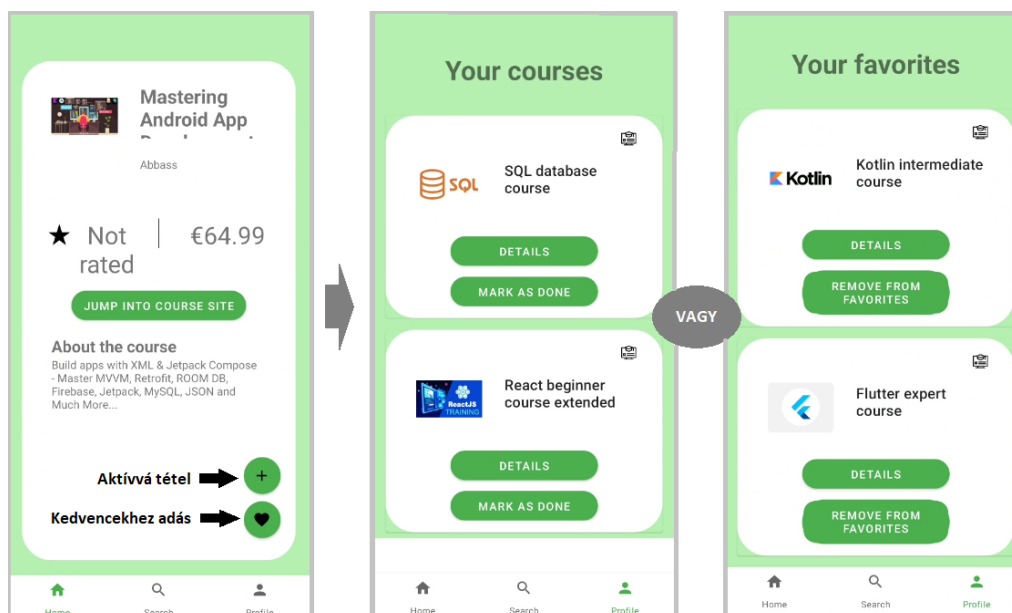
Ezen az oldalon lehet keresni tanfolyamokra kulcsszavak alapján, illetve különböző rendezési és szűrési lehetőségek is vannak.



7.6. ábra. Tanfolyamok keresése

7.1.6. Tanfolyamok kezelése

Ezen az oldalon lehet minden információt megtalálni az adott tanfolyamról, illetve a hozzá köthető műveletek is itt végezhetőek el.



7.7. ábra. Részletek

7.1.7. Tanfolyam megjelölése elvégzettként

Mivel a tanfolyamok elvégzése csak bevallásos alapon történik, így a felhasználóhoz rendelt tanfolyam elvégzése nem ellenőrizhető. Ugyanakkor egy kis teszt hozzárendelhető, ahol valamennyire ki lehet szűrni a potenciális csalókat.

The image shows two mobile app screens side-by-side, both with a light green border. The left screen has a title 'Example question title' and a section 'Answers' with four radio button options: 'Example answer 1' (selected), 'Example answer 2', 'Example answer 3', and 'Example answer 4'. A green 'NEXT' button is at the bottom. The right screen has a title 'Please share your opinion on the course' and a section 'Answers' with a large gray text input area. Below it is a section 'Rate the course' with five radio button options: '1', '2', '3', '4', and '5' (selected). A green 'SUBMIT' button is at the bottom.

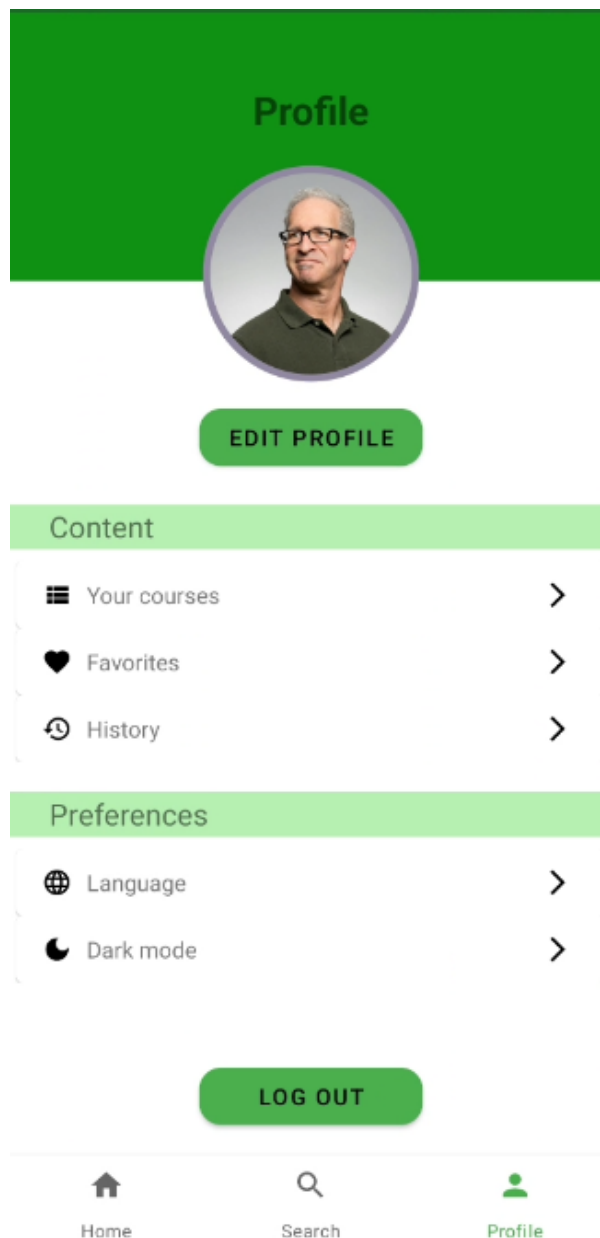
7.8. ábra. Tanfolyamhoz visszajelzés adása

7.1.8. Vendékként való belépés

Néhány oldalon az alkalmazás nem enged hozzáférést a tartalomhoz, amíg a felhasználó be nem jelentkezik.

7.1.9. Profil

Bejelentkezett felhasználók itt tudnak profillal kapcsolatos tevékenységeket végezni és kijelentkezni.



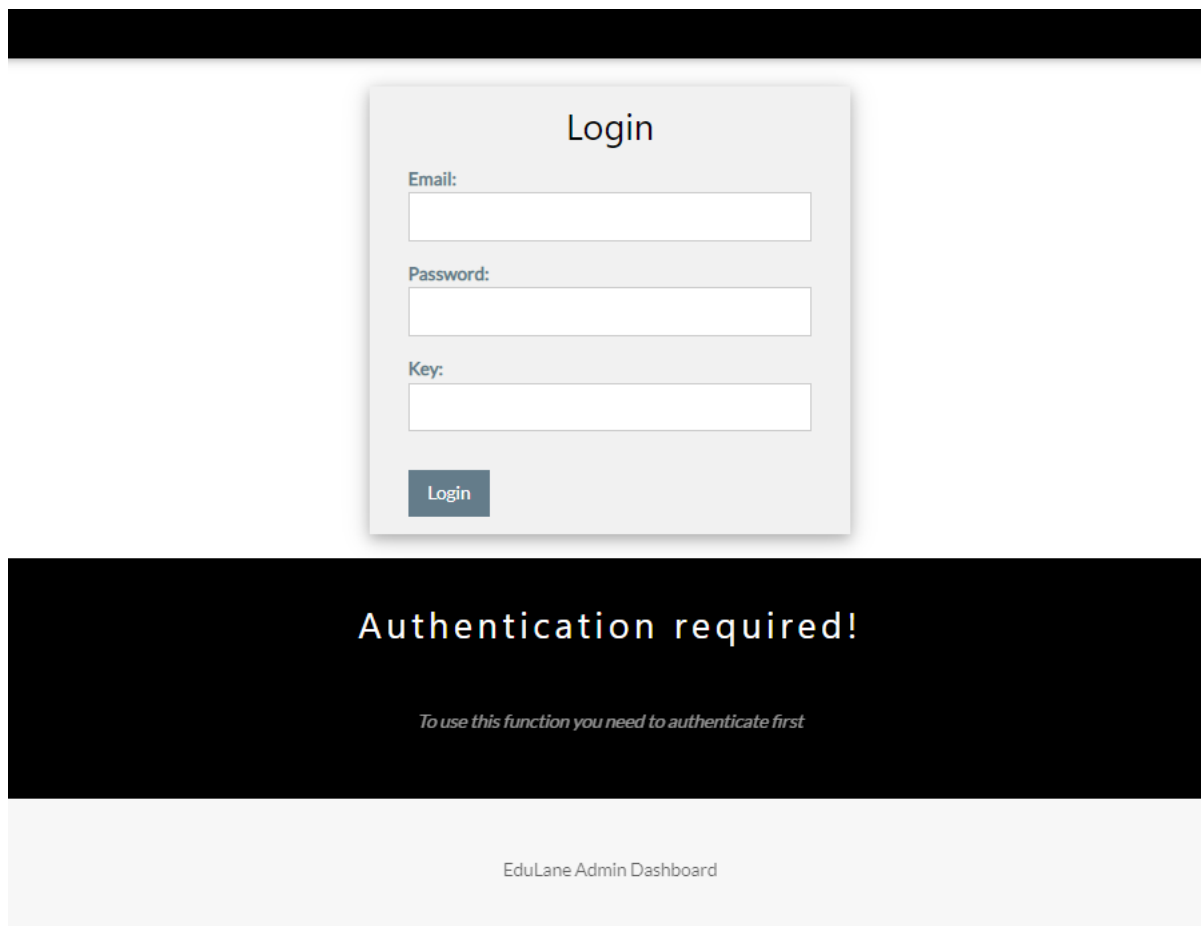
7.9. ábra. Profil oldal

7.2. Webes felület

A webes felület lényegében a teljes adminisztrátori tevékenységnek ad egy vizuális felületet. Ezáltal közvetlenül tudunk adat módosítani az adatbázisban, amely a felhő alapú adatbázis miatt azonnali hatással bír a felhasználó mobilon megjelenő adataira.

7.2.1. Bejelentkezés

A bejelentkezési felület fogadja az admint a webes felület megnyitásakor. Mivel az admin felhasználói profilok is hasonlóan vannak eltárolva, mint a sima felhasználók, ezért egy külön admin kulcs is kell a bejelentkezéshez. Egyelőre ez az érték az *adminkey*.



The image shows a web application interface. At the top, there is a black header bar. Below it, a light gray box contains a 'Login' form. The form has three input fields labeled 'Email:', 'Password:', and 'Key:', each followed by a white text input box. Below the 'Key' field is a blue 'Login' button. Below the login form, there is a large black rectangular area with the text 'Authentication required!' in white. Underneath this, in smaller white italicized text, it says 'To use this function you need to authenticate first'. At the bottom of the page, there is a light gray footer bar with the text 'EduLane Admin Dashboard'.

7.10. ábra. Admin bejelentkezési felület

7.2.2. Felhasználói profil módosítása

Minden felhasználói profil innen módosítható, a meglévő információ automatikusan kitölti a mezőket, ízlés szerint lehet módosítani az adatokat.

7.2.3. Tanfolyam hozzáadása / törlése

Egy felhasználóhoz hozzá lehet csatolni vagy épp el lehet venni tőle néhány kattintással tanfolyamokat. Ez akkor lehet hasznos, ha valamilyen hiba okán a felhasználó nem tudott hozzáadni tanfolyamot, vagy épp egy olyan került hozzá csatolásra, amit nem ő akart.

7.2.4. Felhasználó hozzárendelése felhasználói csoporthoz

Itt lehetséges hozzáadni felhasználókat a különböző, előre létrehozott csoportokhoz.

Group Name	Group Members
Group A	Kimi Dawn
Group B	John Smith
Group C	Alex Wilson
<button>Add Group</button>	<button>Add Member</button> <button>Remove Member</button>

7.11. ábra. Felhasználó hozzárendelése csoporthoz

7.2.5. Tanfolyam hozzárendelése felhasználói csoporthoz

Ez leginkább az oktatók számára fontos, hiszen itt tud hozzárendelni egy oktató a csoportjához elvégzendő tanfolyamokat.

Group Name	Group Courses
Group A	C# programming course
Group B	JavaScript fundamentals
Group C	Python for beginners
	<button>Add Course</button> <button>Remove Course</button>

7.12. ábra. Tanfolyam hozzárendelése csoporthoz

7.2.6. Tanfolyam módosítása

Itt az adatbázisban szereplő összes tanfolyam módosítható, a kilistázott elemre kattintás során automatikusan a megfelelő szerkesztési zóna várja az adminisztrátort.

Update Course

Category

Programming

ID

-NXJzPXu-xuuUZqkWC-O

Description

React test course. It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal

Author

Lorem Ipsum

Name

React beginner course

URL

https://www.pluralsight.com/courses/react-fundamentals-update?utm_source=google&utm_medium=paid-sea

Is Paid

No

Price

0

Currency

EUR

Image URL

https://trisectinstitute.com/wp-content/uploads/2021/12/best-react-js-training-institute-1.png

Rating

4.2

Website

0

Update Course

7.13. ábra. Tanfolyam módosítása

8. fejezet

Limitációk és továbbfejlesztési lehetőségek

8.1. Limitációk

- Az alkalmazás csak internetkapcsolattal működik, mivel szükség van a szervertől lekért információkra
- A felhasználók az elvégzett tanfolyamokat bevallásos alapon tudják behelyezni az elvégzettek közé.
- Jelenleg csak három platform tartalma elérhető, mivel jelenleg csak ezek kínálnak publikus elérést API által a tartalomhoz. Viszont ez kiküszöbölhető azzal, ha saját tanfolyamként adunk hozzá egy másik platformról is tartalmat.
- Az értékelési pontrendszer eltéréseket mutathat az eredetihez képest: sajnos a Udemy és a Youtube is megszüntette a hivatalos értékelések hozzátársítását a tartalmaikhoz, így ez már publikusan nem elérhető - ezért csak az alkalmazás felhasználók általi értékelések képezik az alapot.

8.2. Továbbfejlesztési lehetőségek

8.2.1. Mesterséges intelligencia használata

Az alkalmazás egyik talán legígéretesebb továbbfejlesztési lehetősége az a mesterséges intelligencia beépítése. Ezzel egy olyan modellt lehetne létrehozni, amely a felhasználók szokásai és preferenciái alapján tudnának ajánlani különböző releváns tanfolyamokat - ezzel gyakorlatilag kiváltható lehetne egy adott használati idő után a keresési folyamat, hiszen mindig megkapná a felhasználó azt, amire valóban vágyik.

8.2.2. Értékelések megosztása a felhasználók között

Egy másik hasznos funkció lehetne, ha az elvégzett tanfolyamokat a felhasználók szöveges formában is tudnák értékelni, ezek pedig megjelennének szintén a tanfolyam adatai között. Ezzel pedig meg tudná indokolni egy felhasználó a tanfolyamra adott pontszámot, leginkább szélsőséges esetekben lenne ez kifejezetten hasznos.

Összefoglaló

Dolgozatomban egy tanfolyamokat kezelő alkalmazás létrehozásával foglalkoztam, amelyet különböző programozási technológiák segítségével valósítottam meg. Először ismertettem az online tanfolyamokhoz köthető problémákat és elméleti alapokat, amiket egy szakirodalmi áttekintéssel alapoztam meg. Továbbá az erre épülő alkalmazás alap koncepcióját és célkitűzéseit is ismertettem.

A tervezési és fejlesztési folyamat is bemutatásra került, kiegészítve a felhasznált technológiák ismertetésével és alapvető bemutatásával. Az alkalmazáshoz köthető legfontosabb technológiák közé tartozik a Kotlin, az ASP.NET és a Firebase adatbázis. A három fő alkalmazás komponenset is részletesen bemutattam, rávilágítva a különböző részek szükségességére, illetve, hogy azok milyen módon kötődnek egymáshoz. Ezen komponensek forráskódjai a következő linkeken találhatóak meg:

- [EduLane Mobil alkalmazás forráskódja](#)
- [EduLane Web szerver forráskódja](#)

Végül a már kész alkalmazás is bemutatásra került, amelyekhez valós képernyőképeket használtam fel a jobb szemléltethetőség érdekében. Ismertettem az alkalmazás limitációit és a továbbfejlesztési lehetőségeket is.

Jövőbeni terveimet illetően szeretném megvalósítani az alkalmazás egyik legnagyobb lehetőségét, a mesterséges intelligencia beépítését, illetve mélyebb ismereteket szerezni az ehhez köthető technológiákhoz. Továbbá szeretném minimalizálni az alkalmazás limitációit, hogy az a lehető legjobb felhasználási élményt nyújthassa.

Ábrák jegyzéke

1.1. Alkalmazás logo	12
2.1. Hasonló alkalmazások	14
5.1. Példa GitLab táblákra	23
5.2. Ágak (branchek) a Sourcetree-ben	23
6.1. Mobil alkalmazás architektúrája	26
6.2. Web alkalmazás architektúrája	26
6.3. A mobil app alapvető csomag struktúrája	27
6.4. XML fájlok struktúrája	28
6.5. Navigáció példa	29
6.6. Adatbázis struktúra	30
6.7. Eltárolt felhasználói adatok az adatbázisban	31
6.8. Felhasználói lista	32
6.9. Tanfolyam osztálydiagram	33
6.10. Egyszerűsített tanfolyam osztálydiagram	34
6.11. Felhasználói profil osztálydiagram	34
6.12. Kezdőképernyő az alkalmazás indulásakor	35
6.13. Vendég felhasználó	35
6.14. Bejelentkezett felhasználó	36
6.15. Webes felület (adminisztrátor)	37
6.16. Bejelentkezés szekvencia diagram	38
6.17. Keresés szekvencia diagram	38
6.18. Kedvencekhez adás szekvencia diagram	39
7.1. Alkalmazás ikonja a készülék menüjében	44
7.2. Kezdőképernyő	45
7.3. Regisztrációs oldal	46

7.4. Bejelentkezés oldal	47
7.5. Főoldal	48
7.6. Tanfolyamok keresése	49
7.7. Részletek	49
7.8. Tanfolyamhoz visszajelzés adása	50
7.9. Profil oldal	51
7.10. Admin bejelentkezési felület	52
7.11. Felhasználó hozzárendelése csoporthoz	53
7.12. Tanfolyam hozzárendelése csoporthoz	53
7.13. Tanfolyam módosítása	54

Táblázatok jegyzéke

3.1. Felhasználói követelmények	16
3.2. Adminisztrátori követelmények	17

Irodalomjegyzék

- [AA19] Miguel A Angulo and Ozgur Aktunc. Using github as a teaching tool for programming courses. In *2018 Gulf Southwest Section Conference*, 2019.
- [CGM18] Iuliana Cetina, Dumitru Goldbach, and Natalia Manea. Udemy: a case study in online education and training. *Revista Economică*, 70(3):46–54, 2018.
- [KBF⁺00] Judy Kay, Michael Barg, Alan Fekete, Tony Greening, Owen Hollands, Jeffrey H Kingston, and Kate Crawford. Problem-based learning for foundation computer science courses. *Computer Science Education*, 10(2):109–128, 2000.
- [WBLB14] Jochen Wulf, Ivo Blohm, Jan Marco Leimeister, and Walter Brenner. Massive open online courses. *Business & Information Systems Engineering*, 6:111–114, 2014.