



MOVIE APP

Albert Brigitta és
Mezei Emese
Szoftverfejlesztés I.

Tartalomjegyzék

1.	Bevezető	2
2.	Projekt menedzsment terv	2
2.1.	Feladatok felosztása	2
2.2.	Ütemezési terv	3
3.	A projekt célja és a rendszerkomponensei	3
3.1.	A projekt célja.....	3
3.2.	Rendszerkomponensek	4
3.2.1.	Firebase	4
3.2.2.	Android (Java)	4
3.2.3.	MovieDB.....	5
4.	Feladat tervezése és megvalósítása.....	8
4.1.	Követelmények	8
4.1.1.	Rendszer követelmények	8
4.1.2.	Felhasználói követelmények.....	8
4.2.	Prototípus készítése	9
4.3.	Adatbázis	10
4.4.	Szekvenciális diagramok	12
4.4.1.	Regisztráció.....	12
4.4.2.	Bejelentkezés.....	13
4.4.3.	Profil	14
4.5.	Az alkalmazás architektúrája	14
4.6.	Az alkalmazás működése.....	15
4.6.1.	Splash screen	15
4.6.2.	Login screen	15
4.6.3.	Registration screen.....	16
4.6.4.	Home screen	17
4.6.5.	WatchList screen	18
4.6.6.	Details screen	19
4.6.7.	Profile screen.....	21
5.	Összegzés és továbbfejlesztési lehetőségek	22
5.1.	Összegzés	22
5.2.	Továbbfejlesztési lehetőségek.....	22
6.	Ábrajegyzék	22
7.	Hivatkozások.....	23

1. Bevezető

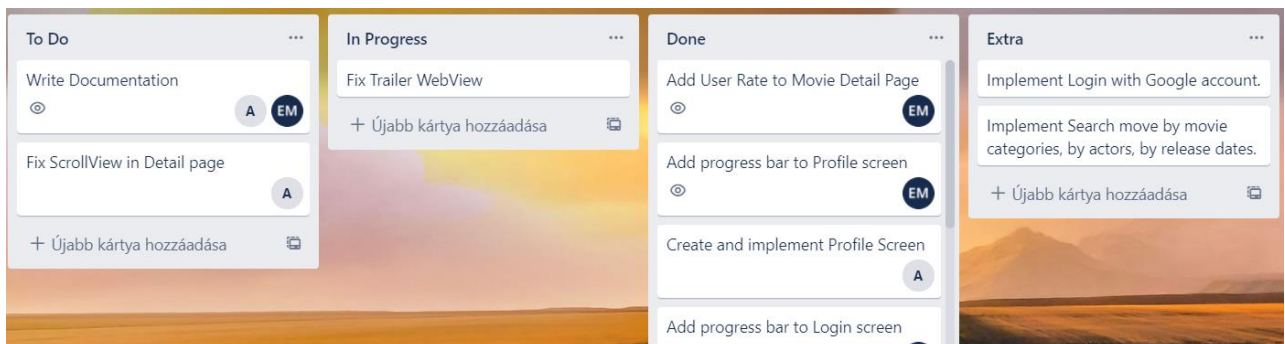
Napjainkban gyakran kerülünk abba a helyzetbe, hogy szabadidőnkben nincs mit tennünk ezért a legtöbb esetben filmek, illetve sorozatok után böngészünk az interneten. Sokunknál előfordul, hogy a keresés közben egyszerre több film is megtetszik, ekkor általában egyet választunk ki megnézésre, és az esetek nagy többségében a többi film címét elfelejtjük. Ebből a gondolatmenetből fogalmazódott meg bennünk egy olyan mobilapplikáció fejlesztése, amely segítségével könnyen találhatunk filmeket ízlésünk szerint, valamint lehetőségünk van megjelölni azokat a filmeket, amelyeket meg szeretnénk nézni.

2. Projekt menedzsment terv

2.1. Feladatok felosztása

A projektünket kezdetben nagyobb feladatokra osztottuk fel, majd ezeket bontottuk le részfeladatokra. Annak érdekében, hogy a feladatok elvégzése gördülékenyen működjön, ezért a Trello weblapú listakészítő alkalmazást használtuk, valamint a verziókövetésre a GitHubot használtuk. A projektünk [itt](#) érhető el.

A Trello alkalmazásban a három alap állapotot használtuk: To Do, In Progress, Done, ezek mellett bevittünk még egy Extra oszlopot is. Ez látható az ábra 1– en. A To Do oszlopba feltüntettünk minden feladatot, amely megoldásra várt, ezt a listát a projekt készítése alatt is folyamatosan bővítettük. Innen áthúztuk azt a feladatot, amellyel éppen foglalkoztunk az In Progress oszlopba, majd, ha elkészültünk vele, akkor átmozgattuk a Done-ba. Az Extra oszlopba azok a feladatok kerültek, amelyek felmerülhetnek a továbbfejlesztés során. Ennek az alkalmazásnak a használata azért volt fontos számunkra, mert tudtunk időpontokat beállítani egy-egy feladat határidőjeként, valamint azért is, mert tudtuk folyamatosan követni, hogy melyikünk hol tart a saját feladataival.

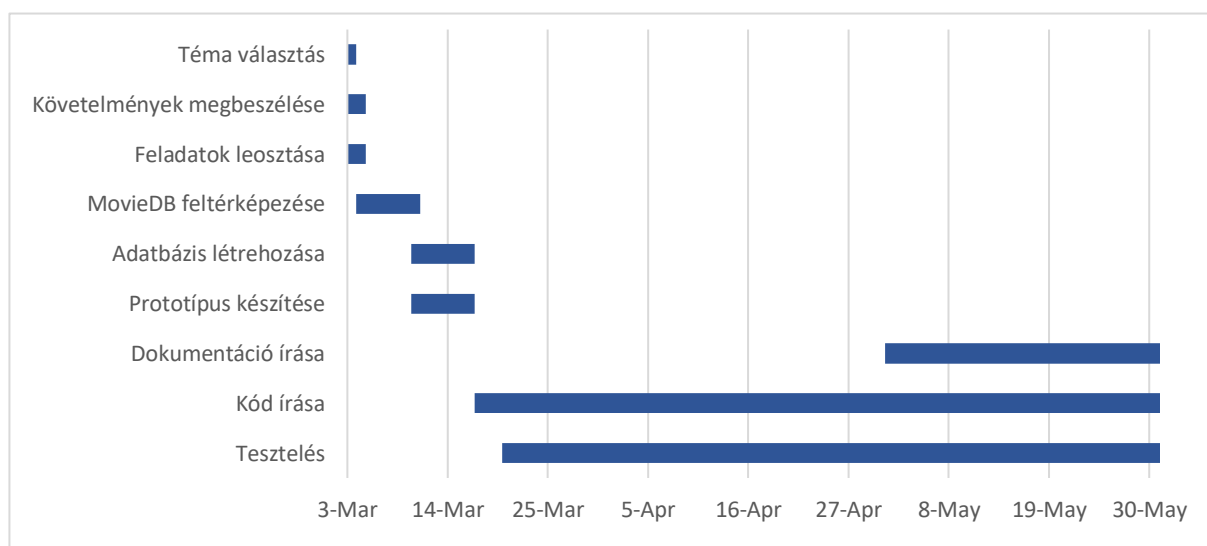


ábra 1 Trello oszlopai

2.2. Ütemezési terv

Amint az ábra 2 ütemezési terven is látható több fő feladatunk is volt. A téma kiválasztása, a követelmények megbeszélése és a feladatok leosztása már az első pár napban megtörtént. Fontosnak tartottuk a ezeket az elején megtervezni, átbeszélni, mivel a későbbiekben erre építkeztünk.

A következő fő lépés a MovieDB feltérképezése volt, mivel ez szolgáltatja a projektünk alapját. Ezt követően az adatbázis létrehozása, illetve a prototípus felépítése egy időpontban történtek. Majd mikor már a fent említett dolgok készen álltak, akkor a kód írása és néhány nap múlva a tesztelés is elkezdődött, ugyanis a folyamatos tesztelés mellett döntöttünk. A legutolsó feladat a dokumentáció írása volt, amelyet már a kód írása és a tesztelési folyamatok alatt elkezdtünk, így párhuzamosan haladtunk ezzel a három fontos feladattal.



ábra 2 Ütemezési terv

3. A projekt célja és a rendszerkomponensei

3.1. A projekt célja

A legfontosabb célkitűzésünk egy letisztult, felhasználóbarát applikáció létrehozása, amely bármilyen korosztály számára megfelelő kezelhetőséget biztosít. A célkitűzéseink közé tartozik, hogy a felhasználók meg tudják tekinteni a legnépszerűbb filmeket, azoknak különböző részleteit, mint az előzetese, kiadási év, felhasználói pontozás, egy rövid leírás,

valamint a hasonló filmeket is meg tudják tekinteni, mindezt egyetlen érintéssel. A következő fontos szempontunk az volt, hogy a felhasználó tudja lementeni a számára tetsző filmeket, hogy a következő alkalommal ne kelljen újra keresgélnie, hanem már a kedvére való filmek közül tudjon válogatni.

Azért döntöttünk úgy, hogy egy mobil applikációt fejlesztünk, mivel napjainkban szinte mindenki rendelkezik okostelefonnal és így mindenki számára könnyen elérhető a filmek keresése. Valamint néhány helyzetben egyszerűséget is nyújt a mobil applikáció, például egy utazás során általában mindenkinél ott van az okostelefonja, ezáltal míg utazgat addig egyszerűen tud filmeket, illetve sorozatokat keresgélni.

3.2. Rendszerkomponensek

3.2.1. Firebase

A Firebase egy olyan platform, amely webes, illetve mobil alkalmazások létrehozására lett fejlesztve a Google által.

A projektünkhöz a Firebase-t választottuk az adatok tárolására, illetve kezelésére. Mivel a Firebase önmagában nem csak egy termék, hanem sok kisebb-nagyobb, rendkívül hasznos tool-ok összessége egy nagy integrált egésszé gyúrva. Lehetővé teszi, hogy mindig azt használd, amire éppen szükséged van. A fő szempont, amely miatt a Firebase-t választottuk az a könnyen kezelhetőség, illetve átláthatóság. Emellett a regisztráció, valamint a bejelentkezés a FirebaseAuth szolgáltatással nagyon egyszerű, mivel mi csak a kliens szolgáltatásra kell koncentrálnunk.

3.2.2. Android (Java)

- Glide Library: képek betöltése.
- Retorfit: API-hívásokra.
- RecyclerView
- Bottom Navigation
- Constraint Layout, Relative Layout, Linear Layout
- Egyéb: TextView, ScrollView, ImageView, WebView, CircleImageView, CardView, CheckBox, ProgressBar, RatingBar, EditText, ImageButton, Button

Az alkalmazás Java programozási nyelven lett írva.

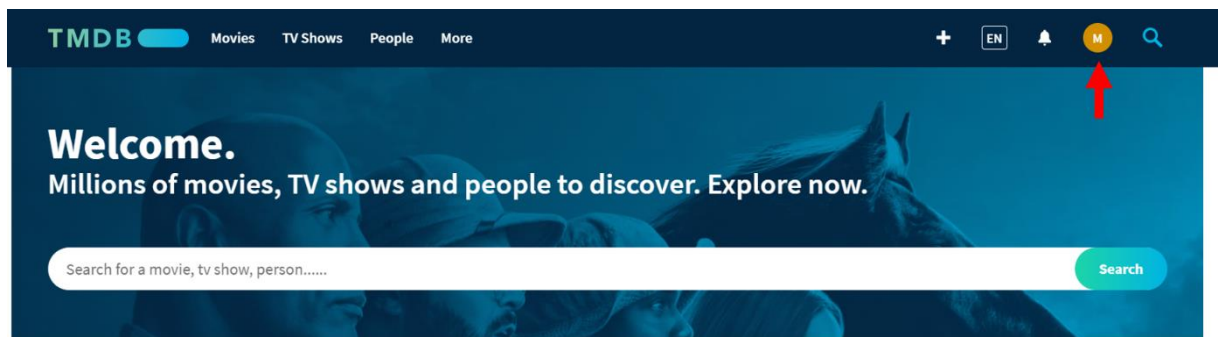
3.2.3. MovieDB

A MovieDb egy nyílt forráskódú filmeket és TV sorozatokat tartalmazó adatbázis, amelyet a maguk a felhasználók vezérelnek. A legújabb a 3-as verzió. 2008-ban alapította meg Travis Bell, de kizárólag filmplakátok gyűjtésére lett létrehozva. 2010-ben eladták az oldalt a Fan Tv-nek, de jelenleg is Travis Bell kezeli. Jelenleg több mint fél millió film és lassan 2 millió felhasználó található az adatbázisban.

Az adatbázisban egy filmről több adat van eltárolva. Ilyenek a film id, cím, leírás, kiadási év, eredeti nyelv, eredeti cím, plakát útvonala, szavazási átlag, szavazatok száma. Ahhoz, hogy fejlesztőként tudjuk használni ezt az adatbázist szükségünk van egy API kulcshoz.

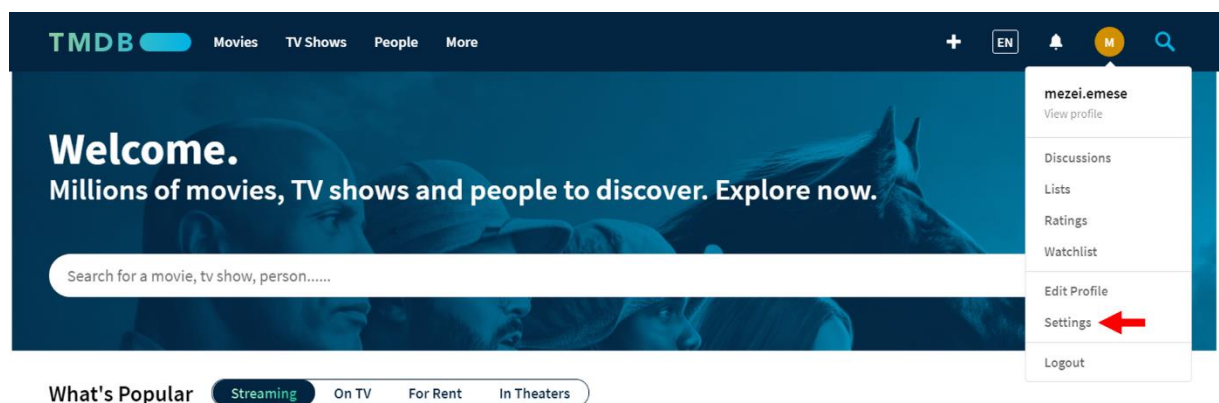
3.2.3.1. API kulcs kérése

Az API kulcs kérése nem működik mobilon, kizárólag asztali számítógépen. Az API kulcs lekéréséhez első sorban regisztrálnunk kell a [The MovieDB oldalán](#). A regisztráció után a jobb felső sorban található, kis körre (amelyben a felhasználónevünk kezdőbetűje található) kattintunk. Az ábra 3– n található piros nyíl rámutat az imént említett ikonra.



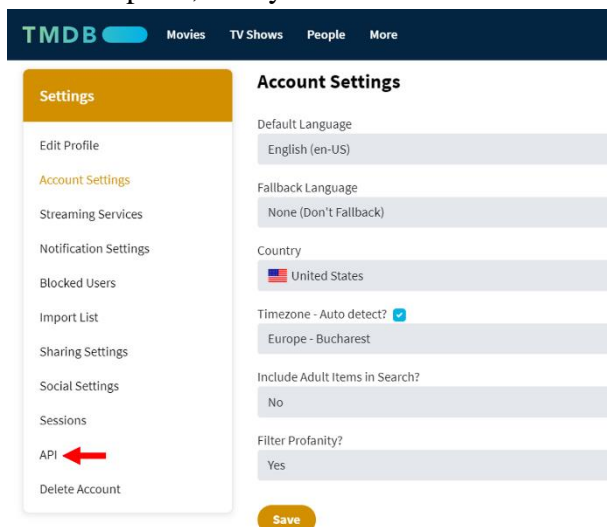
ábra 3 API kulcs lekérése első lépés

Majd a Beállítások (settings) opciót választjuk ki. Az ábra 4– en szintén piros nyíl mutatja ezt.



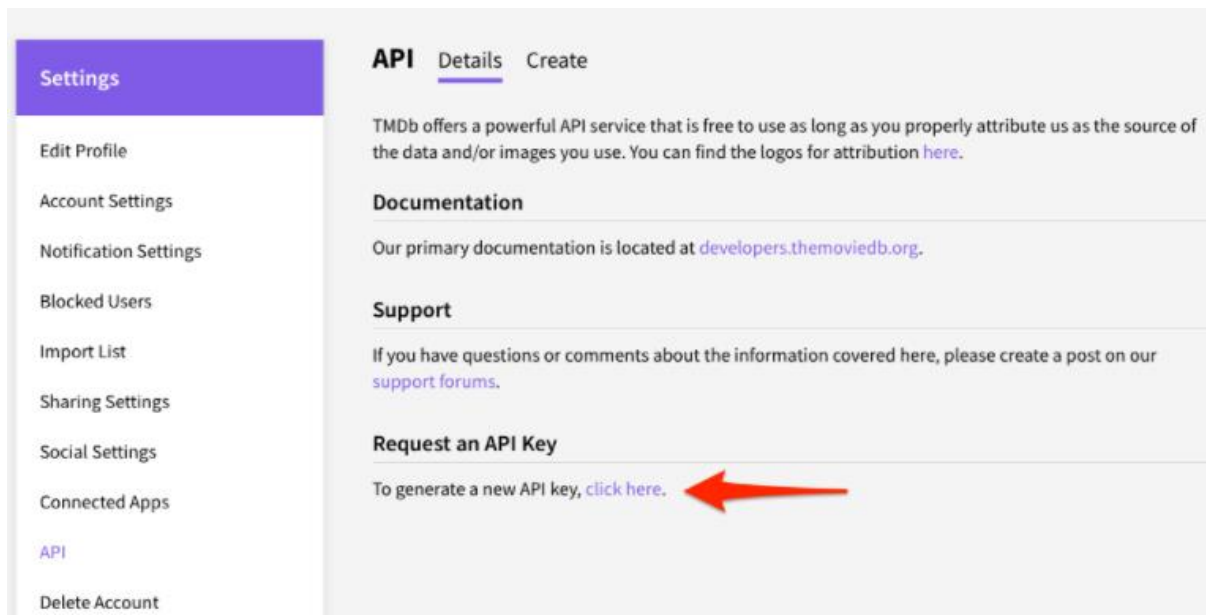
ábra 4 API kulcs lekérése második lépés

A most megnyíló weboldal bal oldalán láthatjuk, hogy milyen beállítási lehetőségeink vannak. Itt kiválasztjuk az API opciót, amely az ábra 5-n ki lett emelve.



ábra 5 API kulcs lekérése harmadik lépés

Ezt követően a jobb oldalon a „Request an API KEY” alcím alatt rá megyünk a hiperhivatkozásra. Az ábra 6 legalján található ez, szintén piros nyíl mutat erre.



ábra 6 API kulcs lekérése negyedik lépés

Amint elvégeztük ezt a néhány lépést máris fejlesztőként használhatjuk az adatbázist.

3.2.3.2. API-k

Amint megnyitjuk a TMDb (The Movie Database) [fejlesztői oldalát](#): láthatjuk, hogy az oldal bal oldalán különböző API-k vannak kilistázva. Amennyiben a listából kiválasztunk egy elemet és ráklikkelünk, akkor a jobb oldalon láthatjuk az adott API leírását, használatát, illetve lehetőségünk van a kipróbálásukra is.

Ezt követően láthatunk két fő fület, az első a Definition, míg a második a Try it out. A Definition részen megtekinthetjük, hogy mik szükségesek a lekérdezések elvégzéséhez, valamint azt, hogy hogyan néz ki egy válasz, amelynek láthatjuk a felépítését, illetve egy konkrét példát is. A Try it out fülön belül, ahogy a neve is mondja, kitudjuk próbálni, hogy mit kapunk a lekérdezés során.

A TMDb fejlesztői oldalán számtalan API kipróbálható, valamint használható, köztük találhatóak filmekre, sorozatokra, TMDb felhasználókra is API-k. Az általunk készített applikáció több filmekre vonatkozó API-t használ: a legfelkapotabb filmek kilistázásánál, keresésnél, a filmek részleteinél, tutoriál videók, képek, illetve hasonló filmek megjelenítésénél.

A TMDb oldaláról a következő API-kat használtuk fel a fejlesztés során:

- Get Popular: Lekérdezés módja: GET /movie/popular. A lekérdezés során kapunk egy listát, amely a legnépszerűbb filmeket tartalmazza. Ez a lista naponta frissül.
- Get Details: Lekérdezés során megkapjuk az általános információkat a filmekről. A lekérdezés módja: GET /movie/{movie_id}. Eredményként megkapunk a movie_id-nak megfelelő film 25 információját.
- Get Images: Lekérdezés módja: GET /movie/{movie_id}/images. A lekérdezés során a movie_id-nak megfelelő film képeit kapjuk meg.
- Get Videos: Lekérdezés módja: GET /movie/{movie_id}/videos. A lekérdezés során a movie_id-nak megfelelő film videóját (thriller) kapjuk meg.
- Get Similar Movies: Lekérdezés módja: GET /movie/{movie_id}/similar. A lekérdezés során a movie_id-nak megfelelő filmhez hasonló filmeket és annak részletei kapjuk meg.

Fontos kiemelni, hogy az olyan fejlesztések során, ahol külső API-kat használunk nagy hangsúlyt kell fektetni a dokumentációra, valamint az API-k tesztelésére, amely után eldönthetjük, hogy a fejlesztési céljainknak megfelelnek vagy sem.

4. Feladat tervezése és megvalósítása

4.1. Követelmények

4.1.1. Rendszer követelmények

A rendszer követelményekhez tartoznak a különböző engedélyek megadásai, ilyenek a:

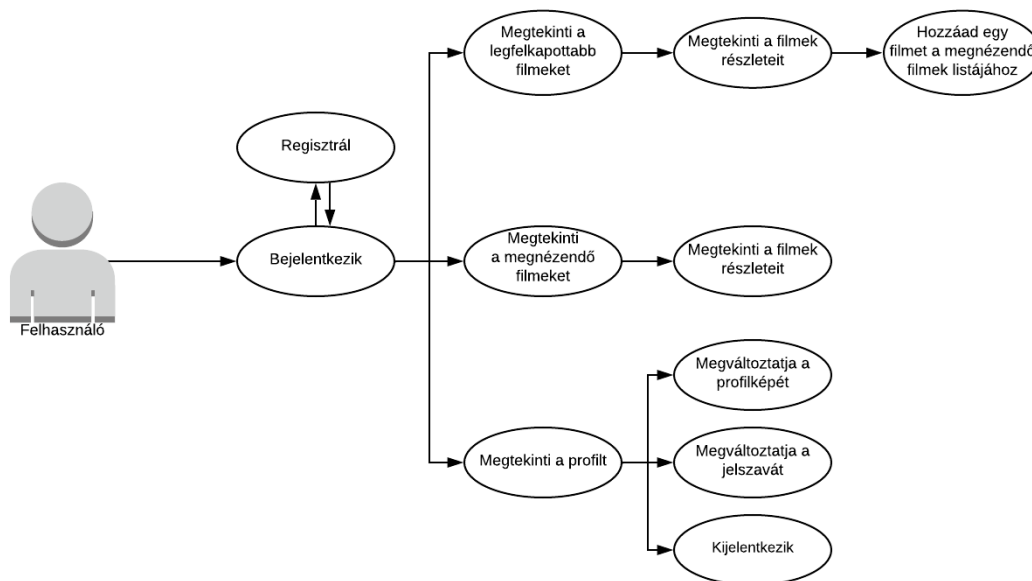
- Hozzáférés a tárhelyhez
- Hozzáférés a galériához
- Internethozzáférés

Az engedélyek megadásán kívül ide tartozik még:

- Minimális API szint: 21
- Minimális Android verzió: Android 5.0 – megközelítőleg a készülékek 94,1% képes futni az alkalmazás.

4.1.2. Felhasználói követelmények

A legfontosabb felhasználói követelmény, hogy a felhasználónak legyen egy android operációs rendszerrel rendelkező okostelefonja. Amint az ábra 7-en is látható arra törekedtünk, hogy minél egyszerűbb applikációt tudjunk készíteni a felhasználó számára.



ábra 7 Használati eset diagram

A felhasználó amint megnyitja az applikációt egy bejelentkezést lát elsőként. Abban az esetben, hogyha rendelkezik már saját fiókkal, akkor egyszerűen bejelentkezhet, hogyha nincs fiókja akkor át tud menni a regisztrációra. A mezők helyesen való kitöltése után a felhasználó regisztrálva lesz, ezt követően újra visszakerül a bejelentkezésre és ekkor be is tud jelentkezni.

Bejelentkezés után a Home oldalon találja magát a felhasználó, ahol a legfelkapottabb filmek listáját láthatja, valamint egy keresésre is lehetősége van a filmek között címek szerint. Hogyha rámegy egy filmre, akkor láthatja a film részleteit, illetve egy gombot, amellyel hozzá tudja adni a filmet megtekinteni kívánt filmek listájához.

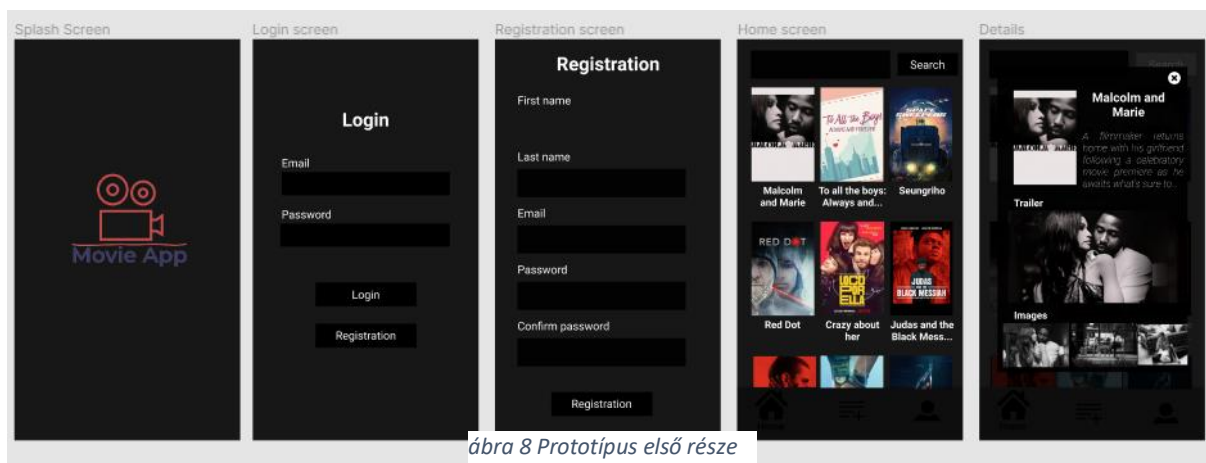
A középső menüpontra rálépve a felhasználó láthatja, hogy milyen filmeket adott hozzá a megtekinteni kívánt filmek közé. A filmek részletei ugyanúgy láthatóak abban az esetben, hogyha a felhasználó rákattint egy filmre.

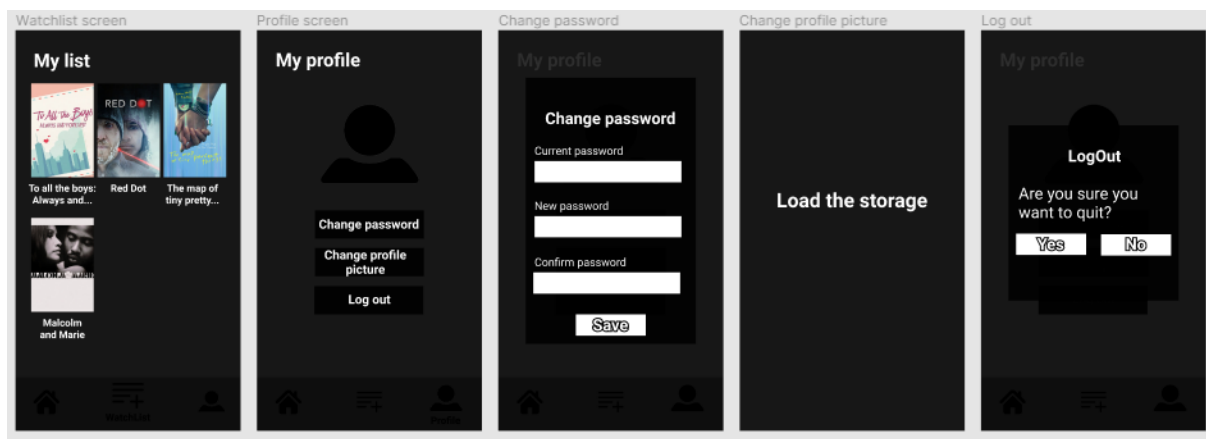
A legutolsó menüpont a Profil, ahol a felhasználó láthatja a saját profilképét, felhasználónevét, email címét. Valamint lehetősége van a profilkép, illetve a jelszó megváltoztatására, valamint a kijelentkezésre.

4.2. Prototípus készítése

A prototípus elkészítésére a Figmát használtuk, amely egy webalapú vektorgrafikus szerkesztő, valamint prototípus készítő eszköz. Az ábra 8-on és ábra 9 -en látható, hogy hogyan terveztük meg az oldalak kinézeteket.

A Figma segítségével képernyőket tudtunk hozzáadni, amelyre ikonokat, alakzatokat és formákat tettünk rá. Lehetőséget ad arra is, hogy színeket állítsunk be, azoknak a telítettségét, valamint az oldalak rétegeinek is többféle beállítása lehetséges.





ábra 9 Prototípus második része

Az általunk megtervezett prototípus a [itt](#) érhető el.

4.3. Adatbázis

A bejelentkezés, illetve a regisztrációra a FirebaseAuth, valamint a Firestore szolgáltatásokat használtuk. A FirebaseAuth segítségével nagyon egyszerű a login funkciót létrehozni egy alkalmazáshoz. A projektünkben egy email-jelszó regisztrációt hoztunk létre, itt a FirebaseAuth hibaüzenetet küld vissza az alkalmazásnak sok esetben. A felhasználói adatok eltárolása az ábra 10-en látható. Például, ha az adott email címmel már volt regisztrálva, vagy ha a jelszó nem elég hosszú, ha nem egyezik az emailcím a jelszóval stb. Mindemellett a FirebaseAuth regisztráció során emailt küld a regisztráció aktiválásáról. A visszaigazoló email az ábra 11-hez hasonló.

Search by email address, phone number, or user UID					Add user	⌂	⋮
Identifier	Providers	Created	Signed In	User UID			
@gmail.com	✉	Apr 24, 2021	May 29, 2021	espQ4CF5Lhhj6qYvB9NAtdkKgZT2			
@gmail.com	✉	Apr 26, 2021	May 19, 2021	kkUJNWAGpvrZGkSpB9qc3qpU8l...			
Rows per page: 50					1 - 2 of 2	<	>

ábra 10 Felhasználó adatainak tárolása az adatbázisban

Hátrányként megemlíteném, hogy a projekt neve, amelyet az email template-t használ, nem módosítható egyszerűen, egy külön kérés folyamat kell hozzá, amelyet a mi esetünkben nem fogadtak el.

Hello,

Follow this link to verify your email address.

https://movieapp-65637.firebaseio.com/.auth/action?mode=verifyEmail&oobCode=7Fgj6yL0rP3Uac2buFgQNig-dvTFC6HRtuO7rlmfayQAAAF4eI75AQ&apiKey=AlzaSyD7Fo_SFXifSjYnD4u0iiRyVGY7cvxvA8w&lang=en

If you didn't ask to verify this address, you can ignore this email.

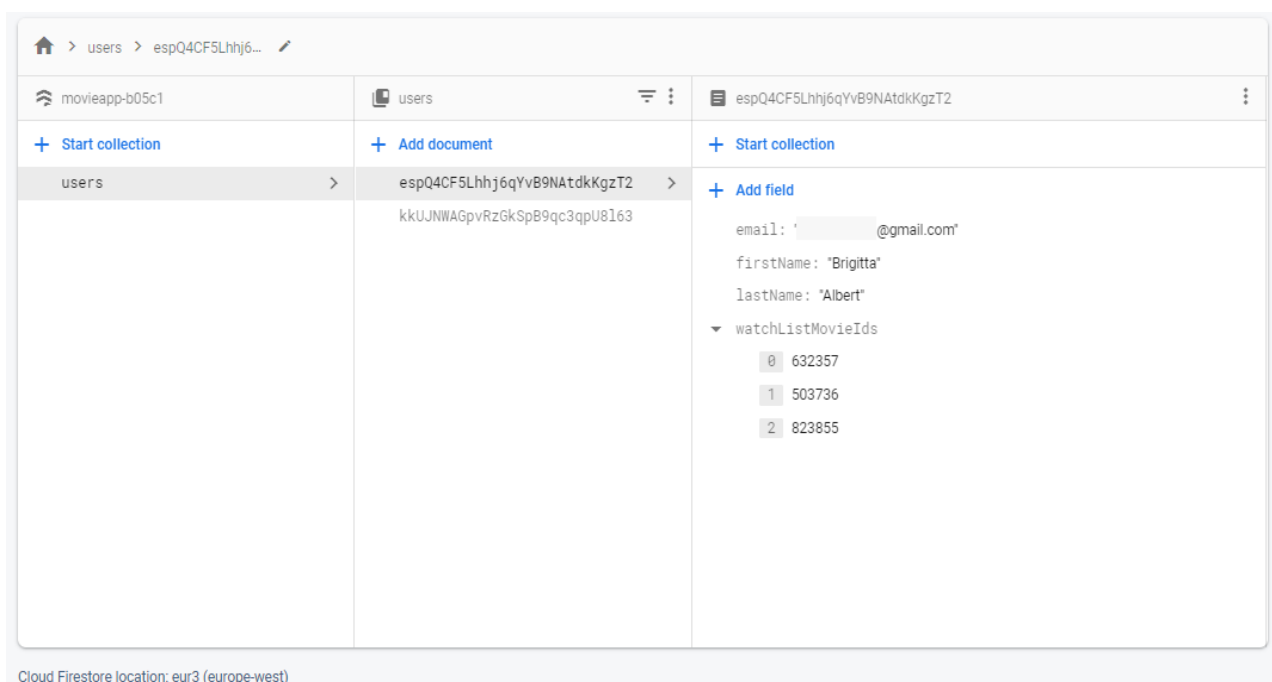
Thanks,

Your project-793487324022 team

ábra 11 A regisztrációt aktiváló email

Az email és a jelszón kívül a további regisztrációs adatokat a FireStore segítségével tároltuk. Emellett a FireStore segítségével tároltuk el, azt, hogy melyik felhasználó milyen filmeket tett rá a megtekinteni kívánt filmek listájára.

Az adatok tárolása, amint az ábra 12-n is fel van tüntetve a következőképpen zajlott. Egy users nevű collection tartalmazta a felhasználókat, minden felhasználó egy külön documentet kapott, és a regisztrációnál kapott User UID-val azonosítódott. Minden felhasználónál eltároltuk az email címét, vezetéket, illetve keresztnévét, valamint a watchlistjére adott filmek azonosítóját.



Cloud Firestore location: eur3 (europe-west)

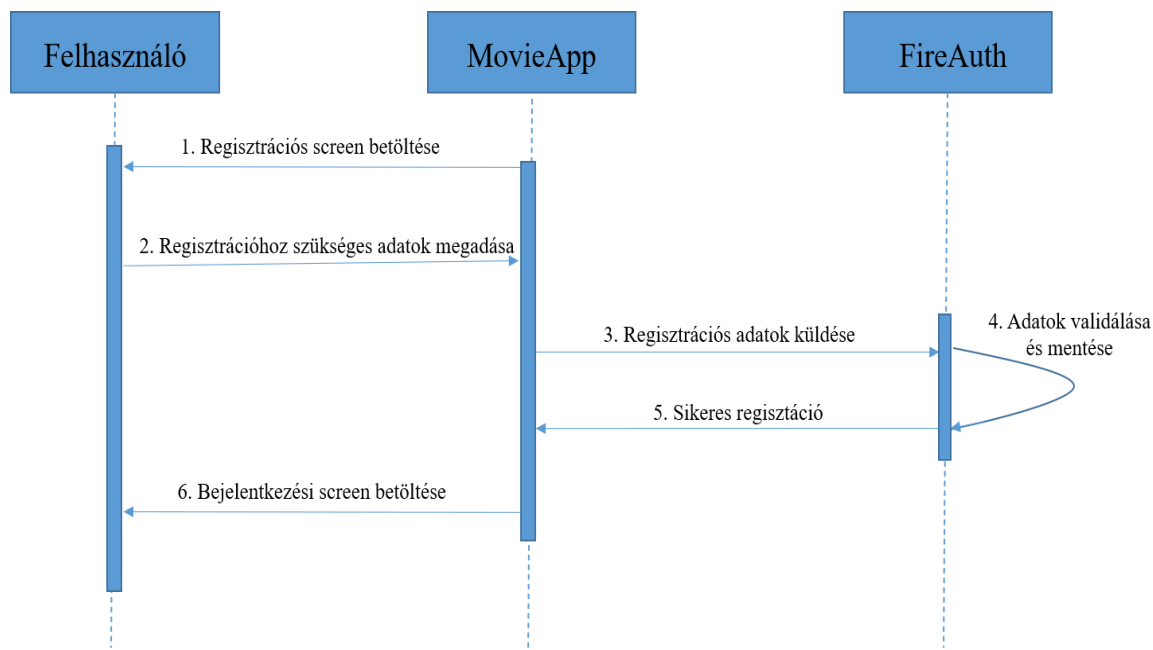
ábra 12 Adatok tárolása az adatbázisban

A felhasználók által hozzáadott képek mentésére a Firebase Storaget használtuk. Ez egy fájlátoló szolgáltatás, valamint nagyon hatékony és egyszerű.

Kiemelnénk még egy tool-t, ami miatt még a Firebase mellett döntöttünk, az Analytics, amely segítségével sok elemzési adat kinyerhető: hány aktív user van, melyek azok a top screenek, ahol a felhasználók sok időt töltenek, melyik országokban és milyen platformokon keresztül használják az alkalmazást.

4.4. Szekvenciális diagramok

4.4.1. Regisztráció



ábra 13 Regisztráció szekvencia diagramja

A felhasználónak az alkalmazás megnyitása után lehetősége van arra, hogy regisztráljon. A regisztrációkor a felhasználónak meg kell adnia különböző adatokat (keresztnév, vezetéknév, email, jelszó és jelszó ellenőrzés), amelyek hogyha nem helyesek vagy, ha üresen maradnak a mezők, akkor a felhasználó értesítést kap erről.

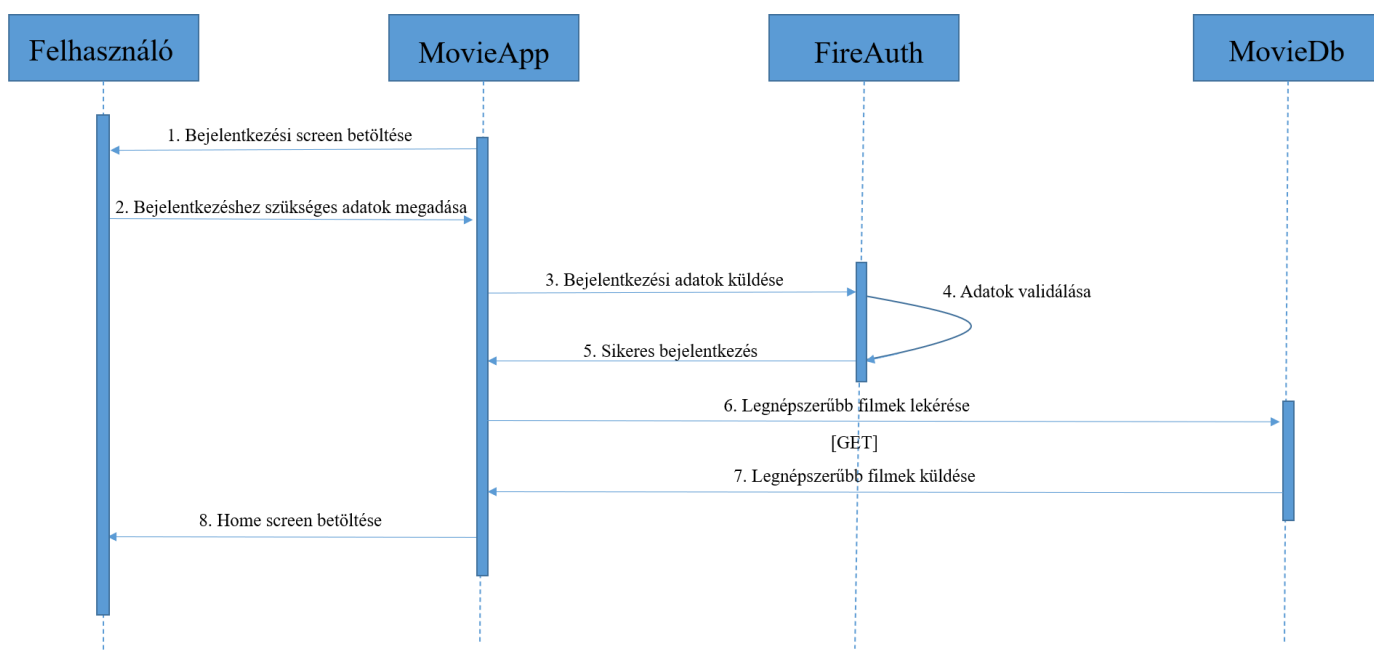
Ha a regisztráció sikeresen végrehajtott, akkor az adatok bekerülnek az adatbázisba, majd ezt követően megnyílik a bejelentkezés oldal. A regisztráció szekvencia diagramja az ábra 13-n részletesen le van vezetve.

4.4.2. Bejelentkezés

Amennyiben a felhasználó már rendelkezik fiókkal, úgy lehetősége van az alkalmazás indulása után bejelentkezni. Bejelentkezéskor a felhasználónak meg kell adnia az emailcímét és jelszavát.

A felhasználó, ahogy elküldte a bejelentkezési adatai, azok ellenőrizve leszünk a FirebaseAuth által, amennyiben az adatok helyesek létezik olyan email és jelszó páros, akkor a felhasználót a rendszer bejelentkezteti. Abban az esetben, ha nem találhatóak az adatbázisban a megadott bejelentkezési adatok az alkalmazás hibaüzenetet jelenít meg.

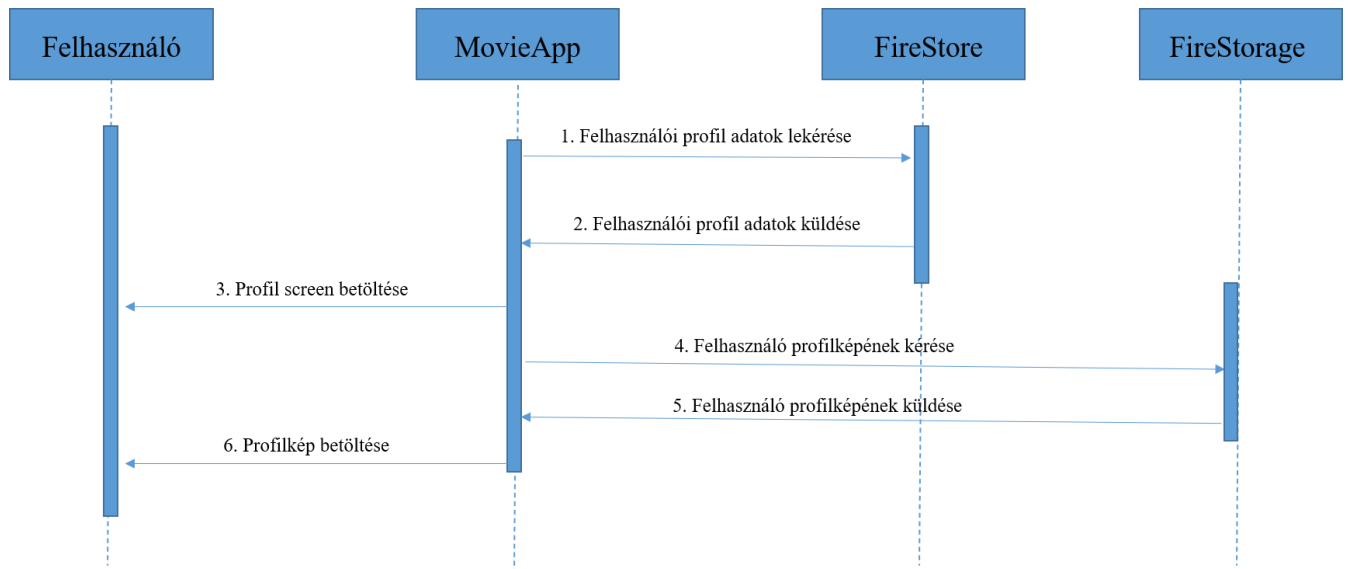
Amennyiben a bejelentkezés sikeres volt az alkalmazás a MovieDB adatbázisból a GET metódus segítségével lekéri a legnépszerűbb filmeket, majd megnyitja a Home screent, ahol a felhasználó megtekintheti a lekérdezés eredményét. A bejelentkezésről szóló szekvencia diagram az ábra 14 lett ábrázolva.



ábra 14 Bejelentkezés szekvencia diagramja

4.4.3. Profil

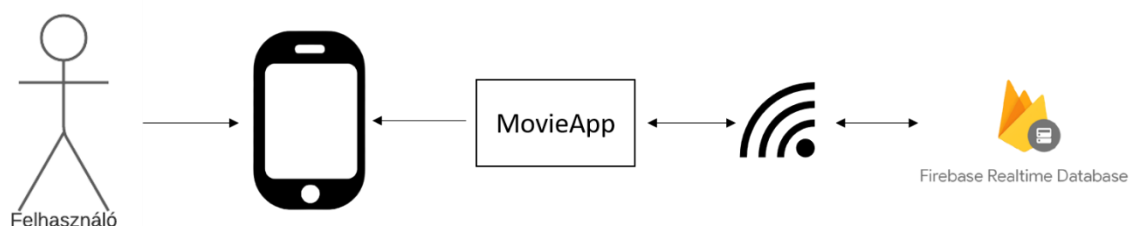
A Profil esetében az applikáció lekéri a FireStore-tól a felhasználó adatokat, az visszaküldi az applikációnak és ekkor betöltődik a Profil screen. A betöltés után a profilkép lekérése következik, amint ezt visszaküldi neki a FireStorage, úgy azonnal betöltődik a profilkép is. A profil esetében a szekvencia diagram az ábra 15 -n került illusztrálásra.



ábra 15 Profil szekvencia diagramja

4.5. Az alkalmazás architektúrája

Amint az ábra 16 is látható a felhasználónak szüksége van egy android készülékre, amelyre fel tudja telepíteni az alkalmazást. Ezt követően az okostelefonnak kell legyen internet hozzáférési lehetősége, hogy el tudja érni a Firebase adatbázist.

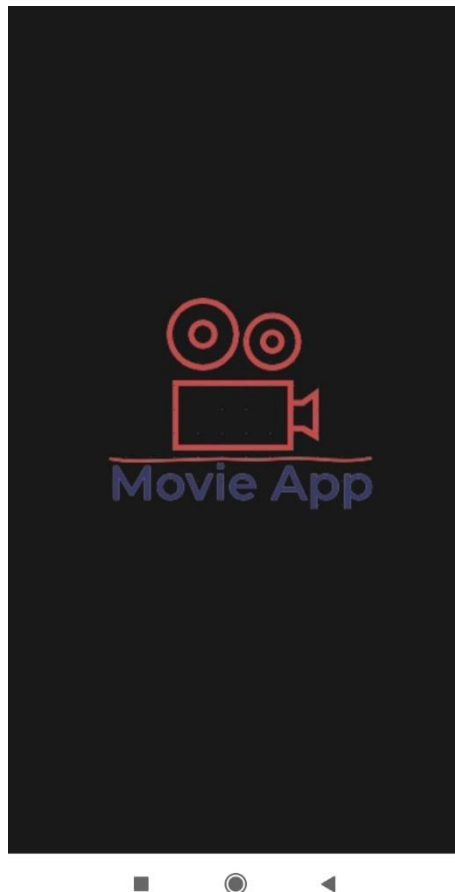


ábra 16 Alkalmazás architektúrája

4.6. Az alkalmazás működése

4.6.1. Splash screen

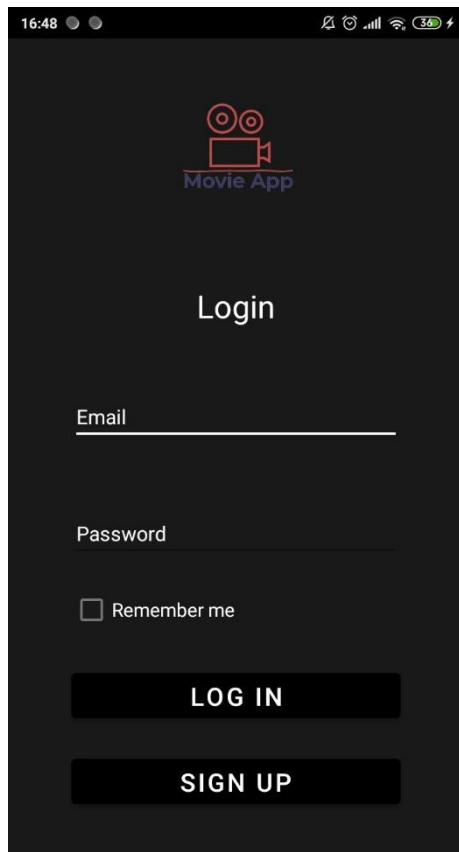
Az alkalmazás elindulásakor a felhasználó egy indulólappal találja szemben magát, amelyet 2 másodpercig láthat. Amint az ábra 17 is mutatja az indulólapon az applikáció logója látható. Amint letelt az fent említett pár másodperc az applikáció automatikusan a következő képernyőre vezeti a felhasználót, jelen esetünkben a Login screenre.



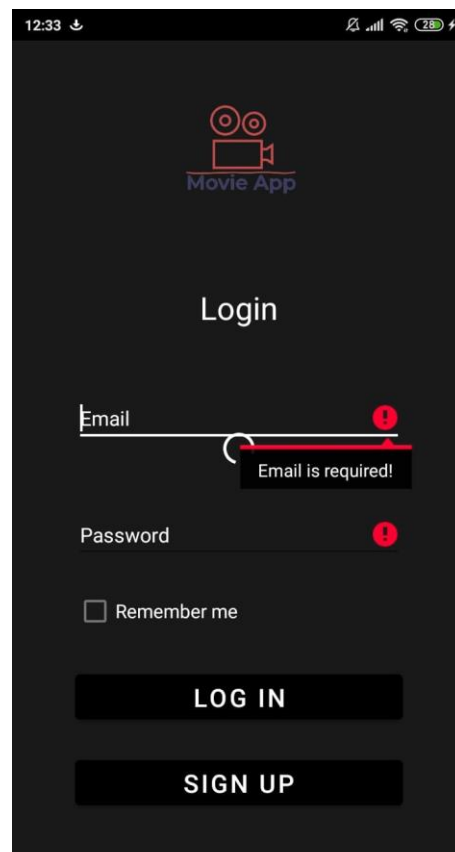
ábra 17 Splash screen

4.6.2. Login screen

A felhasználó a Bejelentkezés oldalon lehetőséget kap arra, hogy bejelentkezzen, amennyiben van már felhasználója. Az alábbi két kép közül az ábra 19 -n látható, hogy hogyan is néz ki ez az oldal. Láthatjuk, hogy a felhasználónak csak az email címét és a jelszavát kell megadnia. Míg az ábra 18 -n látható, ha a felhasználó nem tölti ki a mezőket, akkor megjelenítődik, hogy melyik mezőt hagyta üresen.



ábra 19 Login screen



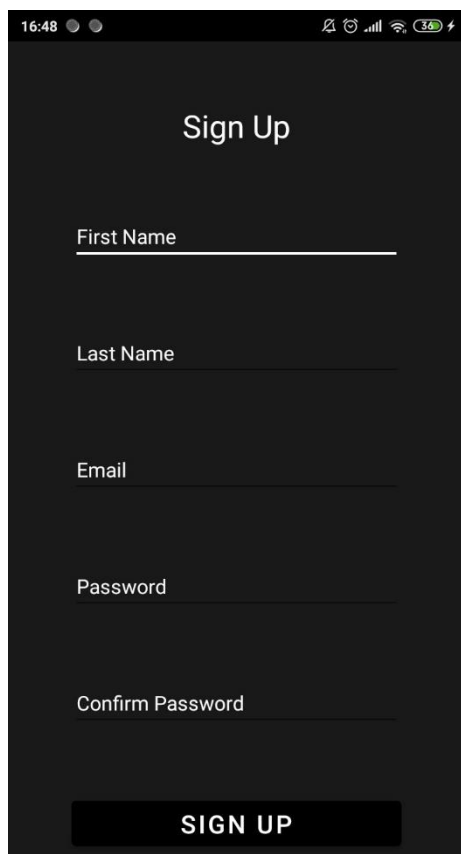
ábra 18 Login screen helytelen adatok
esetében

Az oldal alján látható két gomb a felső („Log in”) gomb lenyomásával a bejelentkezés történik meg abban az esetben, hogyha a mezők helyesen vannak kitöltve és található az email-jelszó páros az adatbázisban. A bejelentkezést követően az applikáció átviszi a felhasználót a Home screenre. Az alsó (Sign Up) gomb lenyomását követően a felhasználó a Regisztrációs oldalon találja magát.

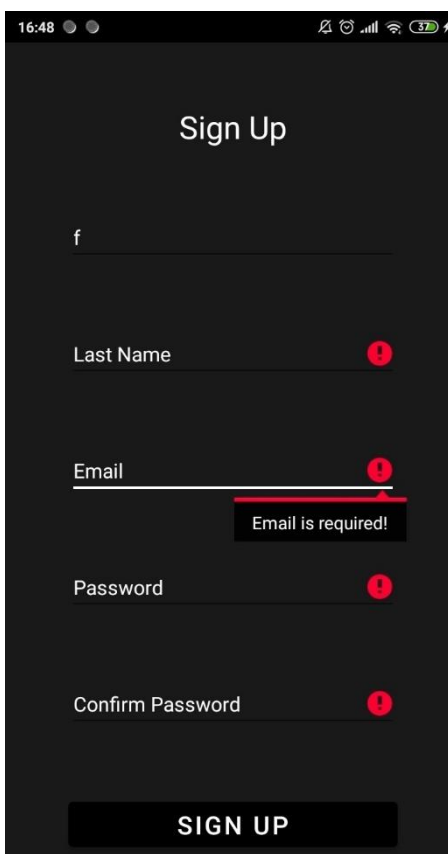
4.6.3. Registration screen

A Regisztrációs screen felépítésileg hasonlít a bejelentkezés képernyőhöz. A felhasználónak meg kell adni a keresztnévét, vezetéknévét, email címét, jelszavát és a jelszó ellenőrzéshez még egyszer a jelszót. Ezt követően a bejelentkezés („Sign Up”) gombra kattintva az alkalmazás beviszi az adatait a felhasználónak az adatbázisba, amennyiben az beírt adatok helyesek. Ha a regisztráció sikeres volt a felhasználót az alkalmazás a bejelentkezés oldalra navigálja, ahol be tud jelentkezni.

Az ábra 21-en és ábra 20– n látható, hogy hogyan néz ki a Regisztrációs oldal, valamint, hogy abban az esetben, ha a felhasználó nem tölti ki az adatokat, akkor ugyanúgy visszajelzést kap, mint a bejelentkezésnél.



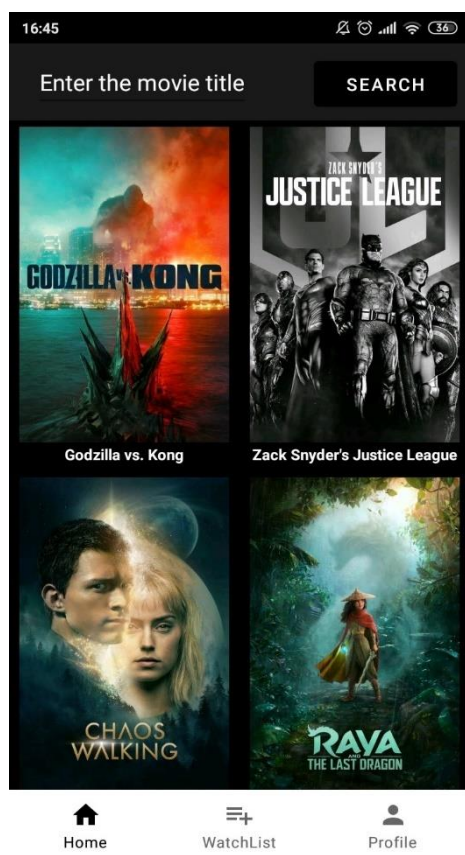
ábra 21 Registration screen



ábra 20 Registration screen helytelen adatok esetében

4.6.4. Home screen

A felhasználó a bejelentkezés után ezen az oldalon találja magát, ahol az oldal legtetején láthat egy kereső sávot és egy gombot, valamint az oldal többi részén filmek plakátjait láthatja, illetve azok címeit, mindezt népszerűség szerint rendezve. A Home screen kinézete ábra 23– on tekinthető meg. Az ábra 22-n az van reprezentálva, hogy ha beírunk egy szót a kereső sávba és rámegyünk a keresés („search”) gombra akkor kilistázza az össze olyan filmet, amelynek a címében szerepel a keresett szó.



ábra 23 Home screen

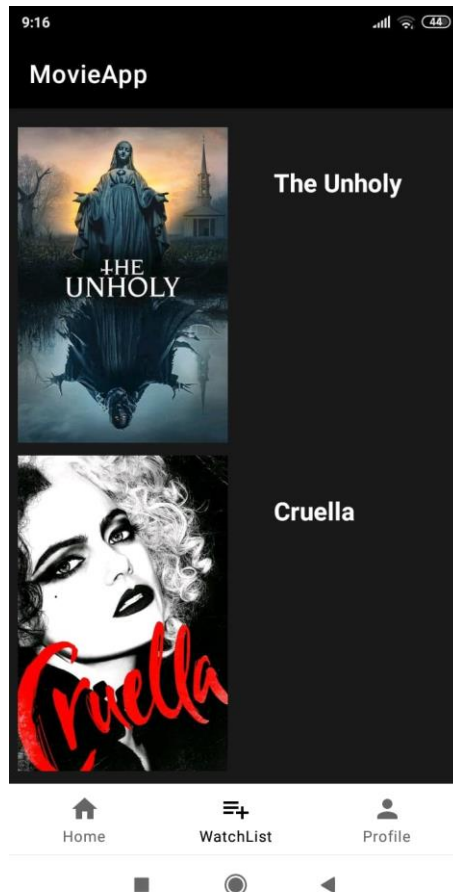


ábra 22 Home screen keresés esetében

Ezt az oldalt úgy is el tudja érni a felhasználó, hogyha a felhasználó bármelyik oldalon tartózkodik és rákattint a jobb oldali menüpontra a lap alján található menüsorból.

4.6.5. WatchList screen

A felhasználó úgy tudja elérni ezt az oldalt, hogyha a menüsor középső elemére rákattint. Ezen az oldalon a felhasználó azon filmek plakátját láthatja, amelyeket hozzáadott a megtekinteni kívánt filmek közé. Ennek az oldalnak a kivitelezése sokban hasonlít a Home screenhez, annyiban különbözik, hogy itt soronként láthatunk egy film plakátját és mellette a címét. Ez megtekinthető az ábra 24-en.

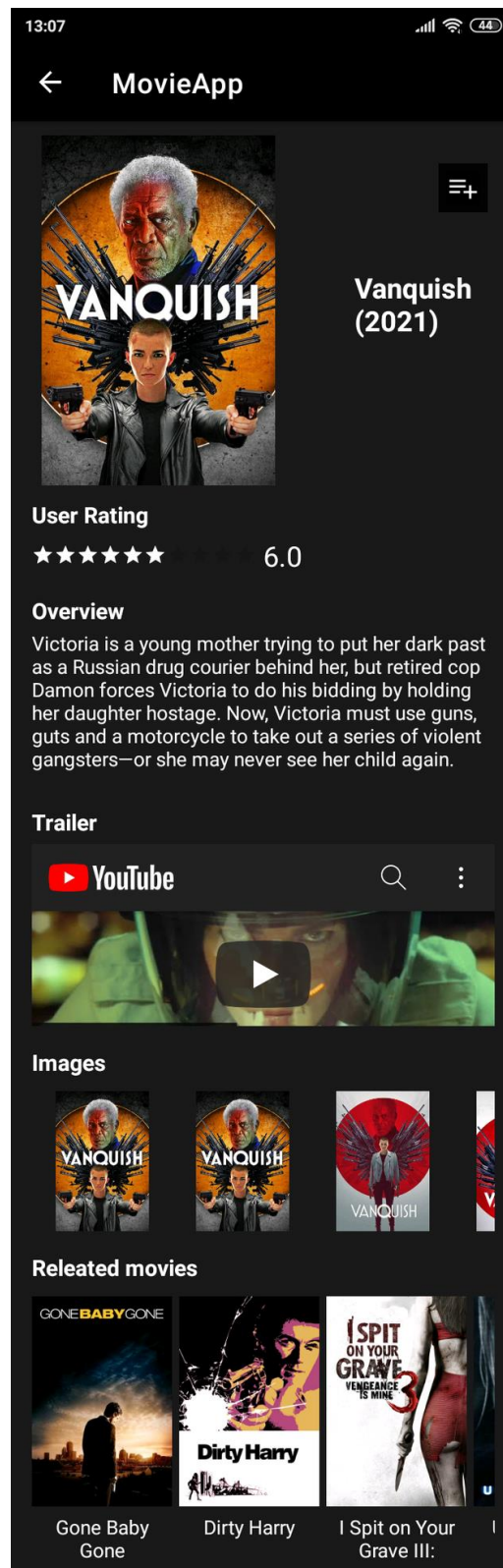


ábra 24 WatchList screen

4.6.6. Details screen

A screen elérhető a Home, valamint a WatchList screenekről, amennyiben a felhasználó kiválaszt egy filmet az ott kilistázott filmek közül. A Details oldalon amint az ábra 25 is mutatja a felhasználó legfelül láthatja a film plakátját, mellette a címét és a megjelenési évét. A plakát alatt egy értékelési sor látható, amely a filmre kapott szavazatok összegét mutatja. Mindezek alatt látható egy rövid leírás a filmről, majd egy előzetes videó látható a filmről, amely a youtube segítségével van bejátszva. Mindezek után van két oldalra görgethető lista, az elsőben a filmhez tartozó képek, míg a másodikban a megnyitott filmhez hasonló filmek plakátjai láthatóak.

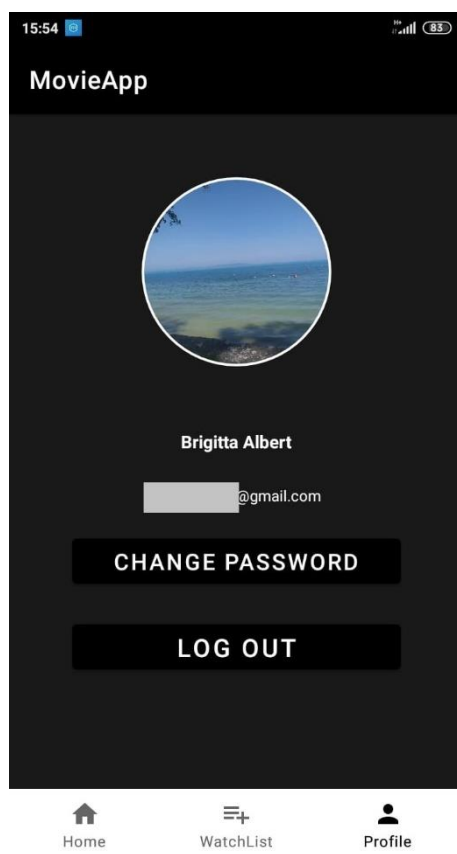
A Details oldal jobb oldalának legfelső részén látható egy ikon, amelyek hozzá tudjuk adni a megnyitott filmek a megnézendő videók listájához.



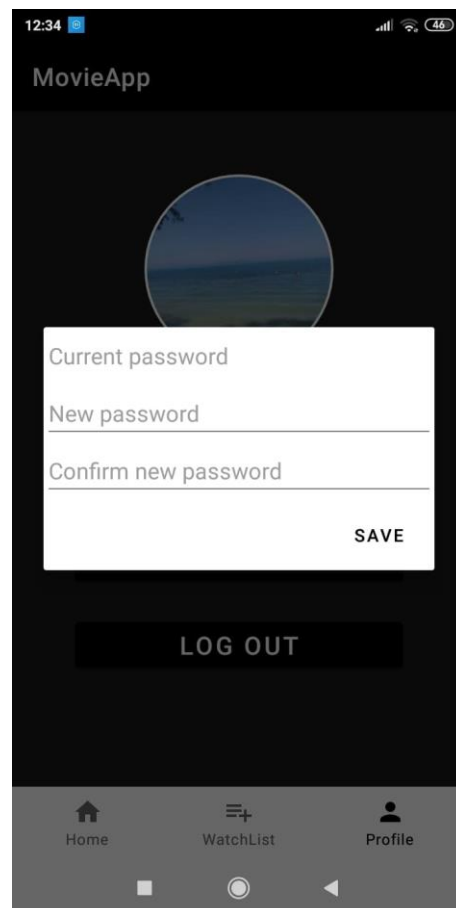
ábra 25 Details screen

4.6.7. Profile screen

Ahhoz, hogy elérje a felhasználó ezt az oldalt, csak a menüsor utolsó (jobb oldali) elemére kell rákattintania. A kivitelezése a szemléltető az ábra 27-en. A felhasználó ezen az oldalon láthatja a saját profilképét, ezt alatt a felhasználó nevét, majd az email címét. Mindezek alatt láthat két gombot is. A felső („Change password”) gomb lenyomásával lehetősége van megváltoztatni a jelszavát, ezt a régi jelszó megadásával illet egy új kiválasztásával tudja megtenni. Ez az ábra 26-on látható. Az alsó („Log out”) gombbal a felhasználó ki tud jelentkezni a felhasználójából. Az utolsó lehetősége ezen az oldalon a felhasználónak, az, hogy rákattint a profilképére és ekkor az applikáció megnyitja a telefon belső tárhelyét és így tud a felhasználó új képet választani.



ábra 26 Profil screen



ábra 27 Profil screen jelszóváltoztatás esetében

5. Összegzés és továbbfejlesztési lehetőségek

5.1. Összegzés

Az applikáció fejlesztése során gyakran ütköztünk problémákba, mivel egyikünk sem gyakorlott az Android, illetve a NoSQL adatbázisok világában. De a projektnek köszönhetően kipróbálhattuk magukat és végeredményben egy felhasználóbarát és könnyen kezelhető alkalmazást sikerült fejlesztenünk. Annak ellenére, hogy 2 személyes csapatban dolgoztunk a taskboard nagyon hasznos volt számunkra, mert így a fejlesztés is gördülékenyebben haladt, mindig átlátható volt a munkánk. Úgy érezzük mobilalkalmazás fejlesztés területén személyes fejlődést értünk el mindketten.

5.2. Továbbfejlesztési lehetőségek

- Filmek szűrése évszám, felhasználói szavazat, színészek, stb. alapján.
- A felhasználóknak lehetőséget adni a filmek értékelésére.
- A megnézendő filmek listáján a felhasználó csoportokat ("albumokat") legyen lehetősége létrehozni.
- Ajánlórendszer a felhasználók számára különböző metrikák alapján.
- Kiterjedés a sorozatok és a mozi világába.

6. Ábrajegyzék

ábra 1 Trello oszlopai	2
ábra 2 Ütemezési terv	3
ábra 3 API kulcs lekérése első lépés	5
ábra 4 API kulcs lekérése második lépés	5
ábra 5 API kulcs lekérése harmadik lépés	6
ábra 6 API kulcs lekérése negyedik lépés	6
ábra 7 Használati eset diagram	8
ábra 8 Prototípus első része	9
ábra 9 Prototípus második része	10
ábra 10 Felhasználó adatainak tárolása az adatbázisban	10
ábra 11 A regisztrációt aktiváló email	11

ábra 12 Adatok tárolása az adatbázisban	11
ábra 13 Regisztráció szekvencia diagramja	12
ábra 14 Bejelentkezés szekvencia diagramja	13
ábra 15 Profil szekvencia diagramja	14
ábra 16 Alkalmazás architektúrája	14
ábra 17 Splash screen	15
ábra 18 Login screen helytelen adatok esetében	16
ábra 19 Login screen	16
ábra 20 Registration screen helytelen adatok esetében	17
ábra 21 Registration screen	17
ábra 22 Home screen keresés esetében	18
ábra 23 Home screen	18
ábra 24 WatchList screen	19
ábra 25 Details screen.....	20
ábra 26 Profil screen jelszóváltoztatás esetében	21
ábra 27 Profil screen	21

7. Hivatkozások

<https://github.com/albertbrigi123/movieapp>

<https://themoviedb.org>

<https://developers.themoviedb.org/>

<https://www.figma.com/file/yhoLt2pa1kIF4Am8IUEiDG/MovieApp?node-id=0%3A1>