

SAPIENTIA EMTE
MAROSVÁSÁRHELY



ANDROID QUIZ APP

Készítette:

Izsák Andrea és Makkai Árpád

SZOFTVERRENDSZEREK TERVEZÉSE

2020

TARTALOMJEGYZÉK

1. BEVEZETŐ	3
2. KÖVETELMÉNY SPECIFIKÁCIÓ	4
2.1 Felhasználói követelmények	4
2.2 Rendszer követelmények	4
2.3 Funkcionális követelmények	4
2.4 Nem funkcionális követelmények	5
3. IMPLEMENTÁCIÓ	5
3.1 A projekt elkészítésének lépései	6
4. DIAGRAMOK	9
4.1 Osztálydiagramok	9
4.2 Use – Case diagram	10
5. AZ ALKALMAZÁS MŰKÖDÉSÉNEK LÉPÉSEI	11
6. FELMERÜLT PROBLÉMÁK / JÖVŐBELI TERVEK..	15
6.1 Felmerült problémák	15
6.2 Jövőbeli tervek	16
7. FELHASZNÁLT FORRÁS	17

1. BEVEZETŐ

Mindannyian tudjuk, hogy a XXI. században az olvasást, a társasjátékozást, valamint az enyéni fejlődést az okostelefonok használata váltja fel, úgy a gyerekek- mint a felnőttek körében. Naponta átlagosan 1 millió új, Android-ot futtató mobileszközt aktiválnak. Mára szinte bármit meg tudunk találni az okostelefonunkon, legyen szó könyvekről, tananyagokról, filmekről vagy játékokról, szinte mindent helyettesíteni tud a telefonunk.

Dolgozatunk fő célja, egy olyan androidos alkalmazás, amely korosztálytól független, bárkinek ajánlott, ezzel szeretnének segíteni fejlődni, főleg a fiatalabb generációnak.

Mivel nagyon az elején járunk az android programozásnak, alkalmazásunkra még jó néhány simítás vár, Mindezek ellenére igyekeztünk úgy összeállítani, hogy minél színesebb, képekkel tele legyen, hisz mindenkinek vonzóbb egy színes, vidám külalakú alkalmazás, mintsem egy szürke, egyhangú.

A projekt bővítésének jövőbeli terveiről a dolgozat végén részletesebben be fogunk számolni.

2. KÖVETELMÉNY SPECIFIKÁCIÓ

2.1 Felhasználói követelmények

- Ingyenes
- Könnyen kezelhető
- Legyen minél tartalmasabb

2.2 Rendszerkövetelmények

- Android készülék
- Android 10.0+
- Aktív internetkapcsolat

2.3 Funkcionális követelmények

- Regisztrálás
- Bejelentkezés
- Kérdéskategóriák kilistázása
- Játék
- Időszámláló
- Eredmények megtekintése
- Elért eredmények feldolgozása

2.4 Nem funkcionális követelmények

- Keretrendszer: Android Studio 4.0
- Nexus S API R emulator
- Adatbázis kezelő rendszer: FireBase Realtime Database
- Használt programozási nyelv: Java

3. IMPLEMENTÁCIÓ

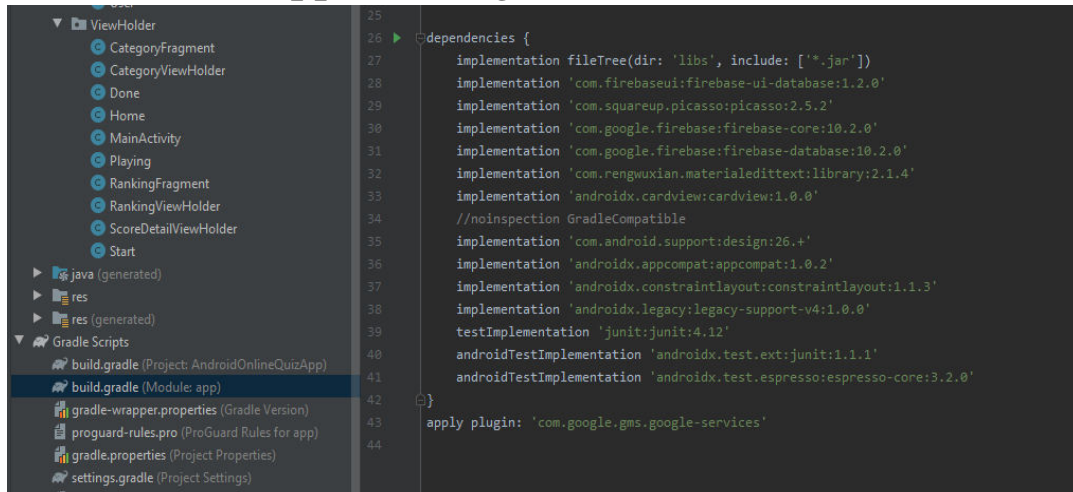
Az alkalmazást Android Studio-ban írtuk, mivel egy android alkalmazásról van szó. A játékot java nyelvben írtuk, a felületet, megjelenítést viszont .xml fájlokba.

Egy realtime adatbázisra volt szükségünk, ezért választottuk a FireBaset. Az adatbázisban tároljuk a felhasználókat, a kérdés kategóriákat, kérdéseket, egyszóval szinte mindent.

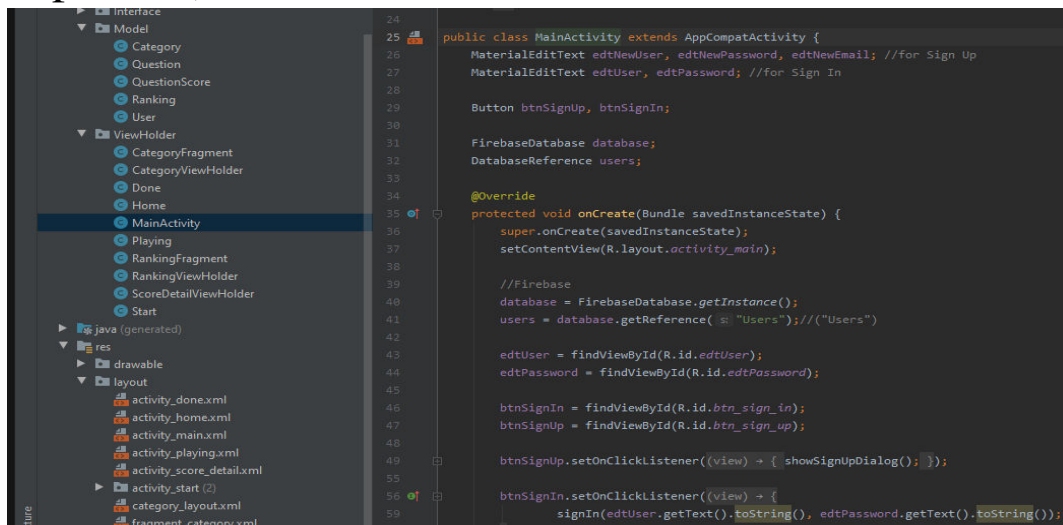
Nem Kellott mást tennünk, mint egy új projektet létrehozni FireBaseben is, és összekapcsolni az Android Studio-s projekttel.

3.1 A projekt elkészítésének lépései

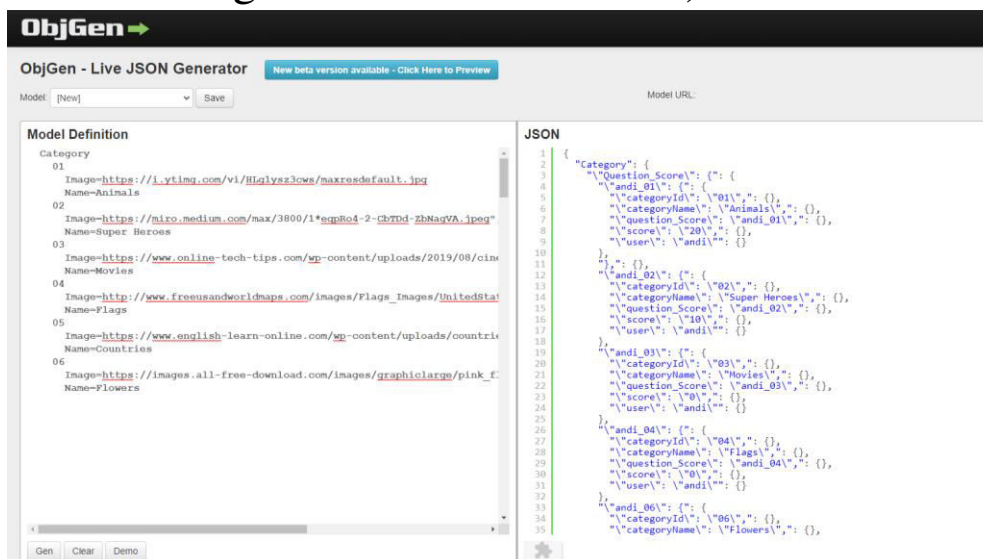
- új projekt létrehozása Android Studio-ban
- új projekt létrehozása FireBase-ben
- összekötés után az alábbi könyvtárak hozzáadása:
(FireBase Core, FireBase DataBase, Material Edit Text, Card View, Support Design)



- “USER” modell létrehozása (UserName, Password, Email adattagokkal), ezt követően layout a SignUp-hoz
- Main Activity, majd az activity_main.xml (a MainActivity-ben van implementálva az adatbázishoz való kapcsolat)



- Home Screen – A kategóriák és a pontozás betöltése (Load Category/ Load Ranking)
- “Home” Activity létrehozása, majd activity_home.layout
- Fragment mindkettőhöz, majd mindkettőhöz megjelenítés (fragment_category.layout, fragment_ranking_layout)
- A kategóriák feltöltése adatbázisba (ObjGen oldalon a Live JSON generator használatával)

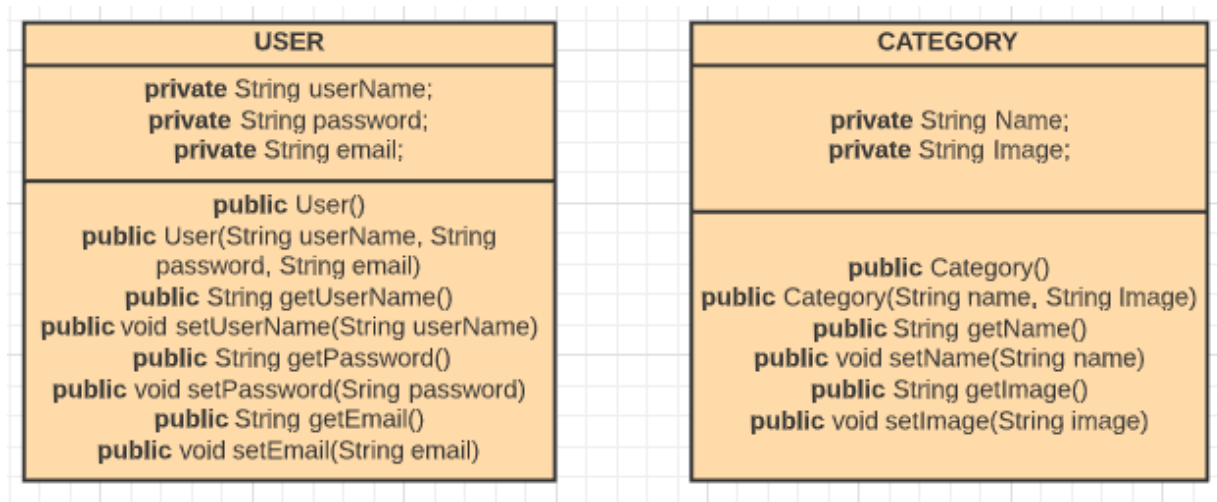
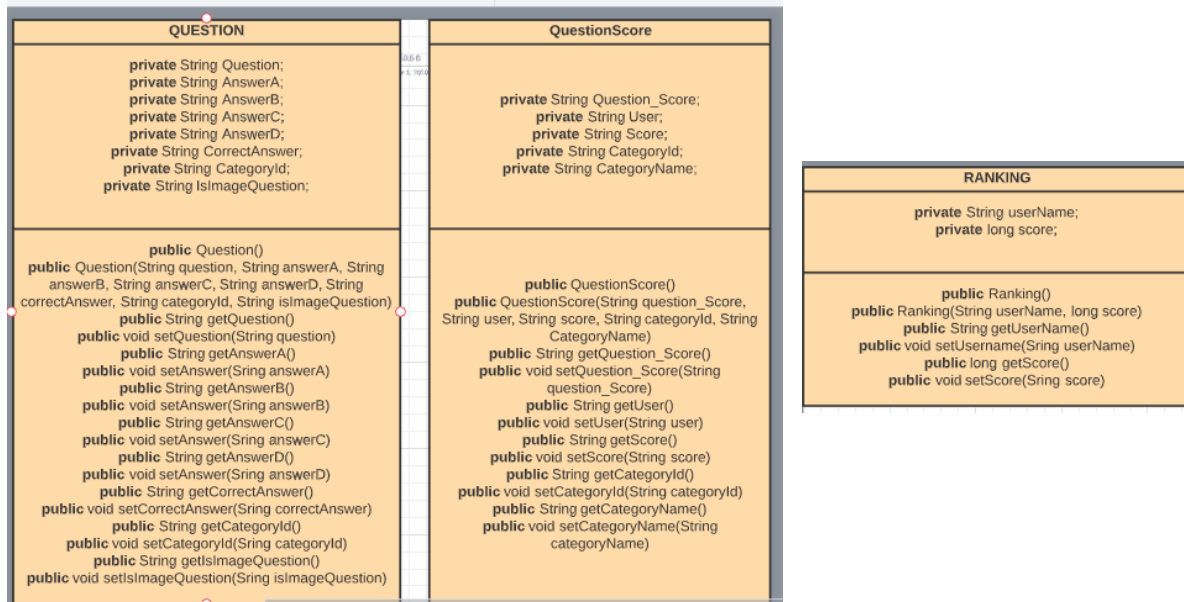


- A kapott json fájlt importálni kell az adatbázisba
- Category model létrehozása, itt elemezzük az adatokat, amiket kapunk az adabázisból, majd category_layout
- Category View Holder – felfolgoz minden elemet
- ItemClickListener interfész létrehozása az onClick implementálásához
- Mivel a kérdéseknél nem csak szöveget, hanem képeket is fogunk betölteni az internetről, ezért importálni kell a FirebaseUI és Picasso könyvtárakat

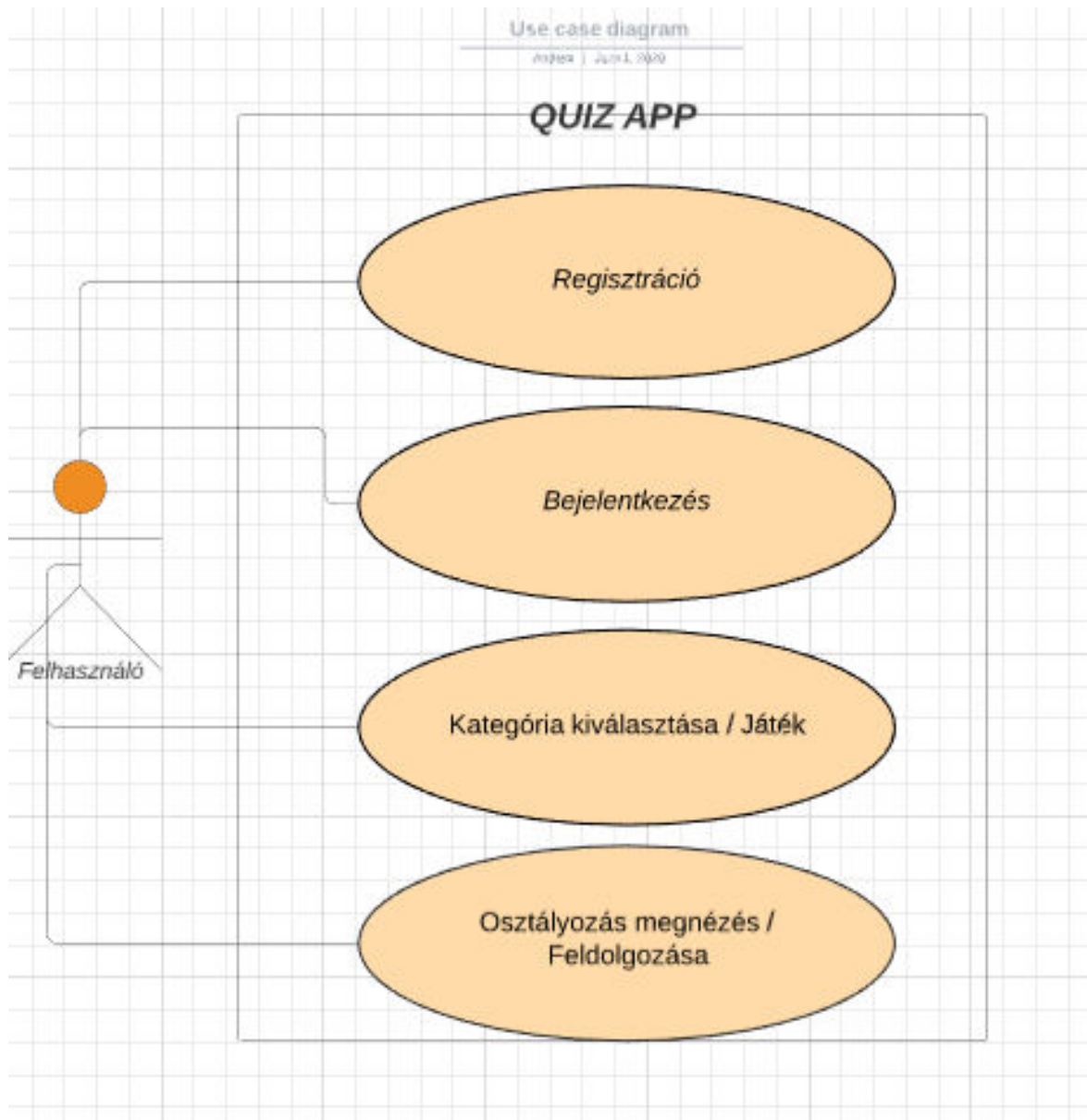
- A játék folyamán addig jönnek a kérdések, amíg véget ér a lista FireBase-ben, vagy, ha rossz választ adunk, kilép
- Létrehozunk egy Common.java állományt a globális változók tárolásához
- Question model létrehozása a kérdésekhez
- Exportáljuk a régi FireBase adatbázis json fájlját, és hozzáadjuk az új json-t a kérdésekkel
- Activity_start_layout – a képernyő a játékhoz
- Playing Activity és playing_layout, itt alkalmazzuk az ImageView és TextView tulajdonságokat ugyanarra a pozícióra, annak függvényében, hogy a kérdés kép vagy szöveg formában van megadva és alkalmazzuk rájuk a visible/invisible tulajdonságot
- Start.java – az összes kérdést sorra kilistázza, ha jó választ adunk, tovább lép a következő kérdésre, ellenkező esetben kilép
- A kérdések betöltésére használjuk a Done activity, majd létrehozuk az activity_layout-ot
- Ranking model, összeszámolja az összes összegyűjtött pontot a QuestionScore-ból és feltölti a Ranking-be
- RankingFragment, ranking_layout, RankingViewHolder
- ScoreDetailViewHolder, activity_score_detail_layout, amikor az osztályozás menübe lépünk, és kíváncsiak vagyunk, hogy egy adott felhasználó melyik kategóriákból hogy teljesített, itt lehet megtekinteni

4. DIAGRAMOK

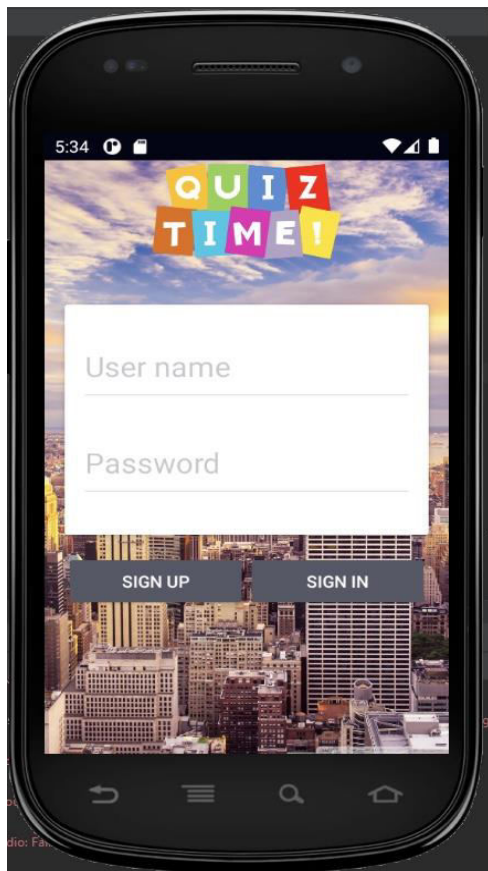
4.1 Osztálydiagramok(UML):



4.2 Use-Case diagram



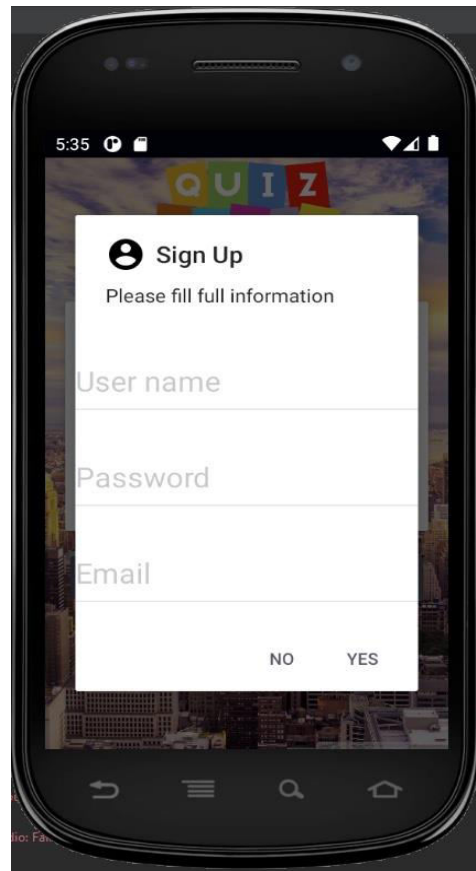
5. AZ ALKALMAZÁS MŰKÖDÉSÉNEK LÉPÉSEI



Kezdőoldal

Sign Up / Sign In

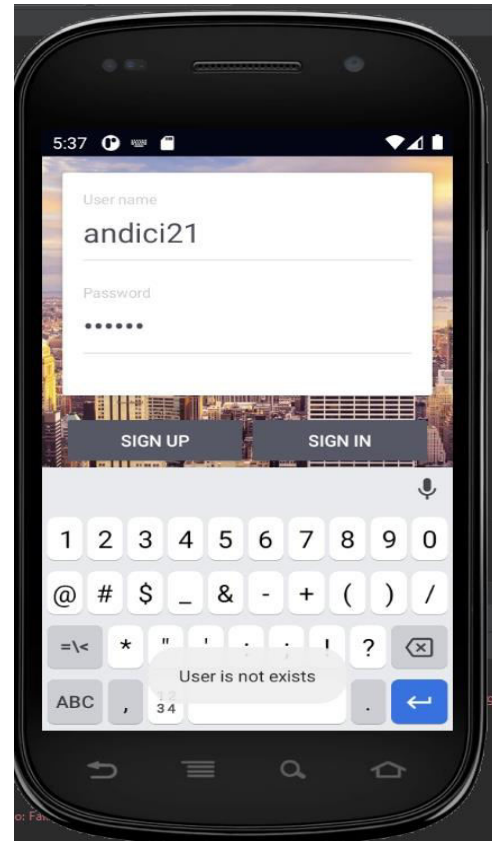
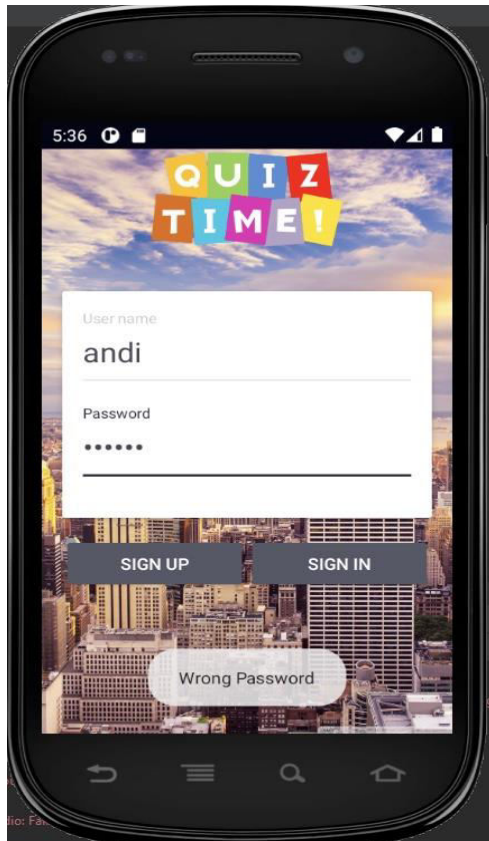
Lehetőséggel



Regisztráció

Felhasználónév, Jelszó és

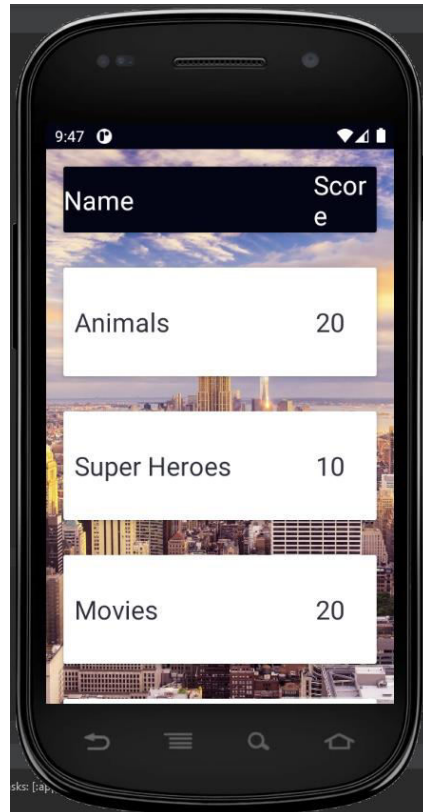
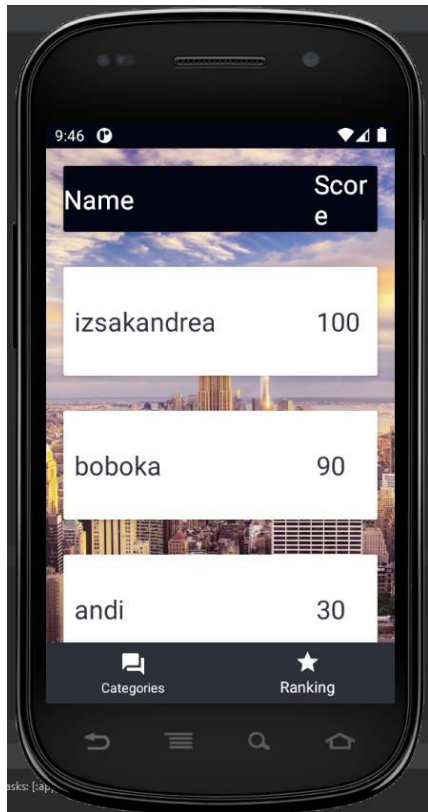
E-mail cím megadása



Belépéskor hibás jelszó esetén “Wrong Password” üzenetet, nem regisztrált felhasználó esetén “User not exists” üzenetet kapunk.



Belépés után a főoldalra kerülünk, ahol a kategóriák jelennek meg. Ezen kategóriák közül választhatunk, vagy a Ranking oldalra léphetünk, ahol a felhasználók eddig elért teljesítményét lehet követni.



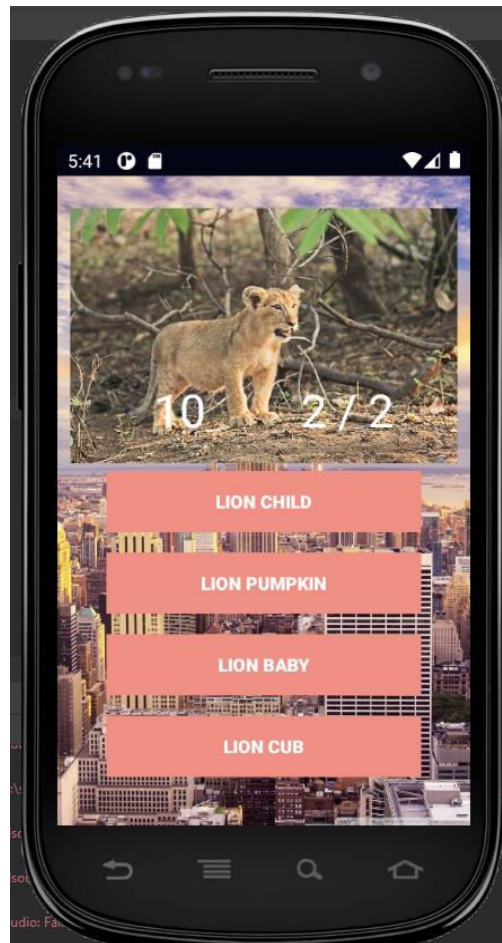
Ranking page

A felhasználók által összegyűjtött pontszámok. (1)

Egyes felhasználók statisztikái - melyik kategóriából, hány pontot sikerült elérniük. (2)



Miután kiválasztjuk a kategóriát, a játék indítása. (1) A játék lejártá után látjuk a pontszámot és hogy hány sikeres válaszunk volt, és hány kérdés. (2)



A fennebb említett kérdések típusai: kép vagy szöveg.

Ezeket a tulajdonságokat kell ellátnunk visible vagy invisible tulajdonságokkal.

6. FELMERÜLT PROBLÉMÁK JÖVŐBELI TERVEK

6.1 FELMERÜLT PROBLÉMÁK

Személyes tapasztalom alapján, a legnagyobb problémát az jelentette, amíg sikerült egy olyan számítógépet találnom, amelyen sikerül egyszerre Android Studio-ban szerkeszteni és emulátort használni, mivel nem rendelkezem android eszközzel.

Az interneten, ha utána olvasunk azt írja, hogy Android Studio használatára elég a 2GB Ram, de a 4GB Ram ajánlott. Sajnos nekem a 4GB Ram sem volt elég. Az Android Studio a processzort, az emulátor a Ramot terhelte le teljesen, szerencsére sikerült nagy kapacitású asztali számítógépet szereznem.

Kódolás során meg egy nagyon- nagyon apró hiba miatt teljesen megszakadt a kapcsolat a projekt és az adatbázis között, így már a regisztráció sem működött. Kis segítséggel sikerült rájönnöm, hogy bizony egy adattag elírása okozta ezt a problémát. A Main Activity-ben, edtPassword (Sign In) helyett edtNewPasswordot(Sign Up) használtam. Emiatt nem sikerült az elején a regisztráció.

6.2 JÖVŐBELI TERVEK

Ezt az alkalmazást egy általam használt játék sablonja alapján módosítanám.

A játék neve: WOW (Words of Wonders)

A játék lényege, hogy különböző országokba utazunk, ahol keresztrejtvényeket kell megoldanunk, amíg kigyűjtünk egy bizonyos pontszámot. Amint kigyűjtöttük a kellő pontszámot, átléphetünk egy következő országra, hogy felfedezzük azt.

Véleményem szerint, ez a Quiz-es alkalmazás nagyon hasznos lehet úgy tanulás-, mint élvezet/időtöltés szempontjából.

Lehetőségük lenne a felhasználóknak új dolgokat tanulni, vagy régi, elavult tudást feleleveníteni.

Emiatt lenne az elsődleges cél, hogy minél több kategóriával és kérdéssel bővüljön, hiszen ez csak egy kis bemutató az elképzelésből.

7. FELHASZNÁLT FORRÁS

<http://developer.android.com/studio/intro/>

<http://stackoverflow.com/questions/>

<http://youtube.com/>

https://app.lucidchart.com/documents#/templates?folder_id= (UML és Use- Case diagramok elkészítésére használt)