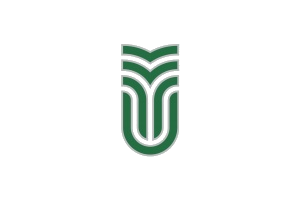
SapiAdvertiser

alkalmazás



|  |  |
| --- | --- |
| Készítette:  Farkas Zsombor  Rapa Csaba Erik  Salamon Evelin | Sapientia EMTE  Marosvásárhely 2018 |

Tartalom

[SapiAdvertiser alkalmazás 1](#_Toc533064336)

[Bevezetés 3](#_Toc533064337)

[Követelmény specifikáció 3](#_Toc533064338)

[Felhasználói követelmények 3](#_Toc533064339)

[Rendszerkövetelmények 3](#_Toc533064340)

[Rendszerkomponensek 3](#_Toc533064341)

[Firebase 3](#_Toc533064342)

[Android 3](#_Toc533064343)

[Felépítés 4](#_Toc533064344)

[Megvalósítás 5](#_Toc533064345)

[Következtetések 10](#_Toc533064346)

Projekt felelős:

Farkas Zsombor

# Bevezetés

Manapság már rengeteg féleképpen létrehozhatunk és hirdethetünk eseményeket, például hirdetőtáblák, tv reklámok, facebook események segítségével. A mi célunk, hogy létrehozzunk egy alkalmazást, amely segítségével összegyűjtjük az összes Sapientiás eseményt egy helyre, így megkönnyítve a diákok, tanárok és mindenki érdeklődő számára az informálódást az aktuális Sapientiás eseményekről.

# Követelmény specifikáció

## Felhasználói követelmények

1. bejelentkezés
2. regisztráció (telefonszám megadását követően az eszközre üzenet formájában érkezik egy megerősítő kód)
3. események listázása
4. események megtekintésének száma
5. új hirdetés hozzáadása
6. profil létrehozása és szerkesztése
7. profilkép megváltoztatása
8. hirdetések megosztása
9. hirdetések jelentése
10. saját hirdetések listázása
11. saját hirdetések törlése

## Rendszerkövetelmények

1. egy androidos készülék
2. Minimum SDK: Android 5 – Lollipop
3. Target SDK: Android 9 – Pie
4. Firebase Realtime Database,Authentication
5. Cloud Storage(képek tárolása)
6. internet
7. 6,5 MB tárhely

## Rendszerkomponensek

### Firebase

1. Firebase telefonszám alapú azonosítás
2. Firebase Real Time Adatbázis
3. Firebase Cloud Storage

### Android

1. Fragment, Activity, Broadcast
2. Constraint Layout, Frame Layout
3. RecycleView
4. Bottom Navigaton
5. Glide library
6. ViewPager
7. Alapértelmezett android widgetek
8. Vector Assetek - ikonok

# Felépítés



Ábra 1: a rendszer felépítése

Alkalmazásunkban a felhasználó csak a View-al áll kapcsolatban, tehát, ez a rész a User Interface ami kommunikál az alkalmazásunk és a felhasználó között. Ide tartozik minden Android komponens (Fragment, Activity, XML, stb.).A munkánk leginagyobb része itt történt.

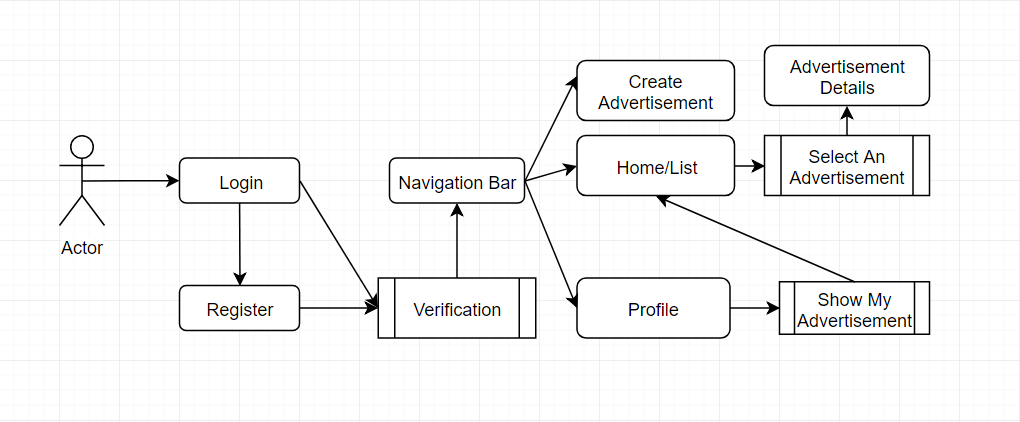
Adatok és

A Controllerünkhöz tartoznak az Adapter es RecycleView osztályok amelyek segítségével valósul meg a kommunikáció a View és a Modell között, itt történik az adatok feldolgozása és idegiglenes tárolása.

A Model alkotja az alkalmazás Backend részét, ide tartoznak a Firebase-el való kommunikáció, vagyis a User és Advertisement osztályok

A Broadcast segítségével figyeljük, hogy az adott eszköz csatlakozva van az internetre vagy nem, ha nem, akkor az alkalmazásban nem navigálhatunk, amíg újra nem leszünk csatlakozva az internetez.

# Állapot Diagram



Ábra 2: Usecase diagram

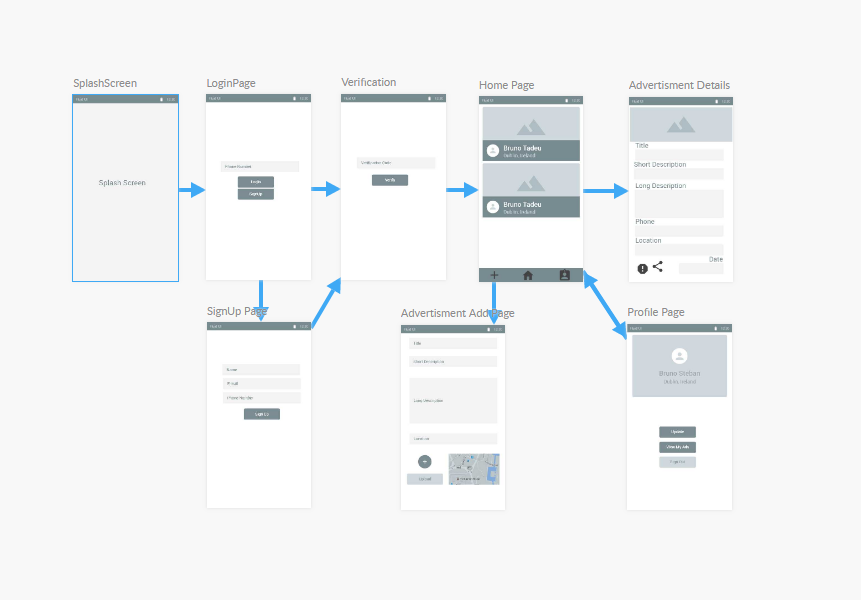
Az alkalmazásunkban először a Login részel találkozik, itt, ha már egy regisztrált felhasználóról beszélünk, akkor tovább tudunk lépni a phone authenticaton részhez (nem szükséges beírnunk a kapott kódot, mivel megoldottuk, hogy az SMS érkezésekor az alkalmazás tovább engedjen).

Ellenkező esetben a felhasználó regisztrálhat, ezután szintén verification történik.

Sikeres bejelentkezés vagy regisztráció után a fő oldal látható. A főoldal felépitése egy alul található navigációs menü és a hozzá tartozó moduláris komponens a fragment található. A navigációs menü által a felhasználó különböző állapotokba léphet át, mint például a saját profil megnézése, új hirdetéseket hozhat létre és megnézhati a saját hirdetéseit.

Minden állapothoz egy jól meg határozott feladat van rendelve. Minden állapot egy fragment vagy egy activity által van képviselve. Például ha a hirdetések listázás állapotban vagyunk és ki váltjuk a a hirdetés megnézésének állapotát akkor egy új fragment lesz betöltve ezzel el látva az állapot funkcionalitását.

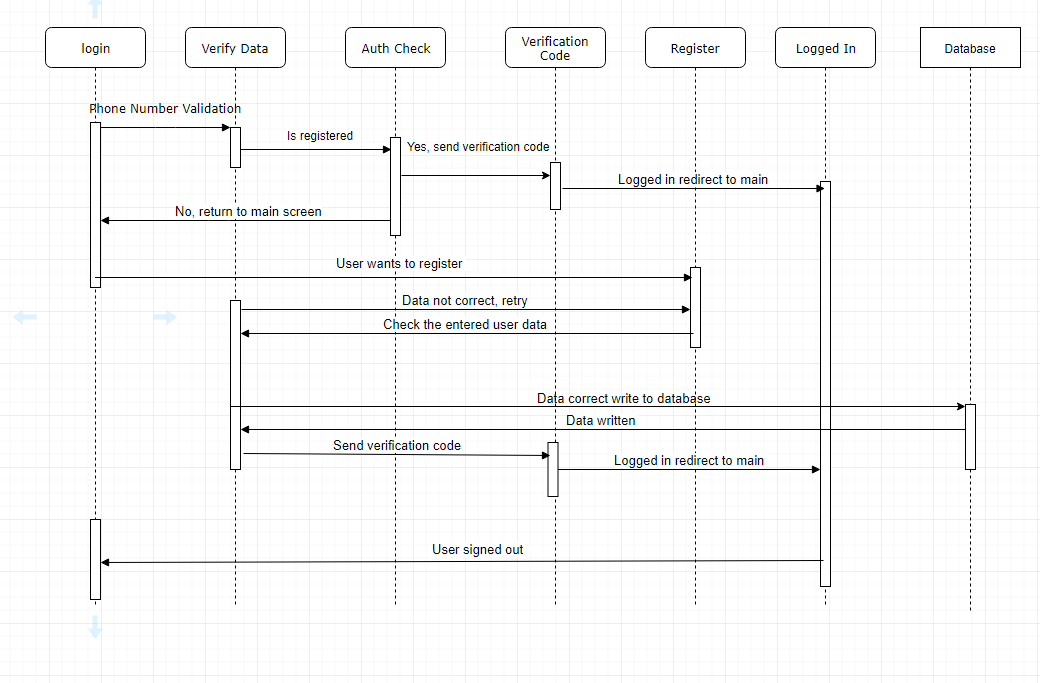
Drótváz rajz



Ábra 2.1: Drótváz rajz

A fenti diagramon, amely egyben az architektúrája az alkalmazásnak láthatjuk, hogy a különböző komponensek milyen kapcsolatban vannak egymással illetve melyik komponensből mely más komponensbe mehetünk át. Az alkamazás a SplashScreen - nel kezdődik. Ezek utána azt mondjuk, hogy be kell jelentkeznie a felhasználónak még akkor is ha nincs regisztrálva. A bejelentkezés után az alkalmazás teljes körű képét kapjuk meg amelyet tetszés szerint használhatunk.

Bejelentkezés szekvencia diagramja



Ábra 2.2.) A bejelentkezés regisztráció szekvencia diagramja

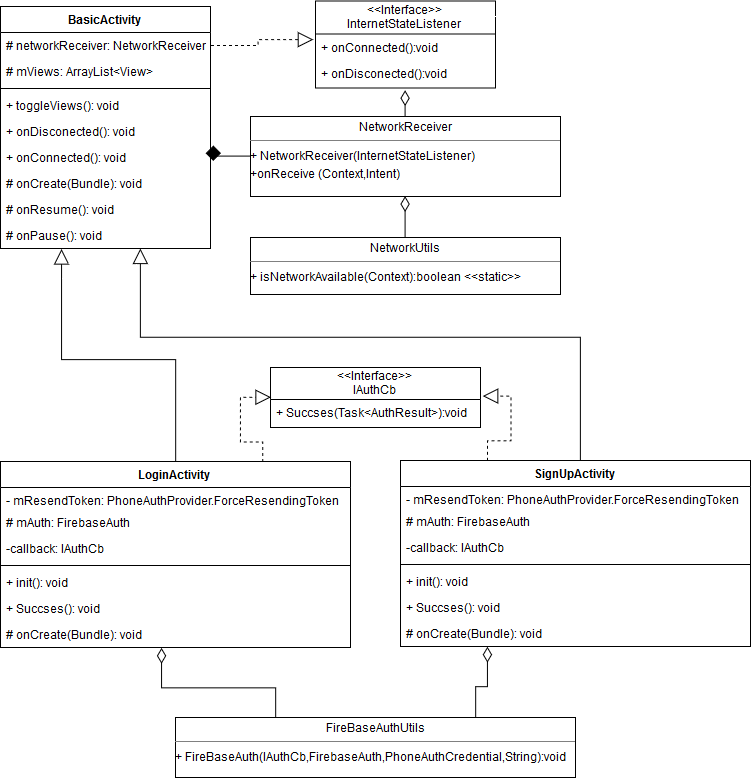
+-

A szekvencia diagramról a bejelentkezés és regisztráció folyamatát olvashatjuk le. Mivel a login screen megadja a lehetőséget, hogy a felhasználó bejelentkezen az alkalmazásba mondjuk ezt teszi meg. A kliens oldalon az információt ellenőrizzük, hogy megfelel az elvárásnak. Ezek után a FirebaseAuth-ban ellenőrzi, hogy bent van a felhasználó. Ha igen akkor a küld egy ellenőrző kódot, amelyet beirva és le ellenőrizve az alkalmazásba kerülünk.

Abban az esetben ha a felhasználó nincs bent a FirebaseAuth-ban, akkor vissza navigálva a login screen-re megkérhetjűk a felhasználót, hogy regisztráljon vagy ne tegyen semmit. A regisztrálás esetén megadja a nevét, e-mail-jét és telefonszámát. Ezen adatok szintén le ellenőrződnek és egy verifikációs kód beirássával már be is jelentkeztünk az alkalmazásba.

Felhasználó ha befejezte a munka menetét akkor a kijelentkezés opcióval vissza kerül a login állapotba.

Login rész osztálydiagramja



Ábra 2.3.) Login rész osztály diagramja

|  |  |
| --- | --- |
| Screenshot_20181220-004114  Ábra 3: Login oldal | Itt látható a Login Fragment, itt vagy megadjuk a telefonszámunkat(ha már regisztráltunk), vagy tovább navigálunk a SIGN UP gomb segítségével a regisztrációhoz.  Ha csak bejelentkezünk, akkor beírjuk a telefonszámot és hamarosan megkapjuk az azonosító SMS-t. |
|  |  |
| C:\Users\Eveliiin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20181220-093353.png  Ábra 4: Regisztrálás | A Regisztrációhoz szükséges egy felhasználónév, e-mailcím és a telefonszám, amely segítségével történik később a Phone Number Authentication. |
| Screenshot_20181220-004930  Ábra 5: Main oldal | Itt található a főoldal, ahol az eddig létrehozott események listája látható, itt nincs megjelenítve minden adat az eseményről, csupán a megadott két/képek(ujjhúzással lapozhatunk, ha több kép van),a feltöltő profilképe, a rövid leírás és a megtekintések száma. |
|  |  |
| Ábra 6: Esemény részletei | Rákattintva egy adott eseményre, megtekinthetünk minden megadott információt.  Egy adott eseményről minden lényeges információt megkaphatunk. Az események bejelenthetők, ha nem kivánatos tartalmat tartalmaznak, valamint meg oszthatjuk az ismerőseinkkel, valamely más applikációval amely képes küldés müveletre. |
| Ábra 7: Esemény részletei | Szintén a főoldalról a + gomb segítségével létrehozhatunk egy új eseményt. Itt a nevét, a hosszú- és rövid leírását, a helyet adhatjuk meg és feltölthetünk képeket is. Egy esemény feltöltése során egy prograssbar jelenik meg, amely a feltöltési feladat idejére felfüggeszti az alkalmazás mükődését. |

|  |  |
| --- | --- |
| Screenshot_20181220-004704  Ábra 8: Profil szerkesztése | A főoldalon található Profile gomb segítségével eljuthatunk a profil szerkesztését megvalósító oldalra, itt elehetőségünk van módosítani minden megadott adatot és lecserélhetjük a profilképet. |

# Következtetések és jövőbeli tervek

Összegzés képpen a projektünk egy már működőképes és jól használható Androidos alkalmazás, amely nagyrészt megfelel a tervezéskor elvártaknak, viszont mi a jövőben még bővíteni fogjuk különböző funkcionalitásokkal, mint például:

* Kommentek hozzáadása eseményekhez
* Térkép hozzáadása a helyszínekhez
* Összekapcsolás Facebookkal
* Landscape üzemmód megvalósitása

Ezek megvalósítása után egy jól használható, optimális alkalmazás lesz minden Sapientiás diák, tanár vagy bárki számára, aki kiváncsi az egyetemünkön lévő eseményekre és akár részt is szeretne venni ezeken.