

Face Trace

Team Face

Horváth István

Felkészítő tanárok: Esztelecki Péter, Kőrösi Gábor

Bolyai Tehetséggondozó Gimnázium és Kollégium, Zenta, Posta utca 18.

1. Bevezetés

A [Face Trace](https://github.com/Isti01/Face-Trace) (<https://github.com/Isti01/Face-Trace>) egy társkereső alkalmazás. Célja az, hogy hatékonyabbá tegye a társkeresést az által, hogy figyelembe veszi az algoritmus, hogy milyen kinézetű emberek tetszenek a felhasználónak. A jó algoritmushoz egy kitűnő felhasználói felület társul, azért, hogy az alkalmazás használata minél könnyebb és élvezhetőbb legyen.

A backendként a Google Firebase-t használja. A Firebase segítségével az alkalmazás akár több eszköz között is valós időben szinkronizálja az adatokat. A Firebase több-százezer felhasználót is elbír.

Az alkalmazás értesítést küld a felhasználónak, amikor üzenete érkezik, vagy amikor egy pár egymásra talál az alkalmazásban, így senki sem marad le a legújabb eseményekről.

A pároknak lehetőségük van chatelni, így megbeszélhetik az első találkát, hiszen a párkapcsolatnak a való életben kell, hogy történjen, az alkalmazásnak csak az a célja, hogy rátaláljunk a párunkra.

2. Az alkalmazás fejlesztésének a menete

2.1 Backend

2.1.1. Az adatbázis

Adatbázisként a [Cloud Firestore](#) került felhasználásra. A Firestore egy dokumentum-orientált NoSQL adatbázis. Az adatbázis kollektiókból (táblázat) és dokumentumokból áll (táblázati sor). A fő különbsége egy SQL adatbázistól az, hogy a dokumentumokban tárolt adatnak nincs előre meghatározott alakja vagy típusa, és nincs kapcsolat az adatok között. Az adatokhoz való hozzáférés kezeléséhez és adat validáláshoz biztonsági szabályokat kell írni.

2.1.2. Tárhely

A felhasználói fileok (képek) tárolására a [Cloud Storage](#)-t használja az alkalmazás. A tárhelyre feltöltött fileokhoz tartozó url-ek kerülnek tárolásra az adatbázisban, majd azok kiolvasásával lehet hozzáférni a tárolt tartalomhoz.

2.1.3. Hitelesítés, beléptetés, regisztráció

A felhasználói fiókokat a [Authentication](#) kezeli. Minden fiókhoz tartozik egy egyedi azonosító (uid), név, profilkép és email cím. Az adatbázisban az uid segítségével utalunk az adott felhasználóra.

2.1.4. Szerveren futó kód, szolgáltatások közötti kommunikáció

A szerveren az adatok aggregálásához az adatbázisban, az értesítések elküldéséhez, az arcok osztályzásához és a potenciális párok ajánlásához a [Cloud Functions](#) segítségével kerül a kód lefuttatásra. A függvényeket meg lehet hívni HTTP kérésekkel és adatbázis eseményekkel (pl dok. íráskor) is.

2.1.5. Értesítések kiküldése

Az értesítéseket a [Cloud Messaging](#) segítségével kapják meg a felhasználók. Az értesítéseket a készülékeknek lehet küldeni, nem a kliensnek. Minden fizikai eszköznek van egy egyedi azonosítója (token). A felhasználókhoz tartozó azonosítók az adatbázisban kerülnek tárolásra, majd, amikor egy felhasználónak értesítést küld a rendszer, akkor az adatbázisból kiolvasott tokenekhez tartozó eszközökre küldi el.

2.1.6. Arcok osztályzása, felhasználók ajánlása

Minden felhasználóhoz tartozik egy vektor, ami az arcáról generálódik regisztrációkor egy Cloud Function segítségével. A vektorokat a [Google FaceNet](#) egy implementációja ([facenet-pytorch](#)) generálja a feltöltött képből. Erre azért van szükség, hogy a felhasználók arcképét olyan formába hozzuk, hogy könnyen össze lehessen hasonlítani másokéval.

Ajánláskor a rendszer figyelembe veszi az érdeklődési kört, távolságot, életkort és figyelembe vesz azt, hogy mennyire hasonlít a személy azokhoz, akiket a felhasználó előzőleg kedvelt. A vektorokat úgy hasonlítja össze, hogy kiszámolja a két vektor közötti szög koszinuszát, tehát skaláris szorzat / két vektor hosszának szorzata, minél nagyobb a kapott érték, annál hasonlóbb a kettő. Az algoritmus ezek a paraméterek alapján készít egy rangsort, amit megkap a felhasználó.

2.2 A mobil alkalmazás

2.2.1. Felhasználói felület

Az alkalmazás a Flutter frameworkkel készült. A framework jó választás, különleges felhasználói felületek készítése esetén, ugyanis a Flutter segítségével minden egyes pixel az irányítása alatt van a programozónak. A

Flutter deklaratív, komponens alapú keretrendszer, könnyű egy megírt elemet újrahasonítani, több helyen is felhasználni az alkalmazásban.

A cél az volt, hogy az alkalmazást érdekes és könnyű legyen használni. Az alkalmazásnak a színe változtatható, egy új színnel egy új hangulatú alkalmazást kaphatunk. Navigációkor minimalisztikus, de ugyanakkor játékos animációk növelik az élményt.

Az alkalmazás nem csak több színben, hanem több nyelven érhető el, így több emberhez juthat el. Screenshotok 1. ábra.

2.2.2. Háttérlogika

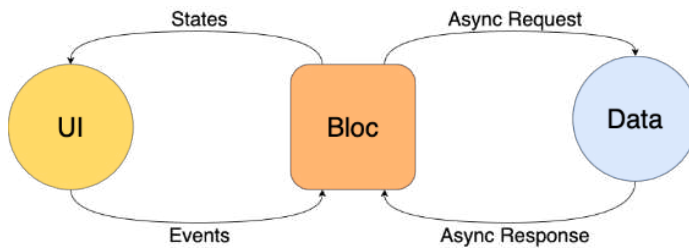
A logika, ami összeállítja, hogy mi jelenjen meg a UI-on, a [BLoC pattern](#) segítségével az alkalmazás valós időben reagál a bejövő eseményekre, legyen szó adatról, ami a backend felől érkezik, vagy akár egy gombnyomásról, ami a felhasználó felől érkezik.



1. ábra: Regisztráció, Kezdőlap, Felhasználó adatainak megtekintése, Saját adatok szerkesztése, Chat, A felhasználó párra talált

A BLoC egy absztrakt osztály. A legfontosabb elemei: A State Stream, az Event Stream, és az az eljárás (mapEventToState), amivel egy eseményből (Event) egy alkalmazás állapotot (State) hoz létre 2. ábra.

A Face Traceban több implementációja működik a BLoC-nak. Egy felel azért, backend által ajánlott emberek a képernyőre kerüljenek. Egy ellenőrzi a megadott adatokat és elvégzi a regisztrációt. Egy másik figyel, hogy éppen be van-e jelentkezve a felhasználó, és ha igen akkor szinkronizálja az adatait az adatbázissal. Egy nyilván tartja a chatszobákat és egy felel a chatelésért.



2. ábra: Bloc felépítése

3. Elért eredmények

Az alkalmazás fejlesztése egy jó élmény volt, sok tapasztalatot szereztem, és rengeteg dolgot megtanultam. Ez az első olyan munkám, ami a felhasználók körében megállna a helyét, az app egy használatra kész termék.

- Tanultam az arcfelismerés működéséről
- További tapasztalatot szereztem a machine learningről
- A cloud szolgáltatásról, a serverless infrastruktúráról
- Szerveroldali programázásról
- A NoSQL adatbázisokról
- Fejlődtem alkalmazáskészítés terén, megtanultam komplex kezelőfelületeket és animációkat írni.

A Face Trace a legnagyobb volumenű projektem, és minden eddig szerzett tudásomat felhasználtam benne.