ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS
INFORMATIKAI KAR

SZAKDOLGOZAT

OE-NIK Hallgató neve: **Karácsony Márton**
2021 Hallgató törzskönyvi száma: **T/005672/FI12904/N**

Óbudai Egyetem
Neumann János Informatikai Kar
Kiberfizikai Rendszerek Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: **Karácsony Márton**
Törzskönyvi száma: T/005672/FI12904/N
Neptun kódja: [redacted]

A dolgozat címe:

Sportegyesület működését támogató alkalmazás tervezése és fejlesztése
Designing and developing a application that supports a sport association

Intézményi konzulens: Nagy András
Külső konzulens:

Beadási határidő: 2021. december 15.

A záróvizsga tárgyai: Számítógép architektúrák
Big Data és üzleti intelligencia specializáció

A feladat

A szakdolgozat kapcsán a hallgató feladata egy a szervezet belső folyamatait támogató alkalmazás fejlesztése, melynek célja az ügyviteli és adminisztrációs feladatok támogatása és nyomon követése. Továbbá elérhetőnek kell lennie Android és IOS eszközökön is. Kutassa fel és vizsgálja meg milyen cross platform fejlesztésre alkalmas környezetek/nyelvek vannak jelenleg a piacon. Ismertesse a választott adatbáziskezelő rendszert és fejlesztői keretrendszert. Valósítson meg egy olyan alkalmazást, amely megfelel a megrendelővel közösen elkészítendő követelményspecifikációnak.

A dolgozatnak tartalmaznia kell:

- a feladat részletes ismertetését,
- a követelmény specifikációt,
- a rendszertervet, mely tartalmazza a funkciókat,
- a választott fejlesztői eszközök ismertetése és a választás indoklását,
- a továbbfejlesztési lehetőségeket.

Ph.

.....
Dr. Fleiner Rita
intézetigazgató

A szakdolgozat elévülésének határideje: **2023. december 15.**
(OE TVSz 55.§ szerint)

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

.....
intézményi konzulens

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 2021.12.11.....

.....
hallgató aláírása

KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun Kód: Tagozat:

Karácsony Márton..... [redacted]..... Nappali Bsc....

Telefon: Levelezési cím (pl.: lakcím):

[redacted]..... [redacted].....

Szakdolgozat / Diplomamunka[1] címe magyarul:

Sportegyesület működését támogató alkalmazás tervezése és fejlesztése

Szakdolgozat / Diplomamunka[2] címe angolul:

Designing and developing an application that supports a sport association.....
.....

Intézményi konzulens: Külső konzulens:

.....Nagy András..... .....

Kérjük, hogy az adatokat nyomtatott nagy betűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2021.03.04.	Feladatkiírás meghatározása	
2.	2021.04.08.	Féléves célok kijelölése	
3.	2021.04.30.	Irodalomkutatás befejezése	
4.	2021.05.06.	Féléves munka bemutatása	

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4 [3] tantárgy követelményét teljesítette, beszámolóra / védésre [4]bocsátható.

.....

Intézményi konzulens

Budapest, 2021.05.10.

---

[1] Megfelelő aláhúzandó!

[2] Megfelelő aláhúzandó!

[3] Megfelelő aláhúzandó!

[4] Megfelelő aláhúzandó!

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun kód: Tagozat:

Karácsony Márton..... [redacted] ..... **Nappali Bsc.**

Telefon: Levelezési cím (pl: lakcím):

[redacted]

Szakdolgozat / Diplomamunka[1] címe magyarul:

Sportegyesület működését támogató alkalmazás tervezése és fejlesztése

Szakdolgozat / Diplomamunka[2] címe angolul:

Designing and developing a application that supports a sport association

Intézményi konzulens: Külső konzulens:

Nagy András..... .....

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2021.10.05.	Féléves terv bemutatása.	
2.	2021.11.02.	Fejlesztés állapotának ismertetése.	
3.	2021.11.16.	Tesztelési eredmények bemutatása.	
4.	2021.12.07.	Kész munka bemutatása	

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4[3] tantárgy követelményét teljesítette, beszámolóra / védésre[4] bocsátható.

Budapest, 2021.12.08. .....

.....
intézményi konzulens

[1] Megfelelő aláhúzandó!
[2] Megfelelő aláhúzandó!
[3] Megfelelő aláhúzandó!
[4] Megfelelő aláhúzandó!

Tartalomjegyzék

1 BEVEZETÉS

A szakdolgozat egy sportegyesület életét támogató cross-platform mobilalkalmazás fejlesztésének folyamatát mutatja be és készíti el. Egy egyesület élete jól megfeleltethető egy KKV-s vállalkozásnak, ezért szükséges erőforrástervezési oldalról is megvizsgálni. Egy egyesület életében ahogyan, egy KKV életében is megvannak ugyanazok a területek, amiknek az összehangolása és transzparens működése elengedhetetlen a mindennapokban. Ugyanúgy szükséges egy jelenléti ív vezetése és adott esetben visszakeresése vagy ebből kimutatások készítése. Számos funkció megvalósítása szükséges, hogy teljes egészében megfelelő legyen az alkalmazás ilyenek például, a fizetés integrációk, automatikus számla kiállítások, chat, naptár és sok egyéb más is. Ezen funkciók mögé elengedetlen egy stabil, jól skálázható háttérrendszer. A háttérendszert illetően fontos paraméterek, hogy mindig elérhető legyen és tartalmazzon már előre elkészített folyamatokat, funkciókat, mint az authentikáció, adatbázis és reporting. Ezekre a feladatokra fontos a megfelelő BaaS rendszer kiválasztása.

1.1 Cross-platform alkalmazás fejlesztés

A 21. század technológiai fejlődésének zászlóshajója a mobilszektor. A telefonok manapság nélkülözhetetlenek az életben, ezek az eszközök az elsődleges kommunikációs eszközei az embernek. Az ilyen mértékű technológiai fejlődés két tech óriásnak köszönhető. A Google-nak és az Apple-nak, akik jelenleg a mobilpiacot, szinte teljesen lefedik. Egy Gartner nevű kutatócég szerint [1] a ma forgalomban lévő okostelefonok 99.5%-án Android vagy IOS operációs rendszer van telepítve. A két platform közötti eloszlásról már jóval érdekesebb következtetéseket lehet levonni. A két operációs rendszer közötti eladásokban hatalmas különbségek vannak 85%-15% a Google javára. Azonban csak Androidra fejleszteni egy alkalmazást költséghatékonyak tűnhet, bár hosszútávon nem jó döntés, mivel az App Store-ban (IOS-es platform alkalmazás áruháza), fele annyi letöltés van egy évben, mint a konkurenciánál, de mégis kétszeres profitot termelnek, mivel az IOS felhasználók hajlandósága a fizetős alkalmazások megvásárlására sokkal magasabb, mint a Play Áruház vásárlóinál. Amennyiben valaki mobilalkalmazás fejlesztésére adja a fejét számolnia kell mindkét platformmal. [2]

Egy mobilalkalmazás fejlesztése komplex és költséges folyamat. A költségek főleg azért olyan magasak, mivel mindkét operációs rendszerre meg kell írni az alkalmazást. A két platform technológiailag teljesen különbözik egymástól, ezáltal a fejlesztés során két külön fejlesztő csapatot kell alkalmazni, hogy mindkét áruházba elérhető legyen az applikáció. A két platform duplázza a fejlesztési költséget, mert ugyanazokat a munkafolyamatokat, mint például a fejlesztés, tesztelés, kitelepítés duplán kell elvégezni. Amennyiben az emberi erőforrás tekintetében nézzük akkor is kétszer akkora ráfordítás szükséges a két csapat miatt. Leginkább erre a problémakörre született megoldásként a cross-platform alkalmazásfejlesztés.

1.2 Mi az a cross-platform alkalmazásfejlesztés?

A cross-platform alkalmazásfejlesztés, nem csak egy hidat jelent az IOS-es és Androidos világ között, hanem akár PC-s asztali alkalmazásokat is lehet fejleszteni egy keretrendszer segítségével. Az eredeti kódbázis akár 80%-át is újra lehet használni. Tehát ahelyett, hogy új kódot írnának minden platformra, a fejlesztők ugyanazt a kódot használhatják újra mindenhol.

A keresztplatformos fejlesztésnek sok előnye van, melyek közül a leglátványosabbak a teljesítmény és a nem szokványos felhasználói felület a natív alkalmazásokhoz képest. Gyakran a keresztplatformos megoldások lassúak vagy nem néznek ki a mai trendeknek megfelelően, és nem rendelkeznek a natív alkalmazások összes előnyével és funkciójával. Az ilyen alkalmazásokat könnyedén fel lehet ismerni, mivel minden szempontból elmaradnak natív társaiktól. [2]

1.3 Milyen keretrendszerek léteznek?

Amennyiben valaki cross-platform mobilalkalmazást szeretne készíteni számos keretrendszer közül van lehetősége választani. Ilyen például a Xamarin, a Microsoft megoldása az egy kódbázisos technológiára. A Xamarin mögött a C# nyelv és a .NET keretrendszer van, bár ez szinte minden szempontból elmarad a két piacvezetőtől. Két lehetőség maradt a Flutter és a React Native. Első a Google terméke az utóbbit pedig a Facebook és annak közössége fejleszti.

1.3.1 React Native

A React Native sikere a webfejlesztőknél keresendő, mivel a React egy JavaScript keretrendszer, ami egy webalkalmazás készítésénél elengedhetetlen követelmény manapság. Nagy hírnek számított amikor a Facebook 2015-ben egy konferencián bejelentette, hogy a cross-platform rendszerek felé veszi az irányt. Sok webfejlesztő felsóhajtott, hogy nem kell egy újabb programozási nyelvet megtanulni, hanem elengedő lesz a jelenlegi JavaScript-es ismeretek mobilos vonalon fejleszteni. Nem véletlenül lett a React Native mottója a "Learn once, write anywhere" - azaz tanuld meg egyszer és írj bármire. [3]

A React Native-nek a sikere két részre bontható, egyik a fent említett JavaScript-es alapok. A másik pedig a natív teljesítmény, ami azt jelenti, hogy az alkalmazás ugyanazokat a natív platform API-kat használja, mint a többi alkalmazás.

1.3.2 PWA

Hosszabban Progressive Web Application, amely tulajdonképpen egy chromium alapú böngésző ablak kiexportálásaként is felfogható. Minden olyan technológia felhasználható benne, ami egy mai weboldalban, ezért is ilyen népszerű mivel a webfejlesztési technológiák már széleskörűen elterjedtek és sok fejlesztő által már nagyon régóta használatban vannak. Ez a legnagyobb előnye, mivel nincs szükség újabb nyelveket, keretrendszereket megtanulni, hanem ezzel az eszközkészlettel lehetséges akár mobilra is exportálni és alkalmazásként futtatni.

A probléma itt a sebességgel és a széleskörű használhatósággal van. A natív kódra forduló versenytársak jóval jobban teljesítenek UX (felhasználói élmény) szempontjából. Továbbá, számos egy alkalmazásban már megszokott funkció nem elérhető, mivel egy progresszív webalkalmazás nem fér hozzá a telefon hardvereihez, mint a giroszkóp, kamera és így tovább. Mindeközben a natív appokban erre megvannak a lehetőségek és a már kialakult best practice-ek. [4]

1.3.3 Flutter

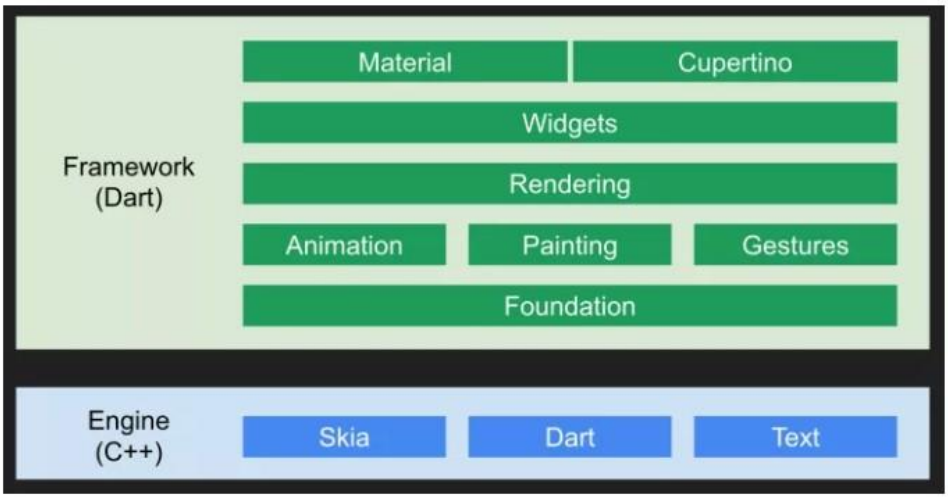
A Flutter 2016-ban látott napvilágot a Google jóvoltából. A Google ezt a keretrendszert egy cross-platform rendszernek tervezte és fejlesztette. Terveik szerint nem csak IOS és Androidos eszközökön lenne elérhető, hanem az idő előrehaladtával minden egyéb platformon is. Ezek a kezdeményezések jó irányba haladnak, mivel a sikeres IOS-s és Android-os megjelenés után újabb térhóditásba kezdet. Egyelőre gyerek cipőben, de a Flutter-nek a webes iránya is sok lehetőséget rejt. A Flutter 2 bejelentésével egyre több kapu nyílik az egy kódbázison alapuló fejlesztésre. Például az Ubuntu telepítők és a Fuschia, a Google következő generációs operációs rendszere is a Fluttert használja majd alkalmazásszintű keretrendszereként. [5]

1.3.3.1 Hot Reload

A Flutter támogatja az állapotfüggő Hot Reload-ot a fejlesztés során, ami a fejlesztési ciklus fellendítésének egyik fő tényezőjének tekinthető. A Stateful Hot Reload lényegében úgy valósul meg, hogy a frissített forráskódot a futó Dart VM-be injektáljuk anélkül, hogy a belső struktúrát megváltoztatnánk. Így a fájl mentésénél szinte azonnal látható az eredmény a csatlakoztatott készüléken vagy az emulátoron. [6]

1.3.3.2 A motorháztető alatt

A Flutter a Dart-ot, mint programnyelvet használja a komponensek építéséhez és a Skia 2D grafikus motort a legkülönbfélébb animációk és felhasználói felületek megjelenítésére. A Flutter-es ökoszisztémát 2 nagy részre lehet bontani:

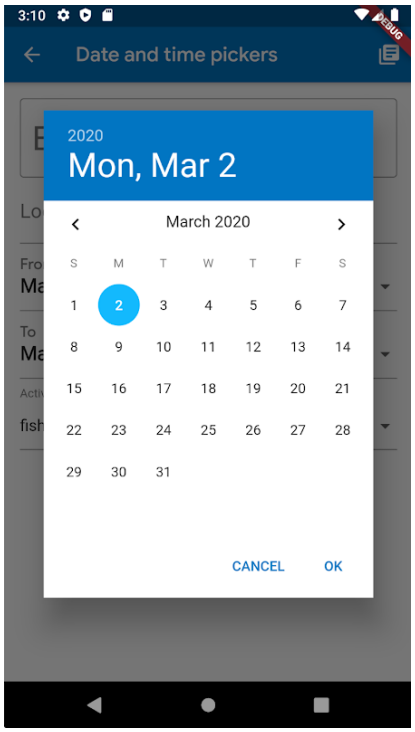


1. ábra - Flutter felépítése

Ahogyan az ábrán is látható a rendszer egyik nagy alkotóeleme egy modern React stílusú keretrendszer, ahol a legfelső szinten az előre definiált, a Flutter-rel jövő widgetek vannak. Mivel a két platform felhasználói egészen más felületekhez szoktak, így ezt a fejlesztőknek át kellett hidalnia, hogy natív élményt és a megszokott komponenseket lássák az Android és az IOS felhasználók is egyaránt. A Material az Androidos világnak lett kitalálva, a Cupertino pedig az IOS-es ökoszisztémának. Ezek között funkcionális különbség nincsen, csak megjelenésben.

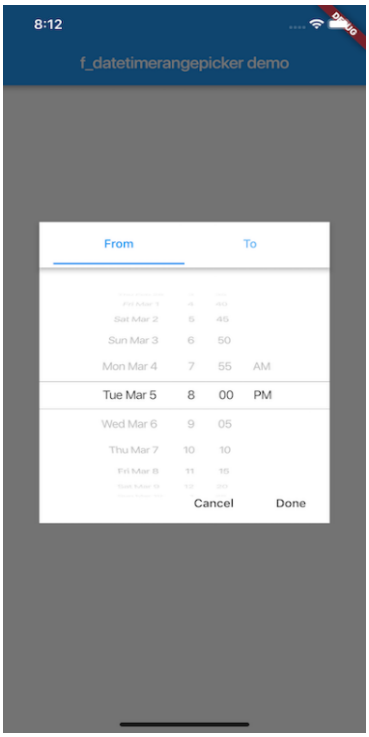
Például:

Material:



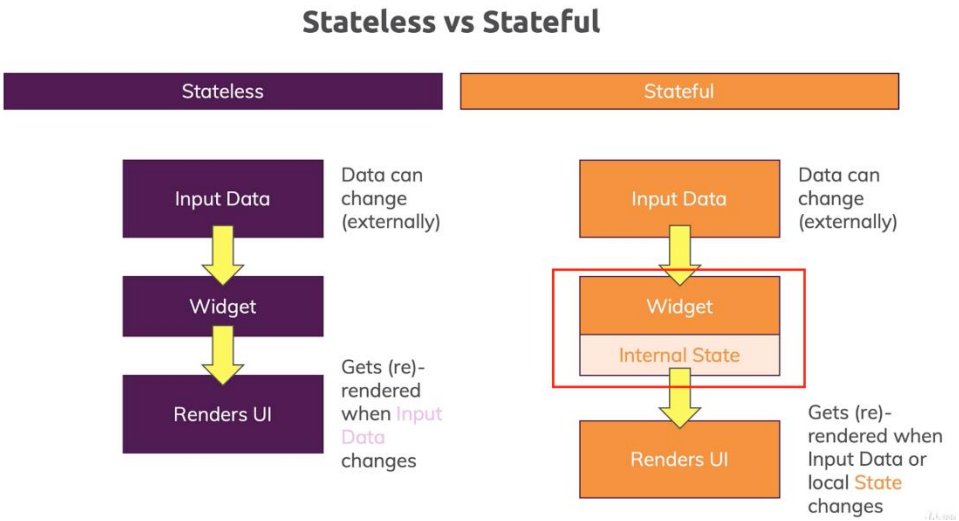
2. ábra - Material widget

Cupertino:



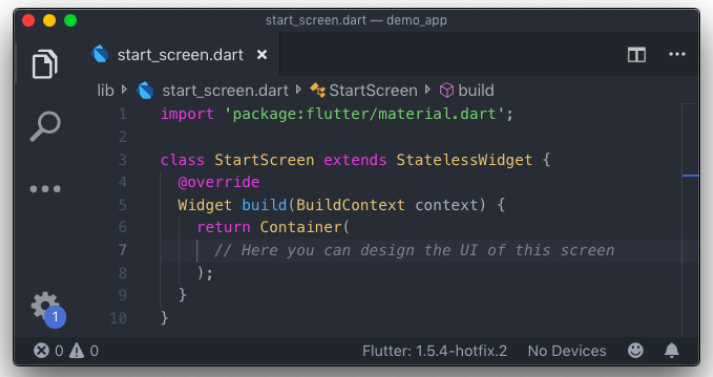
3. ábra - Cupertino widget

A widgeteknek vonzónak és érthetőnek kell lenniük, mert a felhasználó közvetlenül látja és használja őket. A widget a felhasználói felületen elhelyezkedő egyes komponensek, például egy legördülő menü, fénykép vagy egy szimpla gomb. A widgeteknek reactive-nak kell lenniük, azaz reagálniuk kell a felhasználói interakciókra, beleértve a renderelést és az animációkat is. Az OEM (gyári, az operációs rendszer által biztosított) widgetek újra felhasználása helyett, ahogyan azt a React Native teszi, a Flutter csapata úgy döntött, hogy saját widgetek-et biztosít. Ez azt jelenti, hogy a Flutter, mint platform, dönthet arról, hogy mikor és hogyan renderelődjenek a widgetek-et. A Flutter-ben a widgeteknek két fő típusa van, melyek a Stateless és a Stateful widget. [6]



4. ábra - Stateless widget és a Stateful widget különbsége

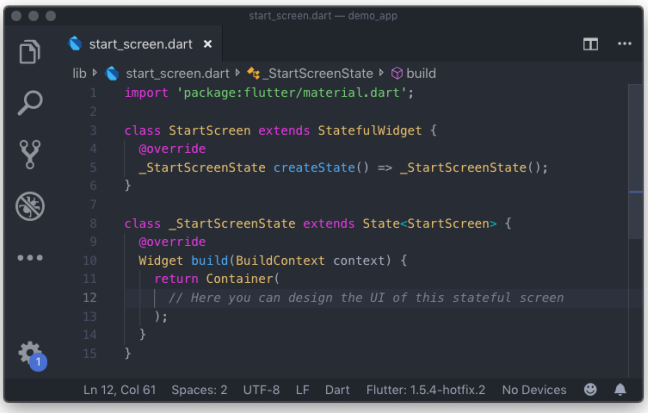
Stateless widget azaz állapot nélküli:
Nevéből adódóan a Stateless widgetek nem lehet újra rajzolni. Ez olyan felületeknél jó, amik főleg csak információ átadására szolgálnak és nem információ befogadására, mint például egy üdvözlő képernyő.



5. ábra - Stateless widget felépítése

Az osztályban felülírjuk a gyári build metódust a saját elképzelésünk szerint, megtervezzük a felhasználói felületet, majd a végén egy widget típussal térünk vissza.

Stateful widget azaz állapotfüggő:
Az állapotfüggő widgetek élettartalmuk alatt többször is kirajzolódhatnak.



6. ábra - Stateful widget felépítése

Itt egy egyszerű build metódus felülírásán kívül sok egyéb dolog is található. Például, hogy nem Stateless widget az ősosztály hanem Stateful. A StartScreen osztály itt nem a build metódust írja felül hanem a createState()-et úgy, hogy egy StartScreen példányt adjon vissza. Továbbá a _StartScreenState osztályban, pedig a Stateless-hez hasonlóan az alkalmazás felhasználói felületét lehet megvalósítani. A komolyabb logikát megvalósító oldalakhoz mindenképpen Stateful widgeteket kell használni. Ahhoz pedig, hogy minden rétegzési alapelvnek eleget tegyünk érdemes a bloc nevezetű Flutter-es csomagot is használni, amely az aszinkron lekérésektől, állapotok kezelésig mindenre lehetőséget biztosít. [7]

2 Mi az a BaaS?

Hosszabban Backend as a Service, amely arra szolgál, hogy a fejlesztő csapatnak ne kelljen az alkalmazásnak a Backend rendszerét is megírnia, elég legyen csak a Frontendet implementálnia. Ezeknek a szolgáltatásoknak a legnagyobb előnye, hogy a háttérfolyamatok, például egy authentikációs folyamat, már szinte kulcsra készen implementálva van benne így nem szükséges a "kerék újra feltalálása" a fejlesztés során. Ilyen szolgáltatások még például a Push értesítéses rendszer és az adatbázis is.

Gondolhatunk úgy egy alkalmazás fejlesztésére BaaS-szolgáltató használata nélkül, mint egy film rendezésére. Egy filmrendező a filmben megjelenő jelenetek tényleges forgatása és rendezése mellett, felelős a stábok, a világítás, a színészválogatás és a gyártási ütemterv irányításáért is. Most pedig képzeljük el, ha létezne egy olyan szolgáltatás, amely gondoskodik az összes színfalak mögötti tevékenységről, így a rendezőnek csak a rendezést és a jelenet leforgatását kellene elvégeznie.

Ez a BaaS ötlete: A szolgáltató gondoskodik a "fényekről" és a „kameráról”, vagyis a szerveroldali funkciókról, így a rendező, azaz a fejlesztő csak az "akcióra" koncentrálhat - arra, amit a végfelhasználó lát és tapasztal.

A Frontend és a Backend kéz a kézben jár és ennek a kettőnek a megfelelő együttműködése és összehangolása a fejlesztés során egy nagyon kemény feladat. Ezt a problémát a BaaS rendszerek API (Application Programing Interface) -n keresztül valósítják meg. Tulajdonképpen ez a híd a Frontend és a Backend között. Ami még nagyságrendekkel lerövidíti a fejlesztést, azok a szolgáltató által biztosított SDK (Software Developent Kit). Ezeken az SDK-on keresztül az összes olyan Backend funkciót könnyedén lehet implementálni, anélkül, hogy nekünk megkellene írni a nulláról. További hatalmas előnyük a BaaS rendszereknek, hogy nem kell szervereket, virtuális gépeket létrehoznunk és bérelnünk külön az alkalmazás üzemeltetéséhez, mivel ezek a szoftvercsomag részeként elérhetőek. [8]

2.1 Milyen alapfunkció vannak egy BaaS-nak?

- Adatbázis-kezelés
- Felhőalapú tárolás (a felhasználók által generált tartalmakhoz)
- Felhasználói hitelesítés
- Push-értesítések
- Távoli frissítés
- Analitikák
- Teszt funkciók

Ezen kívül van még sok apróbb gyártóspecifikus funkció is. [9]

2.1.1 Szervermentes kialakítás

Korábban amennyiben valaki alkalmazás fejlesztésre vállalkozott és szerette volna, hogy egyes kliensek között valamilyen adatáramlás történjen, akkor biztos, hogy komoly költségekkel kellett számolnia a szervert illetően. Tehát, hogy ha egy BaaS szolgáltatást beemelünk már a tervezés során az alkalmazás ökoszisztémájába, egyből sokat csökkenthetünk a költségeinken.

2.1.2 Skálázhatóság

Egy igazából csak a jövőben előjövő probléma. Ahogyan az alkalmazásunk növekszik egyre többen töltik le, ez egyenes arányos azzal, hogy sokkal nagyobb adatot kell eltárolni és továbbitani. Mivel egyre több kérés érkezik a szerverünk felé, így egyre többet kell kiszolgálni. Amennyiben saját magunk üzemeltetnénk a szervert is, akkor ez igen komoly fejfájást okozna, de egy BaaS rendszer a skálázhatóság problémáit is leveszi a vállunkról. Ár érték arányban sokkal jobban megéri használat után kicsit többet fizetni havonta egy

szolgáltatónak, mint méregdrága újabb és újabb szervereket vásárolni és magunk fenntartani.

2.2 Piaci alkalmazások

2.2.1 AWS Amplify

Az Amazon megoldása 2017-ben került piacra, amely egy viszonylag fiatal BaaS megoldásnak számít, de igencsak nagy fejlesztői közösség van mögötte és aktívan fejlesztik a mai napig is.

Az Amplify-ban megtalálhatóak az alapkövetelményeknél szabott funkciók. Mint például a felhasználói Authentikáció, Push értesítés és így tovább. Az Amazon az Amplify-t nagy és komoly alkalmazások BaaS szolgáltatásának szánta. Elég komoly tanulási görbéje és nem túl felhasználóbarát kialakítása van. Viszont lehetőség van innovatívabb szolgáltatások fejlesztésére, mint a képfelismerés vagy beszéd átirat készítés. Egyértelműen nagyobb volumenű mesterséges intelligenciát és gépi tanulást használó projekteknél érdemes használni. Az AWS Amplify-t a funkciók széles választéka és a használat után való fizetős ár képzési modellje miatt dicsérik. [10]

2.2.2 Back4App

Amikor 2017-ben a Facebook megszüntette az akkoriban vezető BaaS-eszközt, a Parse-t, egy lelkes fejlesztői közösség átvette a nyílt forráskódú változatot, a Parse Server támogatását és karbantartását. Azóta több modern BaaS-szolgáltató is a Parse Serverre építette platformját, közülük a Back4App a legkiemelkedőbb.

A Back4App SDK-t kínál Swift, Android Studio, JavaScript, React Native, GraphQL és más elterjedt fejlesztési keretrendszerekhez. A mobil backend-fejlesztési eszköztárában Push-értesítések, automatikus e-mailek, közösségi hálózati bejelentkezés és fájltárolás is szerepel.

Az egyetlen szépséghiba, hogy a Back4App nagymértékben támaszkodik a hivatalosan megszűnt Parse-ra, amelynek stabilitása és továbbfejlesztése most már kizárólag a fejlesztői közösség elkötelezettségétől függ. [11]

2.2.3 Apple CloudKit

A CloudKit is egy BaaS keretrendszer, amelyet az Apple az iOS 8 frissítés részeként adott ki. Az iOS, macOS, watchOS és tvOS számára elérhető megoldás célja, hogy egyszerűsítse az alkalmazások integrációját a szerveroldali iCloud-tárolóval.

Az Apple CloudKit legfontosabb funkciói közé tartozik a felhasználói hitelesítés, a Push-értesítések, a biztonságos adat- és fájlmegosztás, valamint a strukturált eszköztárolás

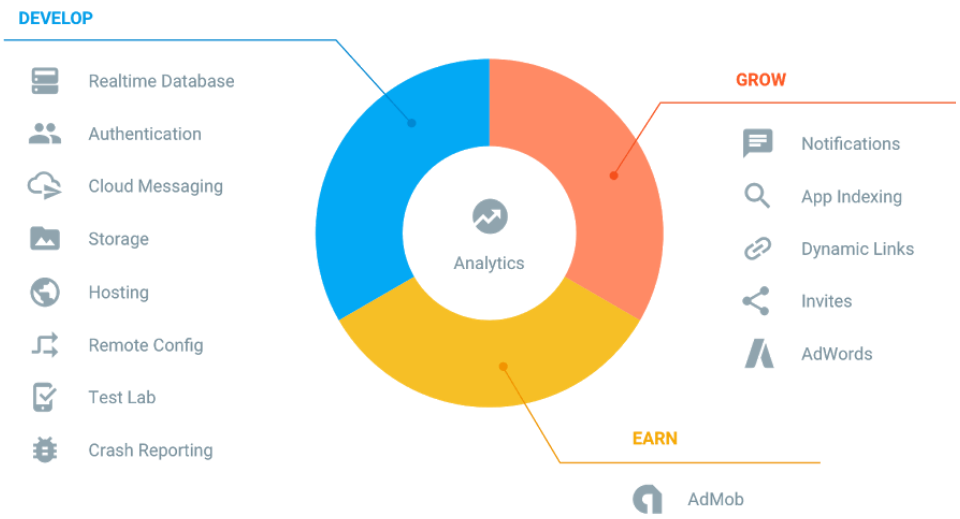
privát, nyilvános vagy megosztott adatbázisokban. A keretrendszer használatához azonban csatlakozni kell az iOS Developer Programhoz.

Mivel a CloudKit kizárólag az Apple ökoszisztémájához készült, és indokolatlanul nehéz más platformokon használni, meglehetősen korlátozott BaaS-eszköz. Mindeközben az iOS-fejlesztők nagy valószínűséggel értékelni fogják a platform nulla licencdíját, a pontszerű funkciókészletet és a hálózati stabilitást. [10]

2.3 Miért a Firebase a legjobb választás és mi az a Firebase?

Firebase első sajtókonferenciáján 2012 áprilisában James Tamplin arról beszélt, hogy ideje a hagyományos szerver architektúrákat leváltani és egy sokkal innovatívabb és főleg költséghatékonyabb irányba elindulni. Ebben az időszakban több startup is próbálkozást tett a BaaS rendszerek piacán, kisebb nagyobb sikerrel. Ekkor tájt indult útnak a már említett Parse is, amit lassan már 4 éve nem is üzemeltetnek.

Ahogyan a fentebb felsorolt termékek sem egy konkrét termékek, így a Firebase sem. Sokkal inkább egy szolgáltatás csomag, sok hasznos eszköz egy nagy integrált egésszé gyúrva. [12]

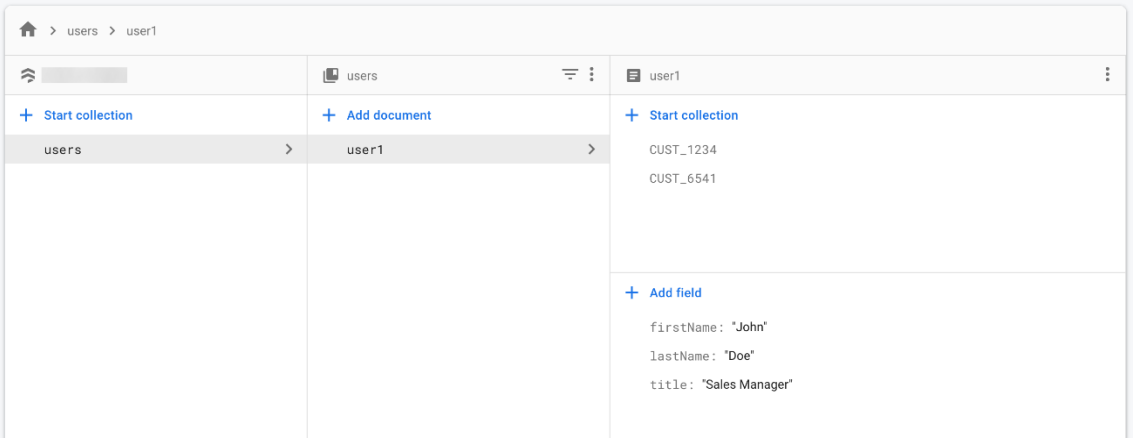


7. ábra - Firebase ökoszisztéma

2.3.1 Cloud Firestore

A Cloud Firestore egy felhőben host-olt, NoSQL adatbázis, amelyet az iOS, Android és webes alkalmazásai SDK-kon keresztül közvetlenül elérhetnek. Egyszerűsíti az adatok szinkronizálását, tárolását és lekérdezését mobil, webes, valamint IoT alkalmazások számára. A Google Cloud előnyeivel a Firestore nagyszerű skálázhatóságot kínál, mindemellett élő szinkronizálást és offline támogatást is biztosít. [13]

A Firestore egy dokumentumtároló adatbázis, leginkább a MongoDB-hez hasonlatos. Három különböző dolgot érdemes megkülönböztetni a Firestore-ban, vannak a gyűjtemények (collections) és annak adattulajdonság mezői és a dokumentumok. A dokumentumok az algyűjtemények mellett összetett és egymásba ágyazott objektumokat is tartalmazhatnak.



8. ábra - Firestore dokumentum tárolás

A valós idejű adatbázishoz hasonlóan a Cloud Firestore is adatszinkronizációt használ az adatok frissítéséhez, bármely csatlakoztatott eszközön. Ugyanakkor, úgy tervezték, hogy egyszerű, egyszeri lekérdezéseket is hatékonyan végezzen. Másik nagyon hasznos funkciója pedig az offline támogatás.

A Firestore gyorsítótárba teszi azokat az adatokat, amelyeket az alkalmazás aktívan használ, így az alkalmazás akkor is tud adatokat írni, olvasni, ha az eszköz offline állapotban van. Amikor az eszköz újra csatlakozik egy hálózathoz, a Firestore felszinkronizálja a helyi változásokat.

Ez a szolgáltatás remekül skálázható a kis, pár gyűjteményes adatbázistól a legnagyobb és legbonyolultabb alkalmazásig.

A Firestore a Google Cloud nagy teljesítményű infrastruktúráját kihasználja, ezért biztosított az automatikus több régióra kiterjedő adatreplikáció, és az erős adatkonzisztencia. [13]

2.3.2 Cloud Functions

A Cloud Function-kel tudunk reagálni az adatbázisban történt változásokra. A Google szerverei figyelik az adatbázis változásokat és ha egy adott eseményre fel van iratkoztatva a függvény akkor a Firebase lefuttatja. Az ilyen függvényekkel lehet Push értesítési folyamatokat elindítani.

A terhelés növekedésével vagy csökkenésével a Google a funkció futtatásához szükséges virtuális kiszolgálópéldányok számának gyors skálázásával reagál. Minden egyes funkció elszigetelten, saját környezetben, saját konfigurációval fut.

Egy ilyen függvény életciklusa a következő:

1. A Firebase-be feltöltésre kerül egy .zip-archívum, ami függvény forrásállományait.
2. A Cloud Build lekérdezi a függvénykódot és elkészíti a függvény forrását.
3. Az elkészült forrásállomány konténerképe feltöltődik a projektben lévő privát Container Registry tárolóba (aminek a neve gcf).
4. Amikor az eseményszolgáltató olyan eseményt generál, amely megfelel a függvény feltételeinek, a kód meghívásra kerül.
5. Ha a függvény túl sok esemény kezelésére lett létrehozva, a Google több példányt készít belőle a gyorsabb munkavégzés érdekében. Ha a függvényt éppen nem használják van, akkor a nem használt példányok eltakarításra kerülnek.

Amennyiben egy funkció törlésre kerül, az összes példányt és zip-archívumot, valamint a Cloud Storage-t és a Container Registry-hez kapcsolódó elemeket is törli. Amellett, hogy listenerek-kel figyeljük az eseményeket, a függvényeket közvetlenül http requestekkel is meg lehet hívni.

Sok esetben érdemesebb a kiszolgálón irányítani az alkalmazáslogikát, hogy elkerülhető legyen a kliensoldalon történő adatmanipuláció. Emellett, az sem elhanyagolható érv, hogy a kódot így nem lehet visszafejteni. A Cloud Functions teljesen el van szigetelve az ügyféltől, így biztosak lehetünk benne, hogy biztonságos. [14]

2.3.3 Analytics

A Google Analytics-el könnyedén és átláthatóan lehet monitorozni, hogy hogyan használják a felhasználók az IOS vagy Android alkalmazásokat. Az SDK automatikusan rögzít számos eseményt és felhasználói interakciót, valamint lehetővé teszi, hogy saját egyéni eseményeket határozzunk meg. Miután az adatok rögzítésre kerültek, a Firebase konzolon keresztül egy műszerfalon lesznek elérhetők. Ez a műszerfal részletes betekintést nyújt az adatokról, mint például az aktív felhasználók és a demográfiai adatok.

Amennyiben egyéni elemzést kell végezni, vagy adatait más forrásokkal kell összekapcsolni, lehetőség van összekötni az Analytics-adatait a Big Query-vel, amely lehetővé teszi az összetettebb elemzést, például a nagy adathalmazokon végzett lekérdezéseket és analitikákat.

Az Analytics naplózza az eseményeket minden egyes összeomlásnál, így egy átfogó képet kaphatunk a különböző verziók vagy régiók összeomlásainak arányáról, és betekintést nyerhetünk abba, hogy mely felhasználókra van hatással. Célcsoportokat is létrehozhatunk olyan felhasználók számára, akik többször is összeomlást tapasztaltak, és

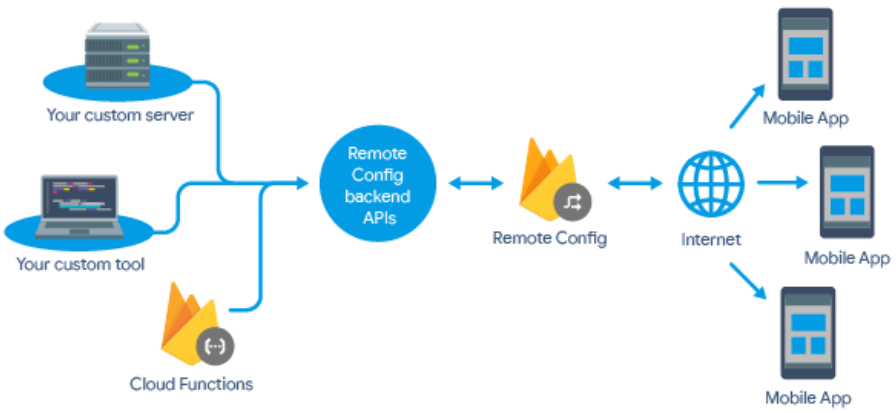
az adott célközönségnek címzett értesítési üzeneteket küldhetünk. Az Analytics automatikusan naplózza az értesítéseket is és támogatja az egyes kampányokról szóló jelentés készítéseket. [15]

2.3.4 Firebase Crashlytics

Az iOS, Android és a Unity számára készült, nagy teljesítményű összeomlás-jelentési megoldás. Segít nyomon követni, rangsorolni és javítani az alkalmazás minőségét rontó stabilitási problémákat. A Crashlytics intelligensen csoportosítja az összeomlásokat, és kiemeli a hozzájuk vezető körülményeket. [16]

2.3.5 Remote Config

A Remote Config-al széleskörűen van lehetőség problémák megoldására. Legyen szó egy A/B tesztelésről vagy távoli nehézség állításról. Egy példán keresztül bemutatva, hogy mire is képes a Remote Config. Adott egy játék, amely kikerül a két áruházba. A felhasználók elkezdenek vele játszani és használni az alkalmazást, kiderül időközben, hogy az egyik szint túl nehézre sikerült. Ahelyett, hogy újra buildelnénk, tesztelnénk kitelepítenénk és várnánk, hogy a felhasználók frissítsék az alkalmazást már a könnyebb pályára, ezt a Remote Config-al könnyedén meg lehet oldani. Lényegében a kódnak a konfigurációs részét kiszervezzük a kódból a felhőbe és ott tudjuk módosítani például a pálya nehézségét. Minden alkalmazás indulásnál, lekérdezésre kerülnek a felhőből a konfigurációk és így midig napra kész lehet a játékunk.



9. ábra - Remote config

Ezt remekül össze lehet kötni az Analytics-el mért adatok alapján, illetve lehet változtatni az alkalmazáson akár csak egy szűk célcsoportot kiválasztva. [12]

3 Vállalati információs rendszer

Az elnevezés tartalma sokat változott a fogalom megjelenése óta, ráadásul az is változó, hogy ma mit értenek alatta. A vállalati erőforrás-tervező, azaz az ERP (Enterprise Resource Planning) rendszerek történetileg a termeléstervezésre fókuszáló Material Requirements Planning rendszerekből fejlődtek ki.

Az ERP rendszereket a következőképpen definiálhatjuk:

"Az EPR olyan komplex szoftvermegoldás, amely egységes rendszerbe integrálja a vállalat összes üzleti funkcióját." [17]

A definíció miatt magyarul inkább integrált vállalatirányítási rendszereknek szokták hívni. Az SAP R/2-es rendszerétől számíthatjuk az ERP megjelenését. Amelynek egy nagyon népszerű verziója a R/3-as programcsomag még ma is sok nagyvállalatnál többé kevésbé jól működik. Mindenesetre ez a rendszer mintaként szolgált a 90-es évek integrált vállalatirányítási rendszerek számára.

Ilyenek voltak például:

- J. D. Edwards Enterprise One,
- Oracle Applications,
- Volán Elektronika Libra.

Ezen kívül manapság több száz ERP található a piacon, amik már modernebb technológiai alapokon készültek. [17]

3.1 Milyen általános jellemző vannak egy Vállalati irányítási rendszernek?

- Olyan nagy egybefüggő szoftvercsomag, ami képes a vállalatot minden üzleti aspektusból segíteni.
- Konstrukciót illetően létezik belőle kész szoftver, amely viszonylag nagy mértékben testre szabható. Léteznek teljesen egyedi fejlesztések, de ezek a szoftverek elhanyagolhatóak mivel egy vállalatnak standardizált folyamatainak kell lennie,
- Mivel egy szoftvercsomagról beszélünk, így egy jól működő ERP rendszernek moduláris felépítésűnek kell lennie. Egyes moduloknak az integritása megfelelő és könnyedén egyesével is bevezethetők a vállalat életébe.
- További konstrukciót illetően lehet egy ERP rendszer On Premise (kitelepített megrendelőnél egy szerveren futó) vagy hostolt tehát maga az ERP szállító a saját szerverein üzemelteti.

- Központi adatbázisra épül. Minden alkalmazási modul a központi adatbázist használja, amely általában relációs típusú.
- Mindegyik modul egy és ugyan azon adatbázist használ a működése során.

Egyes rendszerek között is van különbség, de kimondható, hogy az alábbi modulokat biztosan tartalmaznia kell:

- Pénzügy.
- Könyvelés.
- Termelésirányítás.
- Beszerzés.
- Készletgazdálkodás.
- Értékesítés.
- Marketing.
- HR.

A gyártók ezen kívül még nagyon sok különböző modult kínálnak például projektmenedzsment vagy minőségmenedzsment. Fent felsorolt modulok/alrendszerek ellátják az üzletmenet funkcióit, továbbá jelentések és lekérdezések formájában információval szolgálnak az operatív szintű menedzsment számára. Ez tulajdonképpen nehezen vetíthető le egy KKV életére, mivel ott nem különülnek el ennyire a modulok és kevesebb az alkalmazott is. Egy ERP rendszer bevezetése igencsak időigényes folyamat, egy nagyvállalat esetében akár több évig is eltarthat. [17]

3.2 Kiterjesztett ERP rendszerek

Ezek a rendszerek tulajdonképpen ugyan olyan ERP-k kiegészített modulokkal, mint például a CRM (ügyfélkapcsolat) és SRM (szállítói kapcsolat). Ezek a rendszerek további hasznos, a mindennapokban használt funkcionalitással ruházzák fel a már meglévő ERP rendszert.

3.2.1.1 Ügyfélkapcsolat menedzsment (CRM)

Az ügyfélkapcsolat menedzsment rendszerek nem csak a vevőkkel való kommunikációt segítik, hanem a vevők számára az értékesítési folyamatokat is, és az esetleges panaszkezelési ügyfélszolgálatot is meglehet valósítani a segítségével. Ezzel is növelve a szolgáltatás színvonalát és a vállalat hatékonyságát. Szegmentálni lehet benne a különböző értékesítési csatornákat és célcsoportokokat, mint a B2B (Business To Business), B2C (Business To Customers) vagy akár a B2G (Business To Goverment) is.

Megkülönböztetünk háromfajta CRM megoldást, az operatívat, az analitikust és a kollaboratívot. Az operatív CRM rendszerek feladatai közé tartoznak a következők:

- Marketingkampányok lebonyolítása, folyamatok kialakítása
- Az értékesítési folyamat támogatása.

- Automatizált ügyfélkapcsolati folyamatok végrehajtása.
- Az ügyfélszolgálat támogatása.

A CRM rendszerek működését egy egyszerűbb példán keresztül remekül lehet szemléltetni. Egy általános webáruház esetén, ha vásároltunk valamit és megadtuk az adatainkat (Név, e-mail cím, cím, telefonszám stb.), bekerültünk a webshopnak az értékesítési rendszerébe, adatbázisába. Amennyiben egy szezonális terméket vásároltunk például egy téligumit, hamarosan reklámokat hírleveleket és kapcsolódó termékekről szóló hirdetéseket fogunk kapni. 2-3 év elteltével, amikor már a téligumit cserélni kell, a marketing osztály vélhetően felkeres egy kedvező ajánlattal.

Egy analitikus CRM rendszer modulban pedig a fenti példa értékesítési adatait, jósági tényezőit lehet diagramokon és különféle riportokon nyomon követni. Ezzel is elősegítve, hogy minél jobb marketing stratégiát lehessen kidolgozni. [17]

3.2.1.2 Beszállítói kapcsolat-kezelő rendszerek (SRM)

A beszállítói kapcsolat-kezelő rendszerek a vállalat beszállítóihoz kapcsolódó munkafolyamatokat tartják nyilván és naprakészen. Honnan, mikor és milyen áron milyen szállítási kondíciókkal rendelkeznek egyes beszállítok. Ez nem csak fizikai termékekre, fogyóeszközökre vonatkozik, hanem például egy szoftverfejlesztő cég által szállított szoftver.

Az SRM által támogatott folyamatok a következők:

- Beszállítók kiválasztása
  - Lehetséges beszállítók megtalálása, értékelése,
- Beszerzés
  - Ajánlatkérések elkészítése, elküldése,
  - Ajánlatok kiértékelése, beszerzés jóváhagyása,
  - Megrendelések elkészítése, elküldése.
- A beérkező áru fogadása
  - Áru kezelése, raktározása,
  - Számlák kezelése.

SRM bevezetése a vállalat hatékonyságát nagymértékben növeli, a beszállítókkal való kapcsolattartást, hibák kiszűrését és a költségek alacsonyan tartását támogatja. [17]

3.3 Magyarországon népszerű vállalatirányítási rendszerek:

3.3.1.1 SAP

Az SAP a világon a legelterjedtebb vállalatirányítási rendszerével, több mint 30 éves tapasztalattal és piaci részesedéssel rendelkezik. Nincs olyan vállalatirányítási modul vagy funkció, ami ne lenne megtalálható benne. Az üzleti folyamatait az SAP-nak több

éven át tartó konzultációk és rengeteg tapasztalat alapján fektették le és így vált a mai vállalat irányítási folyamatok sztenderdjévé. Ami az előnye az néha a hátránya is, mivel hatalmas és komplex rendszerről beszélünk, így annak bevezetése karbantartása egy tanácsadó cég vagy komoly háttértudás nélkül elég nehéz feladat. Az árát illetően Magyarországon főleg a nagy multik engedhetik meg maguknak. Az SAP igyekezetet mutat a legkülönfélébb vállalati méreteket is lefedni, így a kis és középvállalkozásoknak szánt SAP Business One egy jó választás lehet itthon is, amennyiben rendelkezésre áll a megfelelő tőke.

3.3.1.2 SAP Business One

A fentebb említett SAP Business One megoldás főleg a kisés közép vállalkozások számára lehet megfelelő választás. Több mint 170 millió felhőfelhasználóval, és több mint 45 éves tapasztalattal rendelkeznek. A Forbes Global 2000 társaságok több mint 90% -a SAP ügyfél. Szolgáltatásaikat nem csak a globális óriások használják. Az SAP ügyfélkörének több mint 80% -át kis- és középvállalkozások alkotják. Ezen túlmenően, az iparág szerinti termékszegmentálás miatt az SAP mindenki számára univerzális megoldásnak tekinthető. Kontrasztban például az Oracle megoldásával az Oracle Cloud, amely csak a nagyvállalati szféra problémáira nyújt optimális megoldást. [18]

3.3.1.3 Microsoft Dynamics NAV (Navision, Business Central)

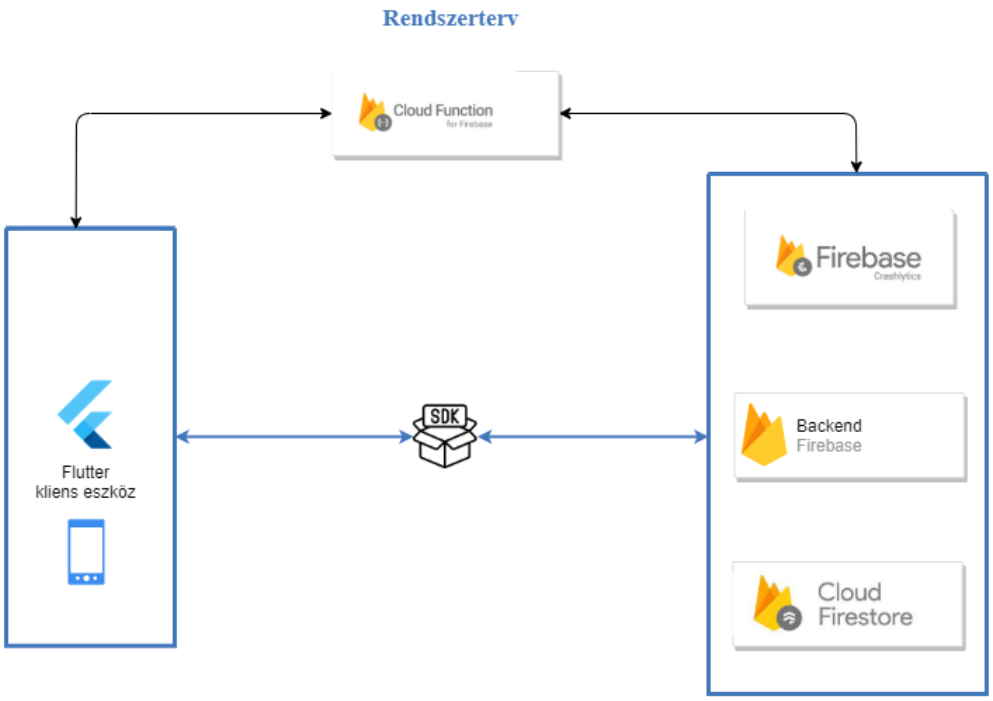
A Navision is egy kis-és középvállalatok számára készült integrált vállalatirányítási megoldás. A fejlesztője pedig a Microsoft, ami egy igen nagymúltú cég, így a Navision is hasonlóan letisztázott, kiforrott üzleti folyamatokon alapszik. Költséghatékonyabb megoldás lehet az SAP megoldásánál és egyszerűbb a kezelőfelülete is. Itt szintén az előny az a hátrány, funkcionalitásban elmarad az előbbiekben részletezett megoldásoktól.

3.3.1.4 Abas

Az Abas ERP rendszer egy átlátható remek dashboardokkal rendelkező szoftver, amely szintén tartalmazza az alapvető és nélkülözhetetlen modulokat. Az Abas a kezdetektől fogva a gyártóiparra specializálódott, ezért a gyártási folyamatok tervezése és irányítása megoldásaik kiemelkedő erősségei közé tartozik. Az Abas ERP aktívan támogatja beszerzési osztályt a döntések meghozatalában, amelyek jobb feltételekhez, és a beszállítóválasztást segítik elő. [19]

4 RENDSZERTERV

Ebben a fejezetben a mobilalkalmazás fejlesztését megelőző tervezési elemeket részletezem. A rendszer magasszintű terve a következőképpen néz ki:



10. ábra - Rendszerterv

4.1 Adatbázis réteg

A következőkben felsorolt gyűjtemények az alkalmazás lényeges és fő funkcióihoz kapcsolódnak, ezeken kívül megtalálhatóak még további rendszerleíró és konfigurációs gyűjtemények is.

4.1.1 Fő gyűjtemények és funkcióik

- **Accomplishments**

  Az egyesület által elért eredményeket hivatott tárolni.

- **Attendance**

  Az edzéseken, órákon való részvétel adminisztrálást teszi lehetővé.

- **Dance Videos**

  Specifikus táncegyüttesi funkció, ahol az órákon készült oktató felvételeket lehet megosztani és az alkalmazásba beágyazni YouTube linkeken keresztül.

- **FAQS**

  Gyakran ismételt kérdések és annak válaszainak a gyűjteménye.

- **Chatrooms**

  Felhasználók közötti kommunikáció tárolására szolgáló gyűjtemény.

- **Documents**

  Egyesület adminisztrációs életében kitöltendő dokumentumok, mint igazolások, programlista stb. Tárolására szolgál.

- **Events**

  A gyűjtemény arra lesz használva, hogy az eseményeket tudjunk benne tárolni. Esemény neve, helyszíne, leírása stb.

- **News**

  Alapvető Push üzenetek működésért felelős gyűjtemény, gyors fontos hírek megosztása majd a főoldalon való listázásáért felelős gyűjtemény.

- **Products**

  A felhasználók egymásközötti „piac” termékeit hivatott tárolni.

- **Users**

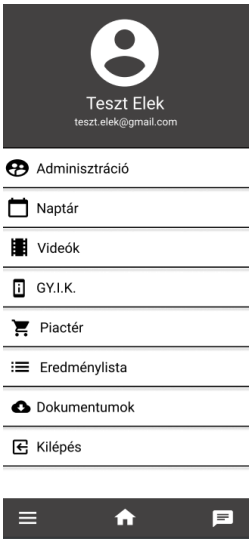
  Maga a felhasználókat tartalmazza majd, és azok alapvető adatait, mint név, profilkép és csoport.

4.2 Kezelőfelület elemei

4.2.1 Menürendszer

A menürendszerből elérhetőnek kell lennie minden az alkalmazás által támogatott funkciónak, a jövőben pedig dinamikusan bővíthető, skálázhatónak kell lennie.

Az alsó ún. „bottom navigation bar”, három különböző oldalra való navigálásért felel. Sorrendben: Menü-Főoldal (Hírek)-Chat lista



11. ábra - Menürendszer terv

4.2.2 Hírek megjelenítő oldal

A hírek megjelenítő oldalának szorosan összehangban kell működnie a Push üzenetekkel, ugyanis, ha egy Push üzenet érkezik a telefonra az a News gyűjteménybe kerül beírásra gyűjtemény tartalma pedig ezen az oldalon kerül megjelenítésre. Ennek kell lennie az alkalmazás bejelentkezés utáni nyitó oldalának. Jobb felső sarokban pedig ki kell alakítani egy kijelentkező gombot.

12. ábra - Hírek megjelenítő oldal terv

4.2.3 Adminisztrációs felület

Az adminisztrációs felület lesz a jelenlegi alkalmazás „lelke”, innen kerül feltöltésre minden az alkalmazásban megtalálható tartalom. Legyen majd lehetőség az alábbi

funkcióban a CRUD (Létrehozás, Olvasás, Frissítés, Törlés) műveleteket elvégezni. Admin funkciók sorrendben: Felhasználókezelés, Hírek kezelése, Jelenléti ív, Videók kezelése, Gyakran ismételt kérdések menedzselése, csoportok kezelése, dokumentumok kezelése, események kezelése. Admin jogosultághoz kötött nézet.

13. ábra - Adminisztrációs felület terv

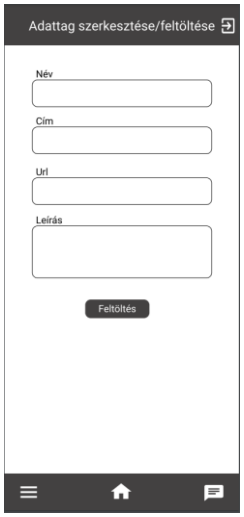
4.2.4 Admin felület dinamikus listázás

Fentebb említett Admin felület funkcióinak a listázó felülete legyen egységes könnyen kezelhető „kártyás megoldás”, a kártyák balra csúsztatásával lehessen az adott adattagot törölni és a névre kattintással pedig szerkeszteni. Admin jogosultághoz kötött nézet.

14. ábra - Admin felület dinamikus listázás terv

4.2.5 Admin felület adattag szerkesztés és új létrehozása

Admin felületen a listából kiválasztás után az adott adattag tulajdonságainak dinamikus megjelenítése és szerkesztése. Ezt a felületet szükséges alkalmazni a létrehozási folyamatok során is. Admin jogosultághoz kötött nézet.



15. ábra - Admin felület adattag szerkesztés és új létrehozása terv

4.2.6 Eseménynaptár

Esemény naptár felületet minden az alkalmazásban jóváhagyott felhasználó legyen képes megtekinteni. A felületein a napokat havi bontásban szükséges megjeleníteni és egy adott nap kiválasztása után a napra vonatkozó eseményeket a naptár alatt listázni. Amennyiben nincs kiválasztott nap, akkor időrendi sorendben az összes esemény látszódjon.

16. ábra - Eseménynaptár terv

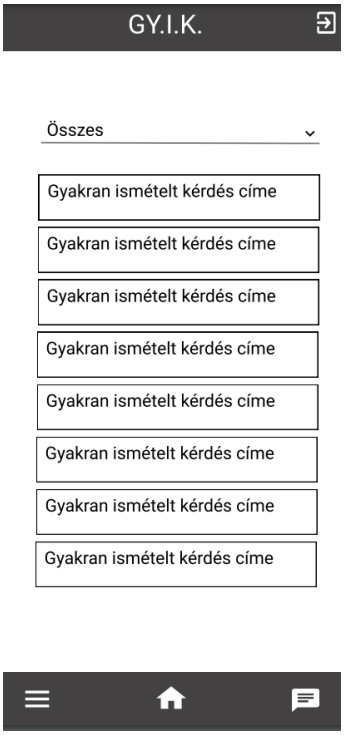
4.2.7 Videók megjelenítés oldala

Az Admin felületen feltöltött videók reszponzív helytakarékos megjelenítő oldala. Mivel az alkalmazás nem tárolja a konkrét videót csak a videó megosztóra feltöltött linkjét. A funkció legyen képes a videó thumbnail képének automatikus lementésére, hogy képet se legyen szükséges feltölteni egy videóhoz. A videóra való kattintáskor nyissa meg a telefonon a legoptimálisabb lejátszó alkalmazást (YouTube alkalmazás, Böngésző).

17. ábra - Videók megjelenítés oldal terv

4.2.8 Gyakran ismételt kérdések felülete

Gyakran ismételt kérdések felületen az adminisztrációs oldalon menedzselt kérdések és annak válaszai legyenek megtalálhatóak. Nagyon fontos mivel sok kérdés lehet idővel, egy szűrésre szolgáló legördülő menü beépítése, ahol kategóriákra lehet majd szűkíteni a listát.



18. ábra - Gyakran ismételt kérdések felület terv

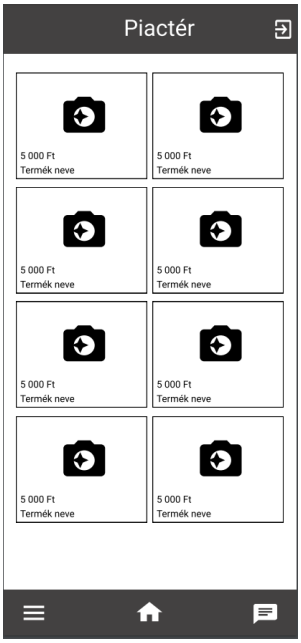
4.2.9 Profil oldal

A profil oldal egy dinamikusan növekvő, saját profil adatállományt hivatott megjeleníteni. Egyelőre az alaptulajdonságok legyenek megjelenítve az oldalon és legyen lehetőség profilképet cserélni.

19. ábra - Profil oldal terv

4.2.10 Piactér

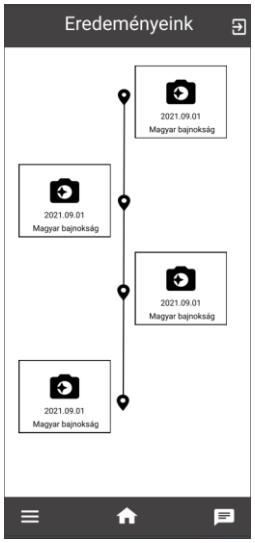
A piactéren az alkalmazásban feladott apróhirdetéseknek a reszponzív megjelenítését kell megvalósítani. Legyen lehetőség közvetlenül üzenetet küldeni majd az eladónak, továbbá minden felhasználó a saját termékét tudja szerkeszteni, állapotát módosítani.



20. ábra - Piactér terv

4.2.11 Eredményeink

Az eredményeink egy úgy nevezet „dicsőségfal”, ahol az egyesületben sportoló emberek eredményei lennének kronológiai sorrendben megjelenítve. Egy eseményre kattintva legyen lehetőség megnézni, hogy ki és milyen eredményt és hol ért el.



21. ábra - Eredményeink terv

4.2.12 Dokumentumok

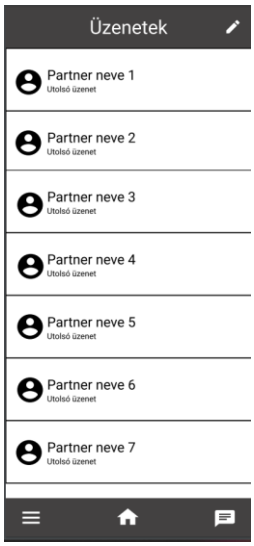
Az egyesület hivatalos közleményei, igazolás sablonjai és egyéb dokumentumokat lehessen egyelőre csak listás nézetben megjeleníteni, későbbiekben egy jogosultsági rendszerhez kell kötni, hogy ki milyen dokumentumhoz fér hozzá.

22. ábra - Dokumentumok terv

4.2.13 Chat üzeneteket listázó oldal

Az alkalmazáson belüli kommunikációért felelős megjelenítő oldal. A felületen a nagy social média platformok alapvető UX/UI megközelítéseit szükséges alapul venni. Tehát az utolsó üzenetváltás kerüljön az oldal tetejére. Jobb felső sarokban pedig egy új üzenet

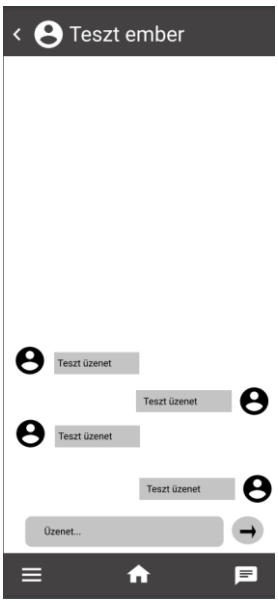
írása oldalt lehessen megjeleníteni, ahol egy dinamikus keresővel lehessen megtalálni a partnert.



23. ábra - Chat üzenetek terv

4.2.14 Chat szoba oldal

Miután kiválasztásra került a beszélgető partner az alábbi felületnek kell elkészülni. Mint minden chat alkalmazásban a képernyő baloldalán a kapott üzenetek, a jobb oldalán pedig az elküldött üzenetek. Az egy az egyhez chat kialakításánál figyelembe kell venni a jövőbeli csoportos üzenetek funkciót és e szerint szükséges implementálni.



24. ábra - Chat szoba terv

5 MEGVALÓSÍTÁS

Az alábbi fejezet az alkalmazás fejlesztését követően előállt eredmény részletezéséről szól.

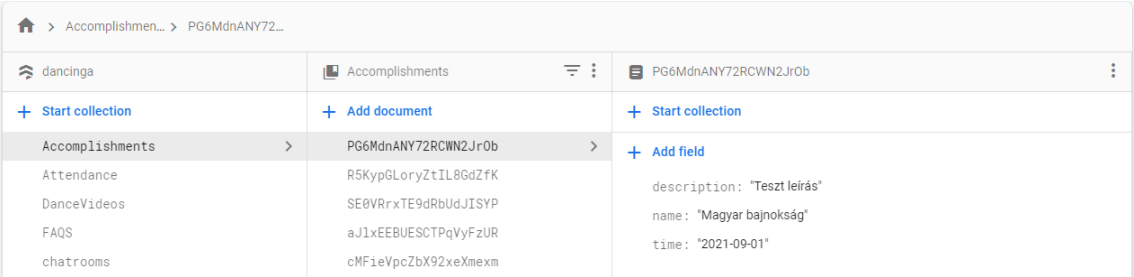
5.1 Adatbázis réteg

5.1.1 Accomplishments

Az egyesület által elért eredményeket hivatott tárolni.

Mezői:

- DocumentID: Firebase által automatikusan generált azonosító.
- description: Rövid leírás.
- time: Időpont, hogy mikor került az adott esemény megrendezésre.

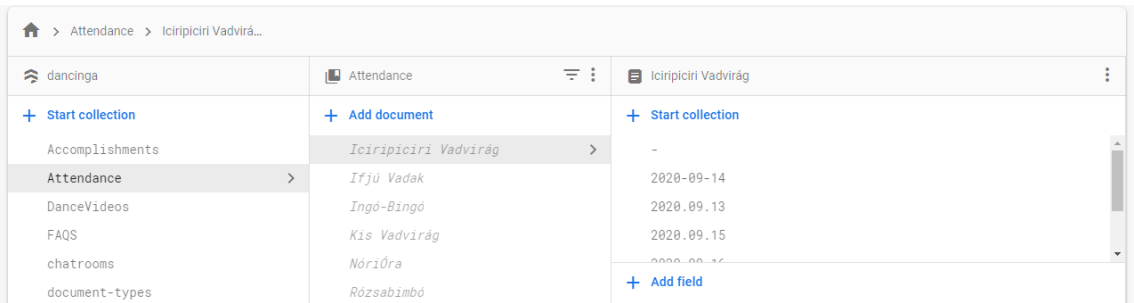


25. ábra - Eredmények gyűjtemény

5.1.2 Attendance

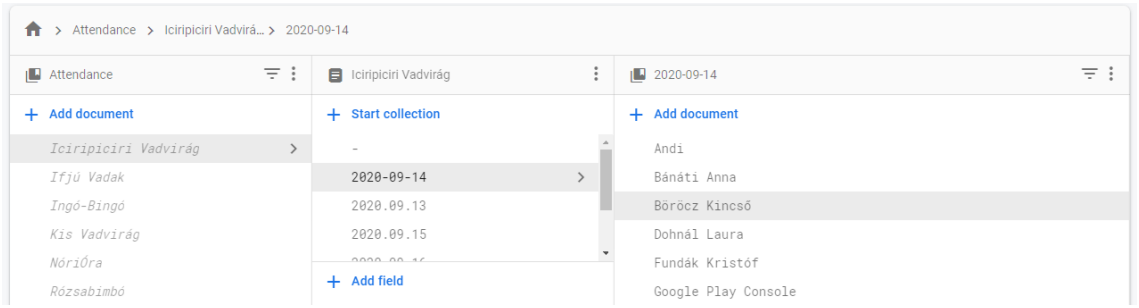
Az edzéseken, órákon való részvétel adminisztrálást teszi lehetővé.

Az Attendance gyűjteményben létrejövő dokumentumok az alkalmazásban előre definiált értékek, amiket a groups gyűjteményből kerülnek szinkronizálásra.



26. ábra - Jelenléti ív gyűjtemény

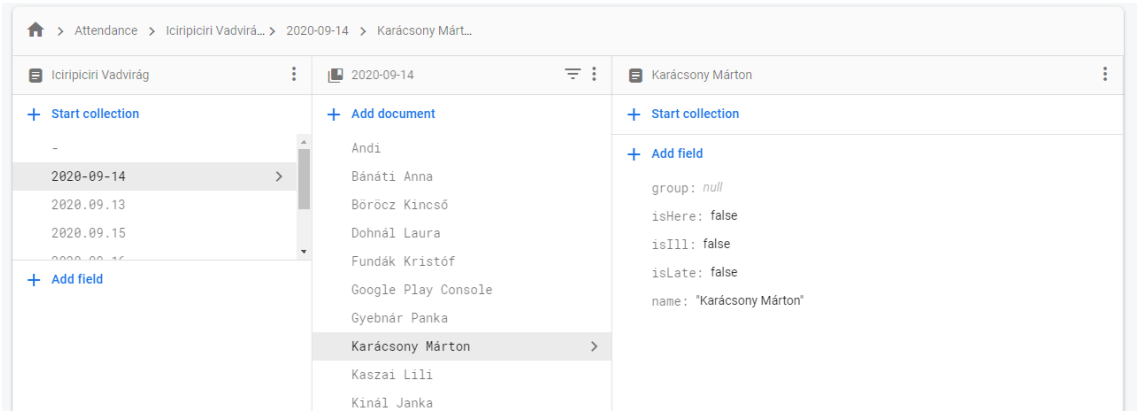
Minden egyes csoport tartalmaz egy algyűjteményt, amelynek az azonosítója az a nap amikor az edzés volt.



27. ábra - Jelenléti ív csoport bontás gyűjtemény

Az algyűjtemény tartalmazza a felhasználókat, akik az adott csoporthoz tartoznak. Egy adott személynek 3 különböző lényeges tulajdonsága van egy jelenléti ívben.

- isHere: Megjelent-e az adott edzésen.
- isIll: Amennyiben nem volt ott, beteg volt-e.
- isLate: Amennyiben megjelent késett-e.



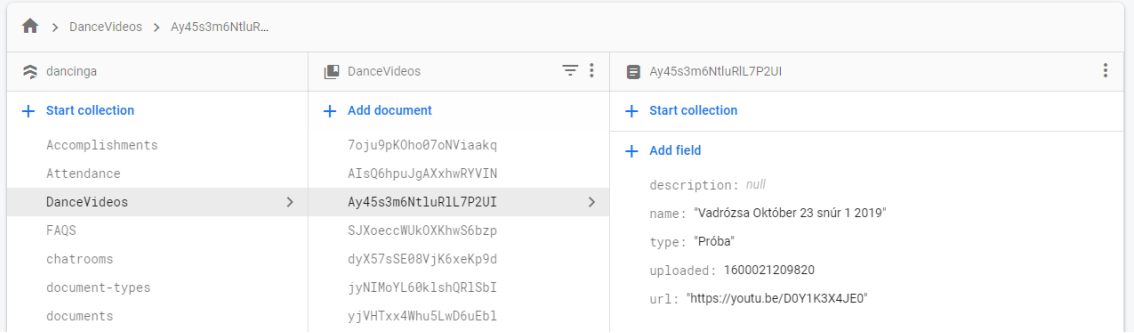
28. ábra - Jelenléti ív személy bontás

5.1.3 DanceVideos

Specifikus táncegyüttesi funkció, ahol az órákon készült oktató felvételeket lehet megosztani és az alkalmazásba beágyazni YouTube linkeken keresztül.

Mezői:

- DocumentID: Firebase által automatikusan generált azonosító.
- name: Videó címe.
- type: Melyik típusú videó kategóriába sorolható.
- uploaded: Feltöltés dátumáról egy timestamp (időbélyeg) integer érték.
- url: A beágyazott videó url-je.



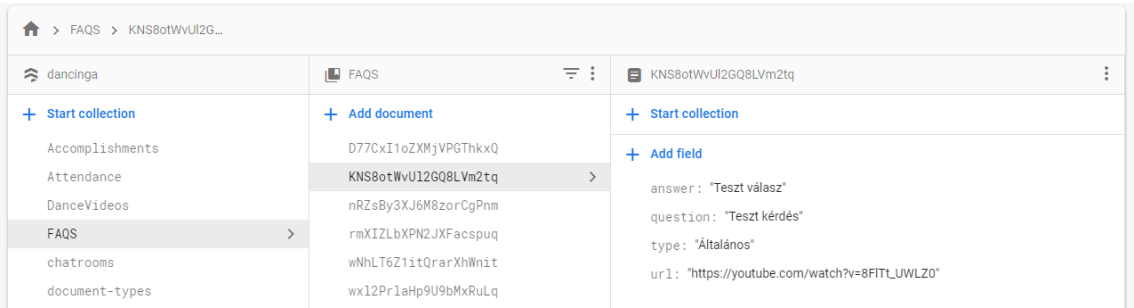
29. ábra - Videók gyűjtemény

5.1.4 FAQS

Gyakran ismételt kérdések és annak válaszainak a gyűjteménye, lehetőség van youtube videó linket beágyazni egy kérdés megválaszolásához.

Mezői:

- DocumentID: Firebase által automatikusan generált azonosító.
- question: Maga a gyakran ismételt kérdés.
- answer: A gyakran ismételt kérdésre adott válasz.
- type: Gyakran ismételt kérdések kategória típusa.
- url: A beágyazott videó url-je.



30. ábra - Gyakran ismételt kérdések gyűjtemény

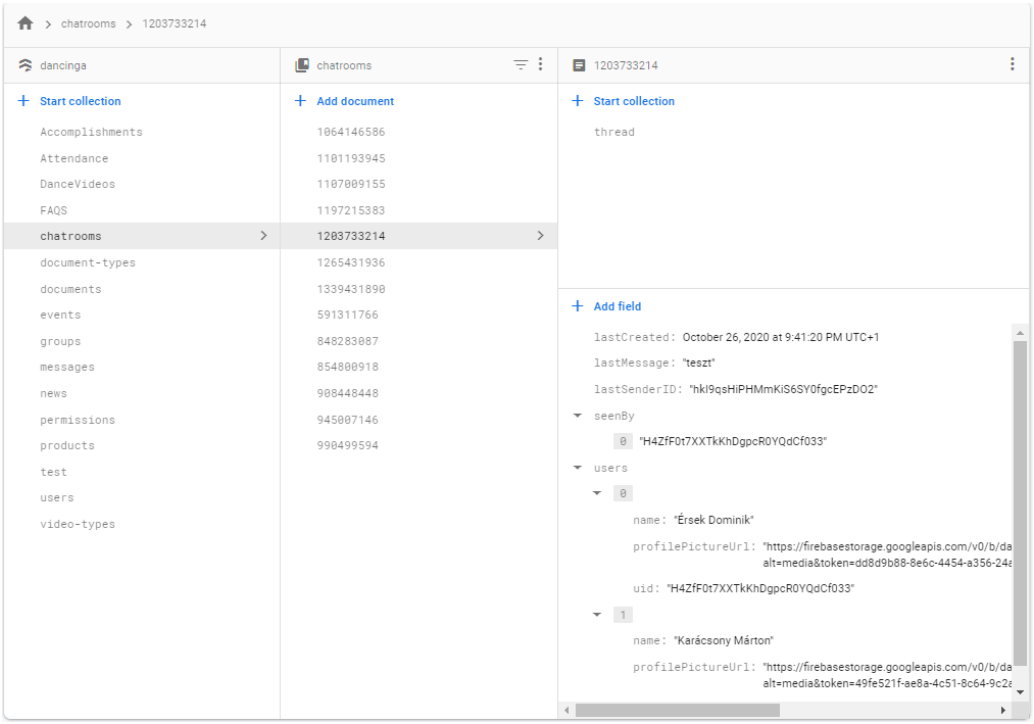
5.1.5 Chatrooms

Felhasználók közötti kommunikáció tárolására szolgáló gyűjtemény.

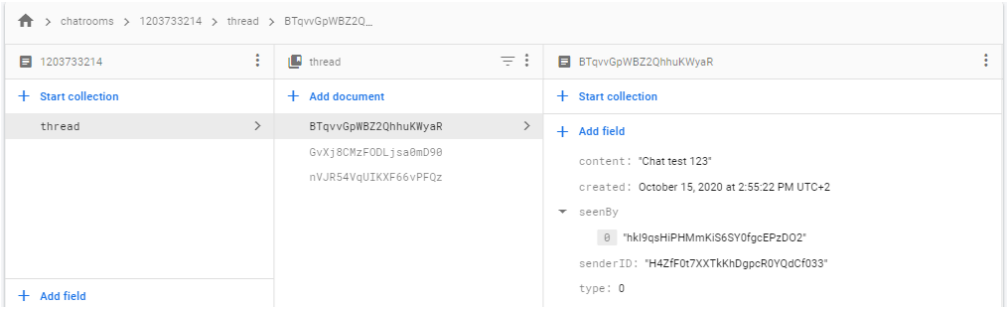
Mezői:

Gyűjtemény:

- lastCreated: Utolsó üzenet mikor lett elküldve.
- lastMessage: Mi volt az utolsó üzenet.
- lastSenderID: Melyik felhasználó küldte az utolsó üzenetet, (UserID)
- seenBy: Felhasználók listája, akik látták az utolsó üzenetet.
- users: A csevegésben résztvevő felhasználok listája.
    - name: Felhasználó neve.
    - profilePictureUrl: Firestore-ban tárolt profilkép url.
    - uid: Felhasználó egyedi azonosítója.



31. ábra - Üzenetek gyűjtemény



32. ábra - Üzenetek algyűjtemény

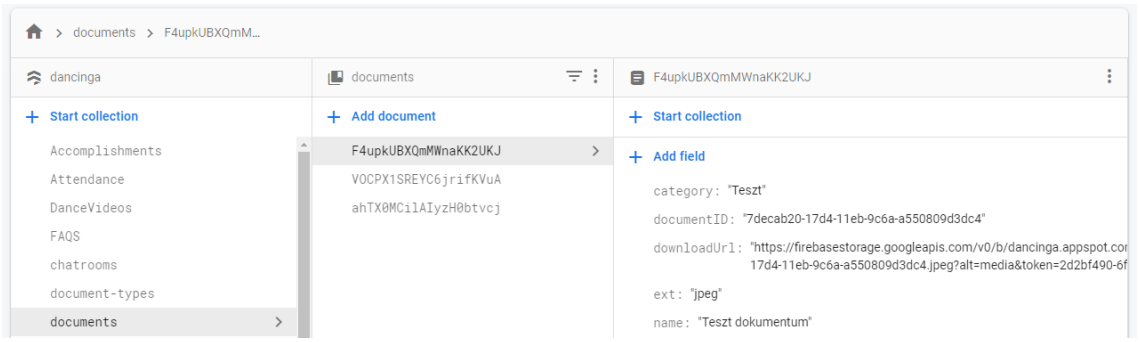
Minden chatroom-hoz tartozik egy thread nevű gyűjtemény, amely tartalmazza az elküldött üzeneteket.

Mezői:

- DocumentID: Firebase által automatikusan generált azonosító.
- Content: Üzenet tartalma.
- created: Üzenet létrehozásának dátuma.
- seenBy: Felhasználói azonosítók listája, akik látták ezt az üzenetet.
- senderID: Felhasználói azonosító, aki küldte az üzenetet.
- type: Az üzenet típusa, ami lehet szabadszöveges tartalom, kép, vagy gif.

5.1.6 Documents

Egyesület adminisztrációs életében kitöltendő dokumentumok, mint igazolások, programlista stb. tárolására szolgál.



33. ábra - Dokumentumok gyűjtemény

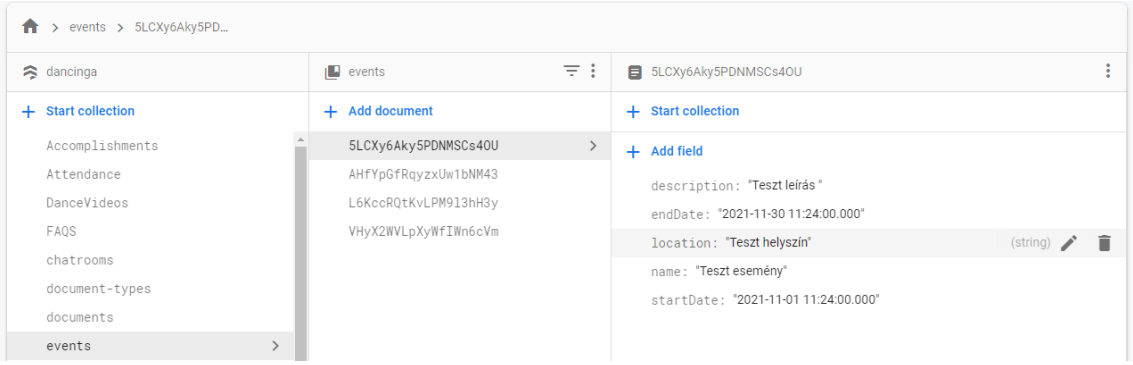
Mezői:

- DocumentID: Firebase által automatikusan generált azonosító.
- category: Feltöltött dokumentum kategóriája.
- downloadUrl: Firebase Firestore-ba feltöltött dokumentum url címe.

- ext: Feltöltött dokumentum kiterjesztése.
- name: Feltöltött dokumentum neve.

5.1.7 Events

A gyűjtemény arra lesz használva, hogy az eseményeket tudjunk benne tárolni. Esemény neve, helyszíne, leírása stb.



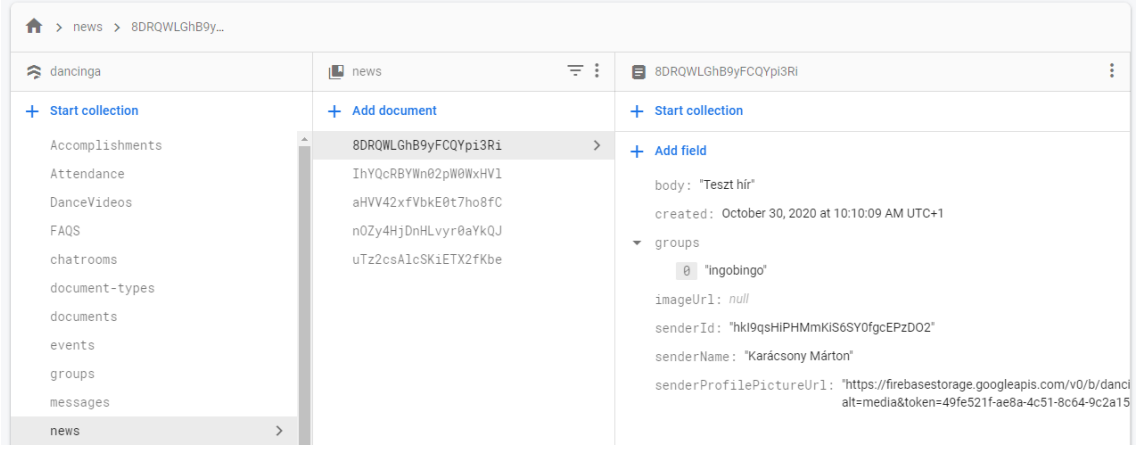
34. ábra - Események gyűjtemény

Mezői:

- DocumentID: Firebase által automatikusan generált azonosító.
- description: Esemény leírása.
- endDate: Esemény végének az időpontja.
- location: Esemény helyszíne.
- name: Esemény neve.
- startDate: Esemény kezdete.

5.1.8 News

Alapvető Push üzenetek működésért felelős gyűjtemény, gyors fontos hírek megosztása majd az főoldalon való listázásáért felelős gyűjtemény.



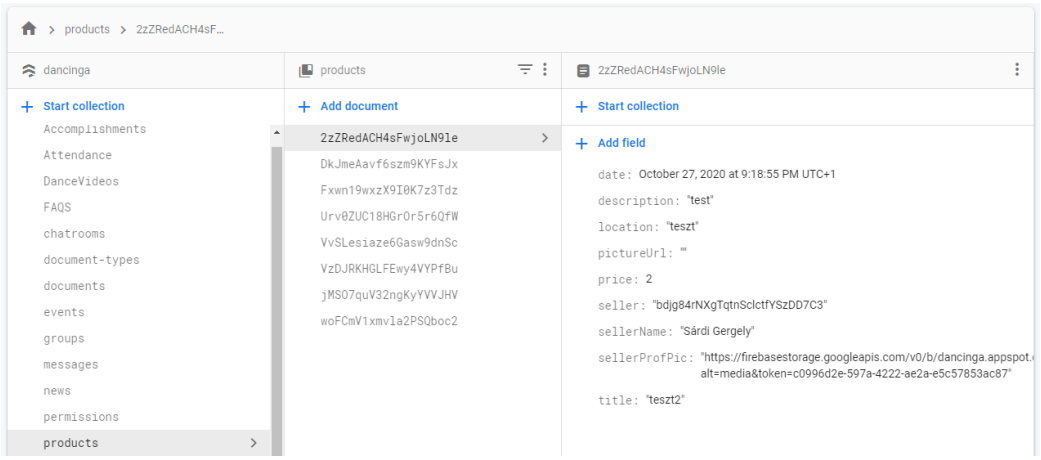
35. ábra - Hírek gyűjtemény

Mezői:

- DocumentID: Firebase által automatikusan generált azonosító.
- body: Hír szövegtörzse.
- created: Hír létrehozásának dátuma.
- groups: Csoport neveket tartalmazó gyűjtemény, akiknek a hír megjelenik a kezdőoldalon.
- imageUrl: Hír képnek a firestore.ba feltöltött url címe.
- senderId: Hír létrehozójának azonosítója.
- senderName: Hír létrehozójának neve.
- senderProfilePictureUrl: Hír létrehozójának profilképe.

5.1.9 Products

A felhasználók egymásközötti „piac” termékeit hivatott tárolni.



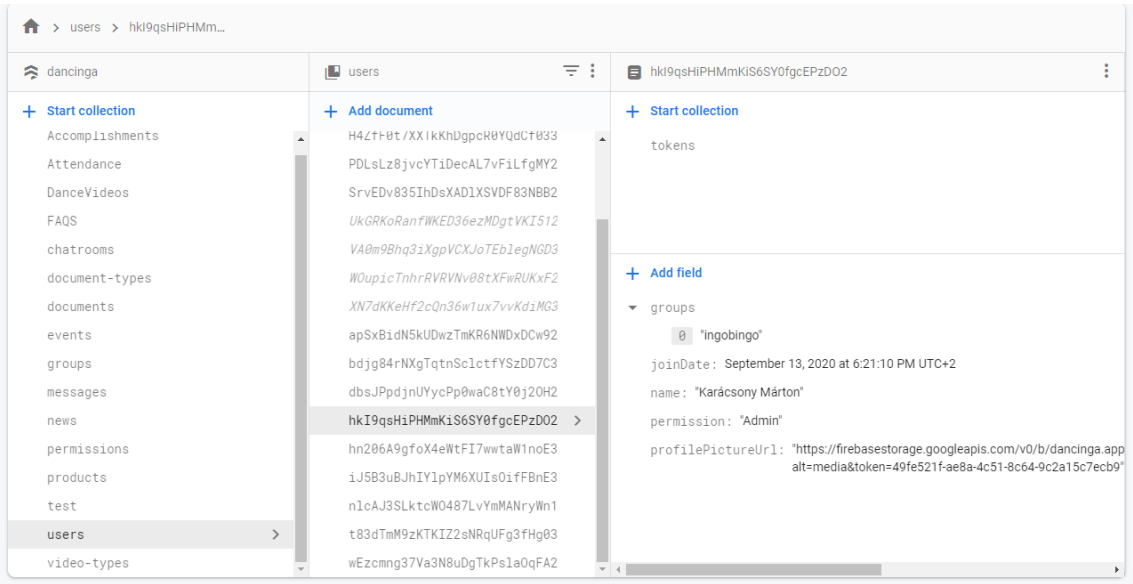
36. ábra - Eladó termékek gyűjteménye

Mezői:

- DocumentID: Firebase által automatikusan generált azonosító.
- date: Hirdetés feladásának dátuma.
- description: Hirdetés leírása.
- location: Termék átvételi helyszíne.
- pictureUrl: Termékről feltöltött kép firestore-os urlje.
- price: Termék ára.
- seller: Eladó azonosítója.
- sellerName: Eladó neve.
- sellerProfPic: Eladó profilképe.
- title: Hírdetés címe.

5.1.10 Users

Maga a felhasználókat tartalmazza majd, és azok alapvető adatait, mint név, profilkép és csoport.

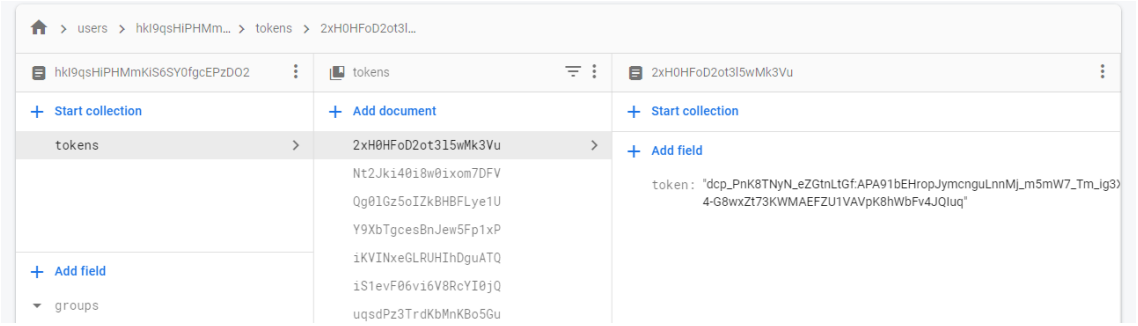


37. ábra - Felhasználók gyűjtemény

Mezői:

- DocumentID: Firebase által automatikusan generált azonosító.
- groups: Felhasználó csoportjainak listája.
- joinDate: Regisztráció dátuma.
- name: Felhasználó neve.
- permission: Felhasználó jogosultsága.

- profilePicture: Felhasználó által feltöltött profilkép firestore-os urlje.



38. ábra- Felhasználók algyűjtemény

Minden felhasználónak létezik egy tokens gyűjteménye, amely tartalmazza a bejelentkezett eszközök egyedi azonosítóját.

Mezői:

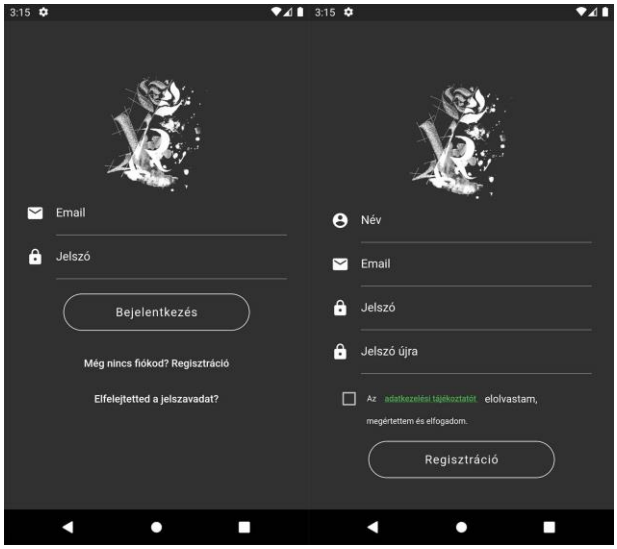
- DocumentID: Firebase által automatikusan generált azonosító.
- token: Egyedi eszköz azonosító.

5.2 Kezelőfelület elemei

Ebben a fejezetben az elkészített alkalmazás képernyőképe/elemei láthatóak.

5.2.1 Authentikáció

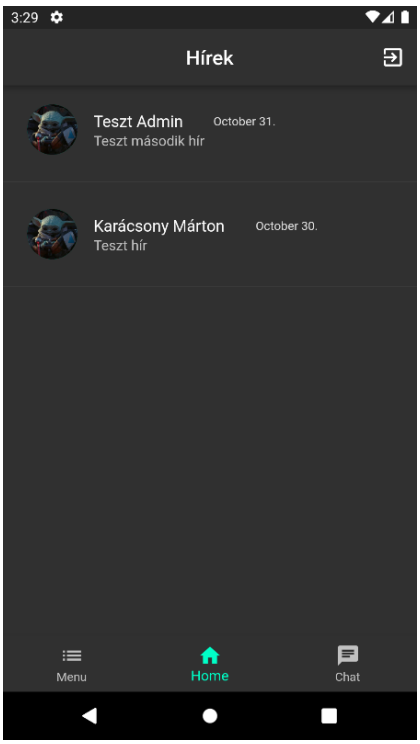
Egyfaktoros authentikációs rendszer letisztult, reszponzív megjelenítése. Alapvető validációk adatokkal az űrlapon, mint az email cím és a jelszó erősség vizsgálat.



39. ábra - Authentikáció megvalósítás

5.2.2 Hírek megjelenítő oldal

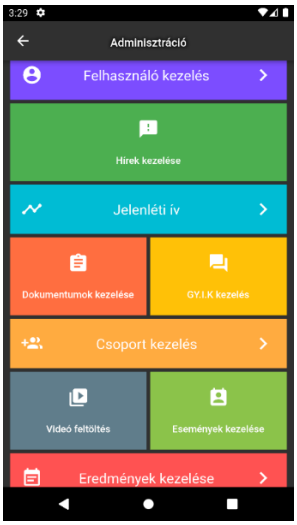
Alkalmazás bejelentkezés utáni kezdőoldala. Dátum szerint csökkenő sorrendben jelennek meg az alkalmazásban közzé tett hírek. Push üzenetre kattintás után erre a screen-re érkezünk.



40. ábra - Hírek megvalósítás

5.2.3 Adminisztrációs felület

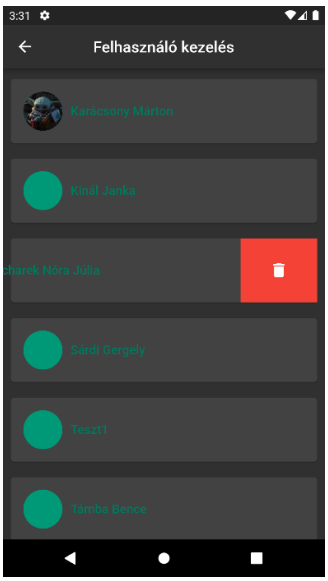
Adminisztrátor funkció kiindulási pontja, minden admin funkciót innen lehet elérni.



41. ábra - Admin felület megvalósítás

5.2.4 Admin felület dinamikus listázás

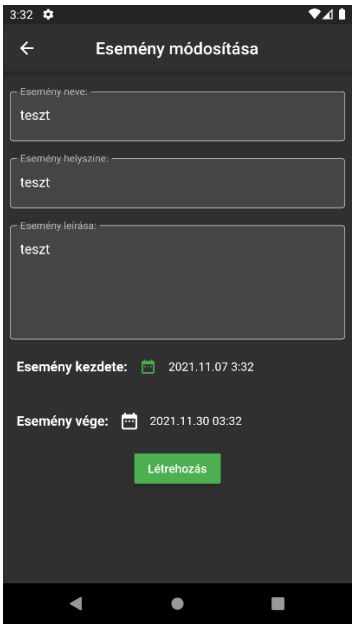
Példa felület az admin felületen lévő CRUD oldalak listázására. Az egyes kártyák balra húzásával lehetőség van törölni az adott listaelemet. Kártyára való nyomással pedig lehetőség van megnyitni a megjelenítő oldalt.



42. ábra - Admin felület dinamikus listázás megvalósítás

5.2.5 Admin felület adattag szerkesztés és új létrehozása

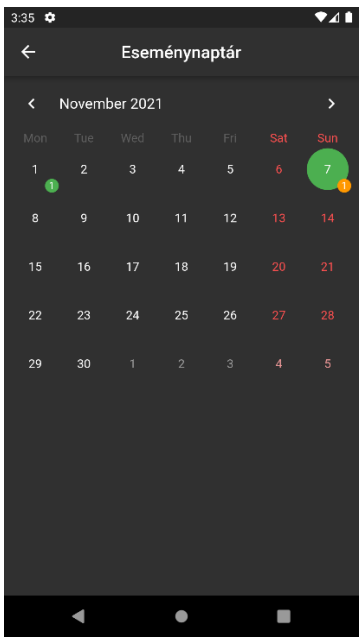
Példa felület egy adattag létrehozására az Admin felületen. Egyszerűbb null validációs metódusok megtalálhatóak az összes beviteli mezőn.



43. ábra - Admin felület adattag szerkesztés megvalósítás

5.2.6 Eseménynaptár

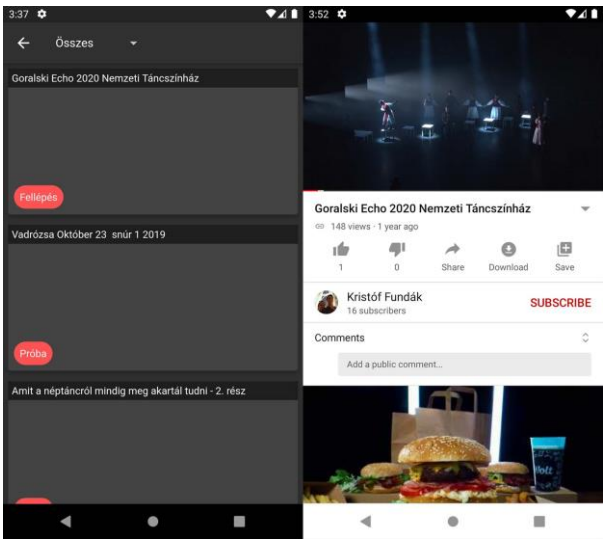
Eseménynaptár felületen a naptári napok jobb alsó sarkában látszódik, hogy hány darab esemény van az adott napon.



44. ábra - Eseménynaptár megvalósítás

5.2.7 Videók megjelenítés oldala

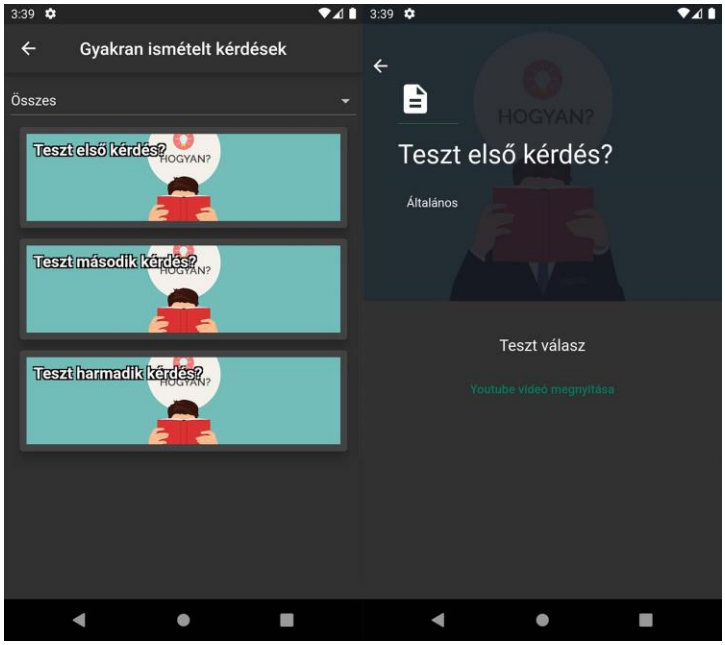
Videók megjelenítési oldala a beágyazott YouTube videót megjelenítésére szolgál, lehetőség van a bal felső sarokban kategóriára szűrni a listát. Adattagra való nyomás után, alapértelmezetten az alkalmazás megpróbálja megnyitni a YouTube alkalmazást, amennyiben nincs telepítve, az alapértelmezett böngészőben megnyitja a videót.



45. ábra - Videók megvalósítás

5.2.8 Gyakran ismételt kérdések felülete

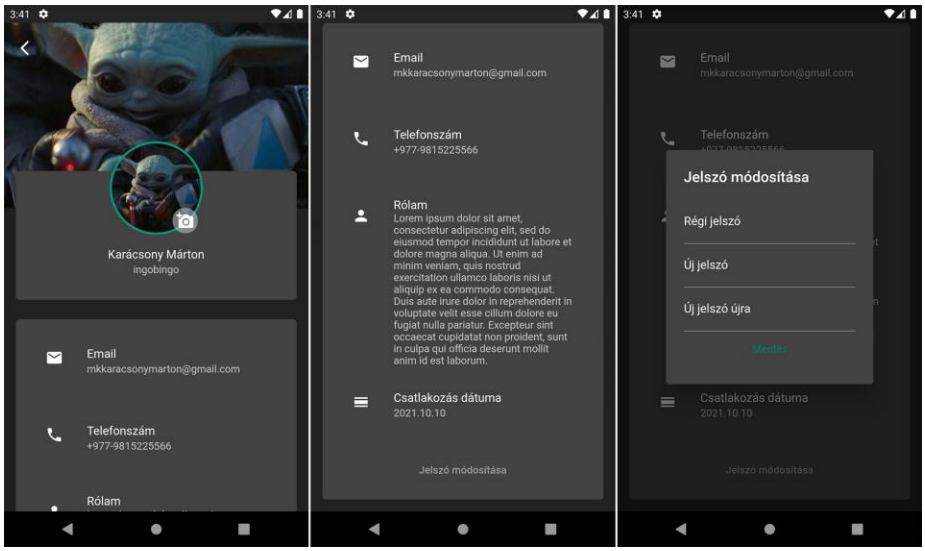
Gyakran ismételt kérdések felületen szintén, az adatbázisban lévő adatok kerülnek listázásra. Ezen a felületen is van lehetőség a listát szűkíteni kategória szerint. Adott kérdésre való nyomás után, megjelenik a jobb oldalon látható GY.I.K. megjelenítő oldal.



46. ábra - GY.I.K. megvalósítás

5.2.9 Profil oldal

Profil oldalon az alábbiakban látható adatok kerültek elhelyezésre:

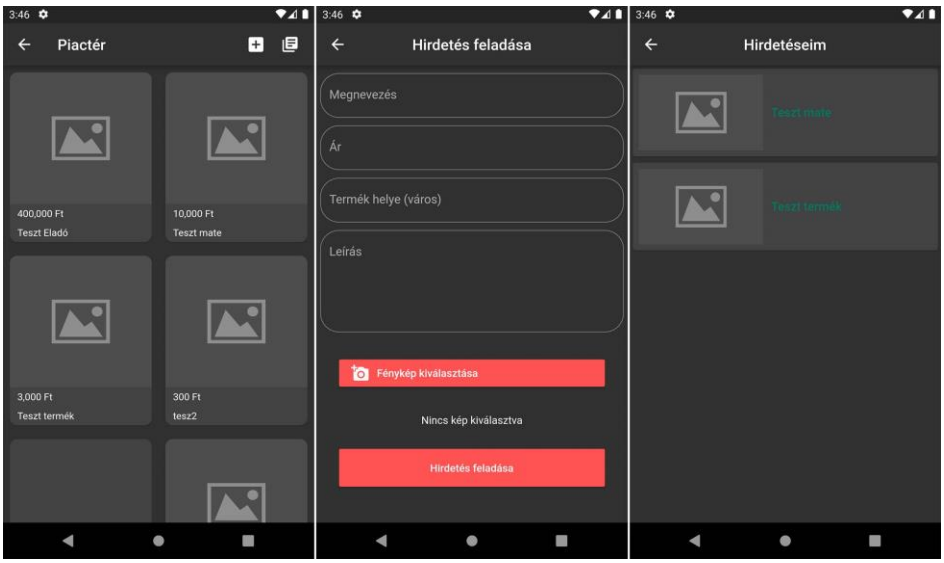


47. ábra - Profil oldal megvalósítás

Funkcionalitását tekintve lehetőség van jelszót és profilképet módosítani.

5.2.10 Piactér

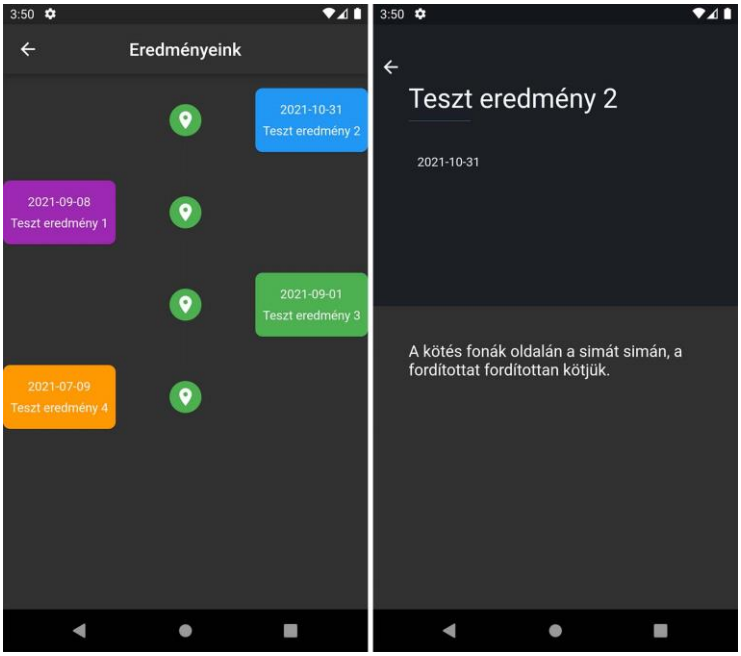
A jelenleg eladó termékek listája található a jobb oldali képen. A középsőn egy hirdetés feladása űrlap felület található, ahova a lista felület jobb felső sarkában lévő plusz gombra való nyomás után juthatunk. Az utolsó felület pedig a saját hirdetések listázása és szerkesztése kapott helyet.



48. ábra - Piactér megvalósítás

5.2.11 Eredményeink

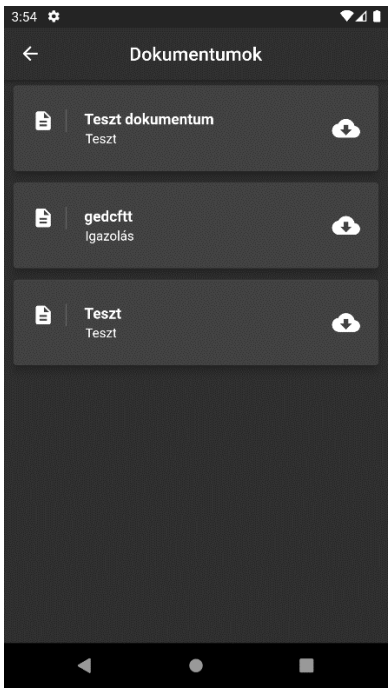
Az eredményeink egy egyszerűbb felülettel, időrend szerűen mutatja meg, hogy milyen eredmények lettek rögzítve az alkalmazásban. Jobb oldali képernyőképen látható egy eredmény megjelenítő oldala.



49. ábra - Eredmények megvalósítás

5.2.12 Dokumentumok

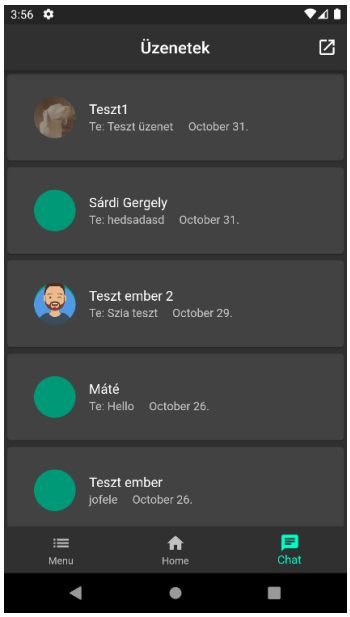
Feltöltött dokumentumok oldalon lehetőség van letölteni a feltöltött anyagokat.



50. ábra - Dokumentumok megvalósítás

5.2.13 Chat üzeneteket listázó oldal

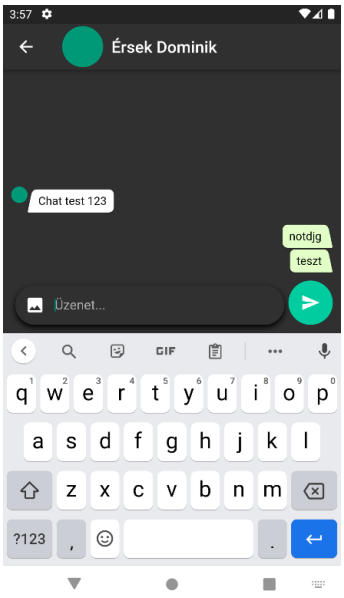
Időrendileg csökkenő sorrendben láthatóak a chat szobák. Egy kártyán látható, hogy ki mikor és mit küldött utoljára.



51. ábra - Chat üzenetek megvalósítás

5.2.14 Chat szoba oldal

Egy adott adattagra való nyomás után, a következő chat felületet kapjuk:



52. ábra - Chat szoba megvalósítás

Az alábbi chat felületen, egy átlátható kommunikációs felületet láthatunk. Bal oldalon a chat partnertől kapott üzenetek, a jobb oldalon pedig az álltalunk elküldött üzenetek. Lehetőség van kép küldésére is.

6 FELHASZNÁLÓI TESZTELÉS

6.1 Bejelentkezés és jogosultsági szint tesztelése

Az alkalmazásban E-mail cím és jelszó párossal van lehetőség bejelentkezni. Ezeknek a folyamatoknak a teszteléséhez a következő tesztesetek kerültek definiálásra:

1. Login jó adatokkal SuperAdmin fiókba
2. Login jó adatokkal sima fiókba
3. Login kísérlet rossz bejelentkezési adatokkal

Teszt sorszáma.	Teszteset	Bemenet	Eredmény	Végeredmény
1.	Login jó adatokkal SuperAdmin fiókba	E-mail cím: mkkaracsonymarton@gmail.com Jelszó: TesztJelszó123	Sikeres bejelentkezés, adminisztrátori menüpont látszódik és annak funkció használhatóak.	Helyes működés
2.	Login jó adatokkal sima fiókba	E-mail cím: sardi.gergo@gmail.com Jelszó: TesztJelszó123	Sikeres bejelentkezés, adminisztrátori menüpont NEM látszódik.	Helyes működés
3.	Login kísérlet rossz bejelentkezési adatokkal	E-mail cím: mkkaracsonymart@gmail.com Jelszó: TesztRosszJelszó123	Sikertelen bejelentkezés, hibajelző snackbar felvillant.	Helyes működés

1. táblázat - Bejelentkezés és jogosultsági szint tesztelése

6.2 Admin oldali adatmódosítások tesztelése

A széles Admin funkcionalitásban lehetőség van egyes adatokat listázni, létrehozni, módosítani és törölni. Ezeknek a tesztelése nélkülözhetetlen az alkalmazás zökkenőmentes működése érdekében.

Admin felület tesztelésére a következő tesztesetek lettek definiálva.

1. Felhasználó módosítása
2. Felhasználó törlése
3. Esemény létrehozása

4. Esemény módosítása
5. Események törlése
6. Csoport létrehozása
7. Csoport törlése
8. Videó létrehozása
9. Eredmények létrehozása
10. Eredmények törlése
11. Eredmények módosítása
12. GY.I.K. létrehozása
13. GY.I.K. módosítása
14. GY.I.K. törlése

Teszt sorszáma.	Teszteset	Bemenet	Eredmény	Végeredmény
1.	Felhasználó módosítása	Az adminisztráció alatt lévő „Felhasználók kezelése” menüpont alatt kiválasztjuk „Karácsony Márton” nevű felhasználót és megváltoztatjuk a csoportját.	Sikeres a mentés, zöld snackbar felvillant az alkalmazás alján a következő szöveggel: „Felhasználó módosítása sikeres”.	Helyes működés
2.	Felhasználó törlése	Az adminisztráció menüpont alatt lévő „Felhasználók kezelése” menüpont alatt kiválasztjuk „Teszt1” nevű felhasználót és balra húzva a törlés gombra nyomunk.	Sikeres a mentés, zöld snackbar felvillant az alkalmazás alján a következő szöveggel: „Felhasználó törlése sikeres”.	Helyes működés
3.	Esemény létrehozása	Az adminisztráció menüpont alatt lévő „Események kezelése” menüponton belül a bal alsó sarokban lévő + jelre nyomunk és kitöltjük a megfelelő adatokat.	Esemény neve, helyszíne, leírása, kezdete, vége paraméter megadása után a létrehozás gombra kattintva. Sikeres mentés jelzés érkezik a képernyő aljára.	Helyes működés
4.	Esemény módosítása	Az adminisztráció menüpont alatt lévő „Események kezelése” menüponton belül a kiválasztjuk a „2021-11-07 15:33- Teszt” eseményt és módosítjuk az esemény nevét.	Sikeres a mentés, zöld snackbar felvillant az alkalmazás alján a következő szöveggel: „Esemény módosítása sikeres”.	Helyes működés
5.	Események törlése	Az adminisztráció menüpont alatt lévő „Események kezelése” menüpont alatt kiválasztjuk „2021-11-07	Sikeres a törlés, zöld snackbar felvillant az alkalmazás alján a következő szöveggel:	Helyes működés

		15:33-Teszt” nevű eseményt és balra húzva a törlés gombra nyomunk.	„Esemény törlése sikeres”.	
6.	Csoport létrehozása	Az adminisztráció menüpont alatt lévő „Csoport kezelés” menüponton belül a bal alsó sarokban lévő + jelre nyomunk és kitöltjük a megfelelő adatokat. Csoport neve paraméter megadása után a létrehozás gombra nyomunk.	Sikeres mentés jelzés érkezik a képernyő aljára.	Helyes működés
7.	Csoport törlése	Az adminisztráció menüpont alatt lévő „Csoport kezelés” menüpont alatt kiválasztjuk „Kis Vadvirág” nevű eseményt és balra húzva a törlés gombra nyomunk. Felugró ablakban pedig a megerősítésre.	Csoport lista frissül és a törölt csoport már nem található meg a listában.	Helyes működés
8.	Videó létrehozása	Az adminisztráció menüpont alatt lévő „Videó feltöltése” menüponton belül kiválasztjuk a videó kategóriát és beillesztjük a YouTube Url címet.	Sikeres mentés jelzés érkezik a képernyő aljára.	Helyes működés
9.	Eredmények létrehozása	Az adminisztráció menüpont alatt lévő „Eredmények kezelése” menüponton belül a bal alsó sarokban lévő + jelre nyomunk és kitöltjük a megfelelő adatokat. Eredmény címe, leírása paraméter megadása után, feltöltjük a kívánt borítóképet és kiválasztjuk az eredmény időpontját és a létrehozás gombra nyomunk.	Sikeres mentés jelzés érkezik a képernyő aljára.	Helyes működés
10.	Eredmények törlése	Az adminisztráció menüpont alatt lévő „Eredmények kezelése” menüpont alatt kiválasztjuk „Teszt eredmény 1” nevű eseményt és balra húzva a törlés gombra nyomunk. Felugró ablakban pedig a megerősítésre.	Sikeres törlés jelzés érkezik a képernyő aljára.	Helyes működés

11.	Eredmények módosítása	Az adminisztráció menüpont alatt lévő „Eredmények kezelése” menüponton belül a kiválasztjuk a „Teszt eredmény 2” eredményt és módosítjuk az eredmény címét.	Sikeres módosítás jelzés érkezik a képernyő aljára.	Helyes működés
12.	GY.I.K. létrehozása	Az adminisztráció menüpont alatt lévő „GY.I.K. kezelése” menüponton belül a bal alsó sarokban lévő + jelre nyomunk és kitöltjük a megfelelő adatokat. GY.I.K. kérdés, válasz, YouTube videó url, kategória paraméter megadása után a létrehozás gombra nyomunk.	Sikeres mentés jelzés érkezik a képernyő aljára.	Helyes működés
13.	GY.I.K. módosítása	Az adminisztráció menüpont alatt lévő „GY.I.K. kezelése” menüponton belül a kiválasztjuk a „Teszt kérdés?” gyakran ismételt kérdést és módosítjuk a választ.	Sikeres módosítás jelzés érkezik a képernyő aljára.	Helyes működés
14.	GY.I.K. törlése	Az adminisztráció menüpont alatt lévő „GY.I.K. kezelése” menüpont alatt kiválasztjuk „Teszt kérdés?” nevű eseményt és balra húzva a törlés gombra nyomunk. Felugró ablakban pedig a megerősítésre.	Sikeres törlés jelzés érkezik a képernyő aljára.	Helyes működés

2. táblázat - Admin oldali adatmódosítások tesztelése

7 ÖSSZEFOGLALÁS

A szakdolgozatom egy sportegyesületeknek szánt belső folyamatokat támogató alkalmazás tervezéséről és fejlesztéséről szól, és ennek folyamatán megy végig az irodalomkutatástól a részletes design és adatbázis terven át az alkalmazás lefejlesztéséig. Az alkalmazás Flutter keretrendszerrel készült és a háttérrendszert egy Firebase serverless architektúra szolgálja ki, beleértve az authentikációt és a NoSQL adatbázist is. Az alkalmazásban az adatok dinamikusan változnak, ezáltal lehetővé teszik az egyesület mindennapi ügyeinek intézését.

A szakdolgozat és abban megvalósított projekt rávilágított arra, hogy egy szoftver soha nincs kész teljesen. A különböző modulok folyamatos pontosítása, továbbfejlesztése egy mindennapi feladat és komoly szoftver után követést igényel. Továbbá, hogy mennyire fontos a tervezés és a vállalat, jelen esetben az egyesület folyamatainak részletes feltárása és átbeszélése.

A szakdolgozatban tárgyalt alkalmazást egy valós sportegyesületen és annak folyamatain alapszik, a fejlesztési folyamat során több, az egyesületben dolgozó és sportoló is segített a tervezési és a tesztelési feladatokban. Az elkészült alkalmazásban nagy jövőképet lát az egyesület és további modulokkal szeretné majd kiegészíteni a jövőben, mint például az tagdíj befizetések automatizálása.

8 SUMMARY

My thesis is about the design and development of an internal process support application for sports clubs, and it goes through the process from literature research, detailed design and database plan to the development of the application. The application is built using the Flutter framework and the backend system is served by a Firebase serverless architecture, including authentication and a NoSQL database. The data in the application changes dynamically, allowing the association to manage its day-to-day business.

The thesis and the project it is part of have shown that software is never complete. The continuous refinement of the different modules is a daily task and requires a serious software follow-up. Furthermore, the importance of accurate planning and, in this case, the detailed exploration and discussion of the company's processes.

The application discussed in this thesis is based on a real sports association and its processes, and during the development process several people working in the association and athletes helped with the design and testing tasks. The association sees a great future potential in the completed application and would like to add further modules in the future, such as the implementation of association payments.

9 Irodalomjegyzék

[1] Gartner, „Gartner.com,” 15 02 2017. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2017-02-15-gartner-says-worldwide-sales-of-smartphones-grew-7-percent-in-the-fourth-quarter-of-2016. [Hozzáférés dátuma: 11 12 2021].

[2] J. Fayzullaev, „Theseus.fi,” 2018. [Online]. Available: https://www.theseus.fi/bitstream/handle/10024/148975/thesis_Jakhongir_Fayzullaev.pdf?sequence=1&isAllowed=y. [Hozzáférés dátuma: 04 2021].

[3] „React Native,” Facebook, Inc., 2021. [Online]. Available: https://reactnative.dev/. [Hozzáférés dátuma: 04 2021].

[4] „Crane.hu,” 11 11 2019. [Online]. Available: https://www.crane.hu/progressive-web-apps-a-jovo-mobilappjai-pwa. [Hozzáférés dátuma: 04 2021].

[5] Google, „Google Developers Blog,” Google, 3 03 2021. [Online]. Available: https://developers.googleblog.com/2021/03/announcing-flutter-2.html. [Hozzáférés dátuma: 04 2021].

[6] W. Wu, „React Native vs Flutter, cross-platform mobile - Theseus.fi,” 01 03 2018. [Online]. Available: https://www.theseus.fi/bitstream/handle/10024/146232/thesis.pdf?sequence=1. [Hozzáférés dátuma: 04 2021].

[7] „Flutter: Stateful vs Stateless Widget - Medium.com,” 6 06 2019. [Online]. Available: https://medium.com/flutter-community/flutter-stateful-vs-stateless-db325309deae. [Hozzáférés dátuma: 04 2021].

[8] „cloudflare.com,” 2021. [Online]. Available: https://www.cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/. [Hozzáférés dátuma: 12 12 2021].

[9] „What is BaaS? | Backend-as-a-Service vs. serverless - Cloudflare.com,” [Online]. Available: https://www.cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/. [Hozzáférés dátuma: 04 2021].

[10] „Iflexion.com,” 17 11 2020. [Online]. Available: https://www.iflexion.com/blog/mobile-backend-as-a-service. [Hozzáférés dátuma: 04 2021].

[1] „Back4App,” [Online]. Available: https://www.back4app.com/. [Hozzáférés dátuma: 04 2021].

[2] „Kipróbáltuk a Firebase-t - Medium.com,” 7 02 2017. [Online]. Available: https://medium.com/mito/kiprobaltuk-firebase-analytics-database-fabric-d6b208d6288f. [Hozzáférés dátuma: 04 2021].

[3] „Bluewhaleapps,” 18 12 2019. [Online]. Available: https://bluewhaleapps.com/blog/7-reasons-to-choose-google-cloud-firestore-as-your-database-solution. [Hozzáférés dátuma: 04 2021].

[4] „Cloud Functions for Firebase,” 2021. [Online]. Available: Cloud Functions for Firebase. [Hozzáférés dátuma: 12 12 2021].

[5] „Firebase Google,” [Online]. Available: https://firebase.google.com/docs/analytics. [Hozzáférés dátuma: 04 2021].

[6] „Crashlytics- Firebase Google,” [Online]. Available: https://firebase.google.com/docs/crashlytics. [Hozzáférés dátuma: 04 2021].

[7] K. T. Kacsukné Bruckner Lívia, BEVEZETÉS AZ ÜZLETI INFORMATIKÁBA, Akadémiai Kiadó, 2019.

[8] SAP, „sap.com,” 11 12 2021. [Online]. Available: https://www.sap.com/hungary/products/business-one.html?pdf-asset=2e4955be-937c-0010-82c7-eda71af511fa&page=1. [Hozzáférés dátuma: 11 12 2021].

[9] AbasErp, „abas-erp-com,” 11 12 2021. [Online]. Available: https://abas-erp.com/en/erp-software. [Hozzáférés dátuma: 11 12 2021].

[20] „A 25 legnepszerubb ERP rendszer,” [Online]. Available: https://www.minuszos.hu/wp-content/uploads/doc/a-25-legnepszerubb-erp-rendszer.pdf. [Hozzáférés dátuma: 04 2021].

[21] K. B. L. Tamás, „Bevezetés az üzleti informatikába [Digitális kiadás.],” Akadémiai Kiadó, Budapest, 2019.

ÁBRAJEGYZÉK

TÁBLÁZATOK JEGYZÉKE