**Firebase Alapú Alkalmazásfejlesztés**

A félév során a cél a Firebase platform megismerése, majd egy jegyzetkészítő alkalmazás készítése Android platformon, amely támogatja a felhasználók autentikációját, illetve képes a felhasználók által készített jegyzeteket a Firebase platformon eltárolni.

Az alkalmazásnak képesnek kell lennie a következőkre:

* Új felhasználó regisztrálása email cím alapján.
* A regisztráció és az email cím megerősítése után az alkalmazásba való bejelentkezés.
* Új jegyzetek létrehozása.
* Meglévő jegyzetek szerkesztése.
* Meglévő jegyzetek törlése.
* A jegyzetek elmentése a Firebase platformra.

A félév során a következő főbb technológiákat használtam:

* Android
* Kotlin
* Android Studio
* Firebase (Authentication, Firestore)
* XML UI

Az alkalmazáshoz az MVC (Model-View-Controller) tervezési mintát használtam.

Az alkalmazás a következő főbb osztálykönyvtárakat használja:

* Firebase Analytics
* Firebase Auth
* Firebase Firestore
* Android Util
* Android OS

A program megírása Android Studio segítségével történt, Kotlin nyelven.

Először is néhány szó a Firebase platformról:

A Firebase egy alkalmazásfejlesztési platform, amelyet 2011-ben alapított James Tamplin and Andrew Lee, és később, 2014 a Google vásárolt meg. Segítségével könnyedén építhetünk és növelhetjük alkalmazásaink és játékaink népszerűségét. A Firebase számos szolgáltatást kínál, amelyek közül néhány a következő:

* Cloud Firestore: Egy NoSQL adatbázis, amely lehetővé teszi az adatok szinkronizálását valós időben a felhasználói eszközök között.
* Authentication: Egyszerű és biztonságos hitelesítési megoldásokat kínál több platformon.
* Crashlytics: Segít a hibák nyomon követésében és a teljesítmény javításában.
* Google Analytics: Részletes felhasználói adatokat és elemzéseket biztosít.
* Cloud Messaging: Lehetővé teszi a push értesítések küldését.
* A/B Testing: Segít optimalizálni az alkalmazás felhasználói élményét különböző tesztelési változatokkal.

A Firebase támogatja az iOS, Android, web, Flutter, Unity és C++ platformokat. A Firebase projektek a Google Cloud által vannak támogatva, így alkalmazásainkat sok felhasználó esetén is könnyedén tudjuk skálázni.

A Firebase árazása:

A Firebase két fő árazási tervet kínál: a Spark és a Blaze tervet.

* Spark terv: Ez egy ingyenes terv, amely számos Firebase termék használatát teszi lehetővé korlátozott mennyiségben. Nem szükséges fizetési információt megadni a Spark terv használatához.
* Blaze terv: Ez egy fizetős terv, amely lehetővé teszi a Firebase termékek nagyobb mennyiségű használatát2. A Blaze terv használatához szükséges egy fizetési számla hozzárendelése.

A Firebase termékek közül néhány ingyenesen használható mindkét tervben, míg másokért fizetni kell. A fizetős termékek használatának díja a használat mértékétől függ.

A következő szolgáltatások mindig ingyenesek:

* A/B Testing
* Analytics
* App Check
* App Distribution
* Cloud Messaging (FCM)
* Crashlytics
* In-App Messaging
* Performance Monitoring
* Remote Config
* Gemini

A következő szolgálgatásokért pedig egy adott határeréték felett (pl. Authentication esetén egy bizonyos számú felhasználó felett) fizetni kell:

* Authentication
* Cloud Firestore
* Cloud Functions
* Cloud Storage
* Hosting
* Firebase ML
* Realtime Database
* Test Lab
* Google Cloud

**Firebase projekt létrehozása:**

Firebase Projekt Létrehozása: A fejlesztés elkezdése előtt az első lépés a Firebase projekt létrehozása volt. Ezt a következő módon tettem meg.

1. Elnavigáltam a Firebase weboldalára. Ott a Google fiókba való bejelentkezés után rákattintottam a „Get started” gombra.

A screenshot of a website

Description automatically generated

1. A következő képernyőn kiválasztottam a „Create a project” gombot.

A screenshot of a computer

Description automatically generated

1. Az ezt követő képernyőn adtam egy nevet a projektnek (NoteApp).

A screenshot of a computer

Description automatically generated

1. Ezután a következő képernyőn bekapcsoltam a Google Analytics-et. Az Analytics segítségével később a projekt során lehetséges lesz a különböző események (pl. gomra kattintás) logolására a Firebase platformra.

A screenshot of a computer

Description automatically generated

1. Az ezt követő képernyőn kiválasztottam Magyarországot és elfogadtam a szerződési feltételeket, majd a „Create project” gombra kattintva létrehoztam a projektet.

A screenshot of a computer

Description automatically generated

1. A projekt elkészülte után rákattintottam a „Continue” gombra.

A screenshot of a computer

Description automatically generated

**Android Projekt létrehozása:**

A Firebase Projekt létrehozása után a következő lépés az Android alkalmazás létrehozása lesz. Ehhez az Android Studio segítségét és a Kotlin programozási nyelvet vettem igénybe.

1. Az Android Studio telepítése és elindítása után rákattintottam a „New Project” gombra a projekt létrehozásához.

A screenshot of a computer

Description automatically generated

1. A következő képernyőn az „Empty Views Acitvity” lehetőséget választottam.

A screenshot of a computer

Description automatically generated

1. Ezután adtam egy nevet és egy package nevet a projektnek Ezen kívül minden más opciót alapértelmezetten hagytam. A package nevet érdemes megjegyezni vagy felírni, mivel később annak a segítségével lehet a Firebase projekthez hozzáadni az alkalmazást.

A screenshot of a computer

Description automatically generated

**Firebase hozzáadása az Android projekthez:**

Az Android projekt létrehozása után a következő lépésként hozzáadtam a Firebase-t a projekthez.

1. A Firebase projekt weboldalán az „Android” ikonra kattintottam.

A screenshot of a computer

Description automatically generated

1. A következő képernyőn megadtam a korábban beállított package nevet, illetve adtam egy becenevet az alkalmazásnak.

A screenshot of a computer

Description automatically generated

1. A „Register app” gomb megnyomása után rákattintottam a „Download google-services.json” gombra. Ezután követtem a gomb alatti utasításokat. Az Android Studioban projekt nézetre váltva a megfelelő helyre mozgattam a letöltött fájlt.

A screenshot of a computer

Description automatically generated

A screenshot of a notepad

Description automatically generated

1. A „Next” gomb megnyomása után hozzáadtam a projektszintű **build.gradle.kts** fájlhoz a következő sort. Ezután megnyomtam a „Sync now” gombot a szinkronizálás lefuttatása érdekében.

A screenshot of a computer

Description automatically generated

Majd hozzáadtam az app szintű **build.gradle.kts** fájlhoz a következő sorokat. A „Sync now”-t természetesen itt is megnyomtam.

A screenshot of a computer

Description automatically generated

Itt később a „dependencies” részhez lehet hozzáadni a használni kívánt library-ket. (pl. firestore)

Ezután ténylegesen elkezdődhetett az Android alkalmazás fejlesztése.

**Autentikáció engedélyezése:**

Az autentikáció megvalósítása igen fontos lépés volt a projekt során, ugyanis ez teszi lehetővé, hogy a különböző felhasználók csak a saját jegyzeteiket lássák az alkalmazásban.

Autentikációt a következő módon adtam hozzá a projekthez.

1. Az Android Studio-n belül a fejlécen rákattintottam a „Tools” menüpontra, majd kiválasztottam a Firebase opciót

A screenshot of a computer

Description automatically generated

1. A jobb oldalt megnyílt ablakban kiválasztottam az „Authentication” és azon belül az „Authenticate using a custom authentication system” menüpontot.

A screenshot of a computer

Description automatically generated

1. A következő képernyőn kiválasztottam az „Add Firebase Authetication SDK to your app” menüpontot. A felugró ablakban elfogadtam a változásokat.

A screenshot of a custom authentication system

Description automatically generated

1. Ezután engedélyeztem az autentikációt a Firebase projektben a következőképpen. A bal oldali menüben rákattinottam a „Build”, majd azon belül az „Authentication” menüpontra.

A screenshot of a computer

Description automatically generated

Ezután rákattiontottam a „Get started” gombra.

A screenshot of a computer

Description automatically generated

Következőként az „Email/Password” menüpontot választottam ki.

A screenshot of a computer

Description automatically generated

Az ezt követő képernyőn engedélyeztem az „Email/Password” opciót, majd a „Save” gombra kattintva elmentettem a változásokat.

A screenshot of a computer

Description automatically generated

**Autentikáció implementálása:**

Az ezt követő lépés az autentikáció implementálása volt az Android alkalmazásban:

1. Hozzáadtam egy új Activity-t az alkamazáshoz. Neve a következő: „CreateAccountActivity”, típusa „Empty Views Activity”

A screenshot of a computer

Description automatically generated

1. Az újonnan létrehozott Activity layout.xml fájlját megnyitva elkészítettem a regisztrációs ablakot. Törekedtem arra, hogy valamilyen szinten esztétikus kinézetű legyen az alkalmazás. Ennek a kódja itt található. A layout fájlok a képen látható könyvtárban találhatók:

A screenshot of a computer

Description automatically generated

1. Hasonlóképpen elkészítettem a bejelentkező oldal Activity-jét és layout-ját is. A neve a következő lett: „LoginActivity”, típusa szintén „Empty Views Activity”. A layout kódja itt található.
2. Ezután engedélyeztem a viewBinding-ot a projektben. (App szintű build.gradle.kts fájl.) Ez a felhasználói felület elemeire való hivatkozásánál hasznos az Activity-ken belül, mivel így nem kell külön minden elemet deklarálni.

A screenshot of a computer code

Description automatically generated

1. A „CreateAccountActivity” kódját átírtam le a következőre.

Mit is csinál ez a kód: Beállítottunk egy listenert a „Create Account” gombra. Ez meghívja a „createAccount” metódust, amely kiolvassa a mezőkből az email címet és a jelszót, majd ezeket validálja. Ha a validáció sikeres, meghívja a „createAccountInFirebase” metódust, amely létrehozza a felhasználót a Firebase-ben és küld egy megerősítő emailt. A bejelentkezés csak az email cím megerősítése után lehetséges.

1. A „LoginActivity” kódja a következő lett. Itt szintén validálja az email címet, majd, ha létezik ezzel az email címmel regisztrált felhasználó és helyes a jelszó, megtörténik a bejelentkezés.

**Jegyzetek létrehozása:**

A következő lépés a jegyzet létrehozás megvalósítása volt.

1. Ehhez először felvettem egy új Activity-t, NoteDetailsActivity néven. Ennek a típusa szintén „Empty Views Activity”
2. Hozzáadtam a MainActivity layout fájljához egy FloatingActionButton-t.
3. A gombra rákkatintva áttér majd az új jegyzet nézetre az alkalmazás. Ehhez a MainActivity-n belül fel kell iratkoznunk a gomb eseménykezelőjére, majd el kell indítani a NoteDetailsActivity-t.
4. A NoteDetailsActivity layout fájl a következő lett.
5. A NoteDetailsActivity kódja pedig a következő.

**Firestore adatbázis létrehozása:**

A következő lépés a Firestore adatbázis létrehozása és hozzáadása a projekthez.

1. A Firestore weboldalán megkerestem a „Firestora Database” menüpontot, majd rákattintottam a „Create Database” gombra.

A screenshot of a computer

Description automatically generated

1. Választottam egy helyszínt az adatbázisnak, majd kiválasztottam a „Test Mode” opciót.
2. Eztuán hozzáadtam a Firestore dependecy-t az alkalmazáshoz. Ehhez fent a Tools, majd azon belül a Firebase menüpontra kattintottam. A jobb oldalt felugró ablakban megkerestem a Cloud Firestore menüpontot, majd a „Get started with Cloud Firestore” opciót.

A screenshot of a computer

Description automatically generated

1. Ezek után a következő menüben az „Add the Cloud Firestore SDK to your app” gombra kattintottam. Ekkor a Firestore dependency automatikusan hozzáadásra került az app szintű build.gradle.kts fájlhoz.

A screenshot of a computer

Description automatically generated

**Jegyzetek elmentése a Firestore adatbázisba:**

A következő lépés az újonnan létrehozott jegyzetek elmentése volt az adatbázisba. Ehhez a „NoteDetailsAcitivity”-hez hozzáadtam a lent látható metódusokat.

1. A „Save” gombra kattinva meghívjuk a „saveNote” metódust.

A close-up of a computer screen

Description automatically generated

1. A „saveNote” metódusban, ha nem üres a jegyzet címe, létrehozzuk az új jegyzetet, majd meghívjuk a „saveNoteToFirestore” metódust. Üres jegyzet cím esetén egy hibaüzenetet jelenítünk meg a cím mezőn.

A computer screen shot of text

Description automatically generated

1. Ezután hozzáadjuk a jegyzetet az adatbázishoz. Itt ellenőrizzük, hogy létezik-e a jegyzet, ennek jelenleg még nincs szerepe, de a meglévő jegyzetek szerkesztésénél is ugyanezt a metódust fogja a program használni. A metódus logolja a sikeres, illetve a sikertelen adatbázis műveletet, illetve egy toast-ot is megjelenít, amely szövegként tartalmazza a művelet sikerességét.

A screenshot of a computer program

Description automatically generated

**Jegyzetek listázása:**

Az ezt követő lépés a meglévő jegyzetek listázása lesz.

1. Ehhez először is szükség lesz egy Adapter osztályra. Az Adapter osztály segítségével fogja tudni az alkalmazás kilistázni a jegyzeteket a MainAcitivity-ben, illetve a jegyzetekre való kattintást is itt kezeljük majd. Az egyes jegyzetekre kattintáskor a NoteDetailsAcitivty-t indító Intent-hez hozzáadjuk a jegyzet adatait. Ennek a kódja így néz ki:

A screenshot of a computer program

Description automatically generated

A hozzá tartozó layout fájl pedig így:

A screenshot of a computer program

Description automatically generated

1. Ezután eljött az idő, hogy feltöltsük a MainActivity-ben lévő RecyclerView-t a jegyzetek adataival. Ez a következőképpen történik:

Az onCreate végén meghívjuk a „fillRecyclerView” metódust.

A computer screen shot of a computer code

Description automatically generated

A metódus lekérdezi az adatbázisból az épp bejelentkezett felhasználó jegyzeteit a „timestamp” mező szerint csökkenő sorrendben, majd feltölti a RecyclerView-t az előbb létrehozott NoteAdapter osztály segítségével.

A screenshot of a computer program

Description automatically generated

1. A jegyzetekre való kattintás után a NoteDetailsAcitivty indul el, de most fel kell töltenünk a mezőket a kiválasztott jegyzet adataival. Ezt az onCreate metódusban tehetjük meg a következő módon: Lekérdezzük az intent-ből a hozzáadott docid stringet, és ha nem üres, akkor feltöltjük a mezőket a megfelelő adatokkal, illetve a felső sávban az „Edit Note” szöveget jelenítjük meg. Ezen kívül a törlés gombot is láthatóvá tesszük.

A screenshot of a computer code

Description automatically generated

**Jegyzetek törlése:**

Az ezt követő lépés a jegyzetek törlésének megvalósítása. Ezt a következő módon implementáltam:

1. Hozzáadtam egy törlés gombot a NoteDetailsActivity layout-jához.

A screenshot of a computer code

Description automatically generated

1. Beállítottam a listener-jét a NoteDetailsActivity-ben.

A close-up of a computer screen

Description automatically generated

1. A deleteNote metódusban kitörlöm a jegyzetet az adatbázisból.

A screenshot of a computer program

Description automatically generated

**Analytics**:

Ezt követte az Analytics bekötése. Itt csak néhány fontosabb funkció (pl. jegyzet létrehozás, törlés, kijelentkezés) logolását valósítottam meg. Külön kihívás volt a tesztelésnél, hogy logolt információk Firebase-en történő megjelenése akár 24 órát is igénybe vehet.

Új jegyzet létrehozása gomb esetén a következőképpen jártam el.

1. A gomb eseménykezelőjében meghívom a noteDetails metódust.

A close-up of a website

Description automatically generated

1. A noteDetails metódusban meghívom a logButtonClick metódust, átadva neki string-ként egy paramétert.

A screenshot of a computer

Description automatically generated

1. Végül a logButtonClick metódusban elküldöm a Firebase-re, hogy milyen esemény történt.

A computer screen shot of a program code

Description automatically generated

Ezen kívül hasonlóan jártam el a következő eseményeknél:

* Logout gomb megnyomása
* Jegyzet sikeres/sikertelen mentése
* Meglévő jegyzet szerkesztése
* Jegyzet törlése

Összességében sok mindent tanultam a félév során. A A célom a projekt továbbfejlesztése szakdolgozat formájában, esetleg egy teljesen új projekt létrehozása.