**Firebase Alapú Alkalmazásfejlesztés**

A félév során a cél a Firebase platform megismerése volt, amelyet egy Android alkalmazásba történő integráció során végeztem el.

A választott alkalmazás egy jegyzet készítő alkalmazás volt. 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 program megírása Android Studio segítségével történt, Kotlin nyelven.

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

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

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.

A következő lépés a jegyzet hozzáadás megvalósítása.

1. Ehhez először felvettem egy új Activity-t, NoteDetailsActivity néven.
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.

A következő lépés a Firestora 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.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A következő lépés az újonnan létrehozott jegyzetek elmentése volt az adatbázisba.

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.

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 metódus lesz használva.

A screenshot of a computer program

Description automatically generated

Az ezt követő lépés a meglévő jegyzetek kilistázása lesz.

1. Ehhez először is szükség lesz egy Adapter osztályra. Az Adapter osztály segítségével fogjuk 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, 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:

A screenshot of a computer code

Description automatically generated

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) 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