

**SAPIENTIA ERDÉLYI MAGYAR TUDOMÁNYEGYETEM
MAROSVÁSÁRHELYI KAR,
INFORMATIKA SZAK**



SAPIENTIA
ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM

A biciklis közösség szociális felülete

DIPLOMADOLGOZAT

Témavezető:
Dr. Kupán Pál,
Egyetemi docens,

Végzős hallgató:
Fodor László-Carlos

Novák Pálma-Rozália
Egyetemi tanársegéd

2024

**UNIVERSITATEA SAPIENTIA DIN CLUJ-NAPOCA
FACULTATEA DE ȘTIINȚE TEHNICE ȘI UMANISTE,
SPECIALIZAREA INFORMATICĂ**



**UNIVERSITATEA
SAPIENTIA**

Platforma socială a comunității cicliste

LUCRARE DE DIPLOMĂ

Coordonator științific:
Dr. Kupán Pál,
Conferențiar universitar

Absolvent:
Fodor László-Carlos

Novák Pálma-Rozália
Asistent universitar

2024

**SAPIENTIA HUNGARIAN UNIVERSITY OF
TRANSYLVANIA**
FACULTY OF TECHNICAL AND HUMAN SCIENCES
COMPUTER SCIENCE SPECIALIZATION



SAPIENTIA
HUNGARIAN UNIVERSITY
OF TRANSYLVANIA

The social platform of the cycling community

BACHELOR THESIS

Scientific advisor:
Dr. Kupán Pál,
Associate professor

Student:
Fodor László-Carlos

Novák Pálma-Rozália
Assistant professor

2024

Declarație

Subsemnata/ul FODOR LÁSZLÓ-CARLOS, absolvent(ă) al/a specializării INFORMATICĂ, promoția 2024... cunoscând prevederile Legii Educației Naționale 1/2011 și a Codului de etică și deontologie profesională a Universității Sapientia cu privire la furt intelectual declar pe propria răspundere că prezenta lucrare de licență/proiect de diplomă/disertație se bazează pe activitatea personală, cercetarea/proiectarea este efectuată de mine, informațiile și datele preluate din literatura de specialitate sunt citate în mod corespunzător.

Localitatea, LIVEZENI

Data: 24.06.2024

Absolvent

Semnătura... Fodor.....

LUCRARE DE DIPLOMĂ

Îndrumător: Drd. Novák Pálma-Rozália Coordonator științific: Conf. dr. Kupán A. Pál	Candidat: Fodor László-Carlos Anul absolvirii: 2024
--	--

a) Tema lucrării de licență:

Proiectarea și dezvoltarea aplicației BicyHub, o platformă de înregistrare și monitorizare a bicicletelor, concentrată pe crearea și susținerea unei comunități de utilizatori care să poată împărtăși informații despre bicicletele lor.

b) Problemele principale tratate:

- Realizarea unui studiu bibliografic privind platformele comunitare și importanța partajării informațiilor între utilizatori.
- Studiul tehnologilor relevante pentru dezvoltarea unei aplicații de gestionare și partajare a bicicletelor.
- Proiectarea și dezvoltarea unei aplicații BicyHub, cu funcționalități esențiale pentru crearea unei comunități de bicicliști.
- Facilitarea interacțiunii dintre utilizatori, permitându-le să împărtășească informații despre bicicletele lor, să vizualizeze istoricul proprietarilor anteriori și să colaboreze în cadrul unei comunități active.

c) Desene obligatorii:

- Diagrame de arhitectură software pentru aplicația realizată
- Diagrame de use case
- Diagrame de secvență
- Diagrame de baze de date

d) Softuri obligatorii:

Dezvoltarea unei aplicații utilizând tehnologii moderne, precum Node.js pentru backend și React Native pentru frontend. Aplicația va include funcționalități pentru înregistrarea și monitorizarea stării bicicletelor, permitând utilizatorilor să își împărtășească experiențele și informațiile despre bicicletele lor. Utilizatorii vor putea vizualiza istoricul proprietarilor anteriori, să ofere și să primească recomandări și să colaboreze pentru întreținerea optimă a bicicletelor. Interfața va fi intuitivă și ușor de utilizat, pentru a încuraja participarea activă a comunității.

e) Bibliografia recomandată:

- Putnam, L. (2018). The Bike Owner's Handbook.
- Samuel Gocník. Device control system and mobile application.
- <https://www.bicycling.com/>

f) Termene obligatorii de consultații: săptămânal

g) Locul și durata practicii: Universitatea „Sapientia” din Cluj-Napoca,
Facultatea de Științe Tehnice și Umaniste din Târgu Mureș

Primit tema la data de: 1 mai 2023

Termen de predare: 24 iunie 2024

Semnătura Director Departament

Semnătura responsabilului programului de studiu *Antal*

Semnătura coordonatorului

Semnătura candidatului *Fodor*

Kivonat

Projektem témája egy biciklis közösségnak szánt alkalmazás, amely egy olyan platformot kínál a felhasználóknak, ahol kifejezhetik szenvedélyüket és érdeklődésüket egy olyan közösség számára, amely hasonlóan elkötelezett a biciklizés iránt.

Köztudott, hogy mindenkinek vannak hobbijai, de olyan esetek is előfordulnak, amikor bizonyos baráti körökben senki sem osztja meg ugyanazt a szenvedélyt egy adott hobbi iránt, egyik ilyen példa lehet a biciklizés. Ezen okból kifolyólag lett létrehozva az alkalmazás, hogy kapcsolatot teremtsen olyan emberek között akik ugyanannyira érdekeltek a biciklizés témajában. Bármilyen jártasak is vagyunk egy adott területen, mindig tanulhatunk valami újat, fejlődhetünk, és bővíthetjük ismereteinket. Nyilvánosan fel lehet tenni kérdéseket, ahol más felhasználók válaszolhatnak és ismertethetik tapasztalataikat az illető témaival kapcsolatban. Valamint, figyelembe véve a bicikli vásárlási lehetőséget, a felhasználónak rendelkezésére áll a privát üzenetváltás kezdeményezése.

Az is említésre méltó, hogy a biciklizés egyik legfontosabb része az útvonalak, amelyeken tűrözünk. Emiatt, az alkalmazás lehetőséget biztosít arra, hogy ezeket az útvonalakat elmentsük és meg is osszuk más felhasználókkal.

Nem szokatlan, hogy más szociális platformokon az emberek képeket osztanak meg olyan dolgokról, amelyekre büszkék. A BicyHub felhasználói is képesek erre, mivel megoszthatják saját bicikli gyűjteményüket egy olyan közegben, ahol az emberek igazán értékelni tudják a tulajdonukat és az apró részleteket, amelyekre csak egy bicikliszerető szemlélő képes.

Kulcsszavak: alkalmazás, szociális platform, közösség, hobbi, bicikli vásárlás

Rezumat

Tema lucrării mele este o aplicație încunjurată de biciclete, care oferă utilizatorilor o platformă socială unde își pot exprima pasiunea și interesul într-o comunitate care este, de asemenea, profund implicată în subiectul ciclismului.

Fiecare persoană are propriile sale hobby-uri, dar există și situații în care în anumite cercuri de prieteni nimici nu împărtășesc aceeași dorință pentru un anumit hobby, ciclismul este un astfel de exemplu. Din acest motiv, a fost creată aplicația pentru a facilita conexiunea între cei interesați de subiectul ciclismului. Indiferent cât de pricepuți suntem într-un anumit domeniu, întotdeauna putem învăța ceva nou, ne putem dezvolta și ne putem extinde cunoștințele. Pot fi adresate public întrebări, unde alții utilizatori pot răspunde și își pot împărtăși experiențele legate de tema respectivă. În plus, având în vedere opțiunea de achiziționare a unei biciclete, utilizatorul are posibilitatea de a iniția mesaje private.

Este demn de menționat că una dintre cele mai importante aspecte ale ciclismului o constituie rutele pe care parcurgem. De aceea, aplicația oferă posibilitatea de a salva și de a împărtăși aceste rute cu alții utilizatori.

Nu este neobișnuit că pe alte platforme sociale oamenii să împărtășească imagini cu lucruri de care sunt mândri. Utilizatorii BicyHub pot face și ei acest lucru, deoarece pot împărtăși propria colecție de biciclete într-un mediu în care oamenii pot aprecia cu adevărat proprietățile lor și detaliile mai sofisticate pe care doar un iubitor de biciclete le poate observa.

Cuvinte cheie: aplicație, platformă socială, comunitate, hobby, achiziționarea bicicletelor

Abstract

The topic of my thesis is a bicycle-centric application that provides users with a social platform to express their passion and interest in a community equally committed to cycling.

Everyone has their own hobbies, but there are cases where in certain friend circles, no one shares the same passion for a specific hobby, cycling is one such example. Therefore, the application was created to establish connections among those interested in cycling. No matter how knowledgeable we are in a particular area, there is always something new to learn, room for development, and the opportunity to expand our knowledge. Questions can be publicly posed, where other users can respond and share their experiences related to the given topic. Additionally, considering the option to purchase a bicycle, the user has the ability to initiate private messaging.

It is worth mentioning that one of the most important aspects of cycling is the routes we take. Therefore, the app provides the possibility to save and share these routes with other users.

It is not uncommon for people to share pictures of things they are proud of on other social platforms. BicyHub users can do the same, as they can share their own bike collections in an environment where people can truly appreciate their possessions and the subtle details that only a bike enthusiast would notice.

Keywords: application, social platform, community, hobby, bicycle purchase

Tartalomjegyzék

1. Bevezető	11
1.1. Célkitűzések	11
1.2. Hasonló alkalmazások	13
1.2.1. CycleStreets	13
1.2.2. Map My Ride	14
1.2.3. Cyclemeter	14
1.2.4. Fill That Hole	15
2. Szoftver technológiák bemutatása	17
2.1. React	17
2.2. Firebase	18
2.2.1. Firebase Authentication	18
2.2.2. Firestore Database	19
2.2.3. Firebase Storage	20
2.3. Gluestack	21
2.4. TypeScript	21
2.5. Zod	22
2.6. Zustand	23
2.7. Git verziókövető	24
2.8. NPM - Node Package Manager	24
2.9. Kódbazis minőségét javító könyvtárak	24
2.9.1. Prettier	24
2.9.2. ESLint	25
2.10. EAS - Expo Application Service	26
3. Felhasználói követelmények	27
3.1. Nem funkcionális követelmények	27
3.2. Funkcionális követelmények	27
3.2.1. Vendég mód	29
3.2.2. Bejelentkezett mód	29
4. Az alkalmazás bemutatása	30
4.1. Vendég felhasználó	30
4.2. Bejelentkezett felhasználó	33
Összefoglaló	45

Ábrák jegyzéke **46**

Irodalomjegyzék **47**

1. fejezet

Bevezető

Teljesen természetes, hogy valaki olyan társaságra vágyik, akikkel könnyedén megérzik egymást, amikor olyan témáról beszél, ami érdekli, de talán nem mindig van lehetősége részletekbe merülni. Az alkalmazás célja, hogy egy segítőkész bicikli közösséget hozzon létre, ahol a felhasználóknak lehetőségük van kialakítaniuk kapcsolatokat.

Minden egyes hobbihoz tartozik egy nagy társaság akik élvezik úzni azt a bizonyos tevékenységet, a biciklizés egyik a sok közül. A BicyHub projektnek a lényege, hogy az embereket összehozza és új barátságokat alakítson ki. Ezen kívül az emberek segítségét kérve, kihasználva az elérhetőséget amit az alkalmazás nyújt, útmutatóként is szolgálhat egy olyan személynek aki most fedez fel a biciklizést.

Míg ezek lennének e telefonos alkalmazásnak a főbb céljai, más célokra is szolgál. Például, bejelentkezve meg lehet tekinteni más felhasználóknak a profiljait, ahol rá lehet pillantani a tulajdonukban lévő biciklikre. Természetes módon megtörténhet, hogy egy bizonyos kerékpárt megkedvel a felhasználó és meg szeretné vásárolni. Erre is megoldást nyújt a BicyHub, közös megegyezés alapján, a tulajdonos megossza a kulcsot a felhasználóval, és annak a beillesztésével a biciklit a profiljának a tulajdonában fogja megtalálni.

Az alkalmazás segítségével privát üzeneteket is küldhetnek a felhasználók egymásnak, hogy részletesebben tudjanak tárgyalni egy adott témáról, ha úgy érzik, nem szeretnék nyilvánosan tárgyalni azokat.

GPS követővel is rendelkezik a BicyHub, amelynek fő célja egy túrázás útvonalának elmentése. Az elmentett útvonalakat ismét meg lehet tekinteni, vagy megosztani más felhasználókkal.

Összességében az alkalmazást egy bicikliszeretők számára kialakított társasági platformnak tekintethetjük, más néven a “Bicikli Facebook”-nak.

1.1. Célkitűzések

Az alkalmazás fejlesztésének kiindulópontja a nyári gyakorlaton történt részvétel volt, ahol a cél ennek a projektnek a megvalósítása volt. Felosztottak minket több fős csapatokra, viszont ennek a projektnek a fejlesztéséhez én és a szaktársam, Hegyeli András járultunk hozzá és voltak mentoraink akiknek a tapasztalatuknak és segítségüknek hála gyorsabban tudtunk haladni a projekt kódolásával.

A projekt folyékony munkamenetének érdekében az első két hetet egy „dummy-project” elkészítésével töltöttük el amit react-native és typescript-ben írtunk meg. Ennek

a projektnek a célja az volt, hogy gyakoroljuk ezeknek a programozói nyelvezeteknek a használatát, hogy csökkentsük az akadályok számát a fő projekt megírásakor. Az Agile ágból származó Scrum munkamenetet követtük a csapatban belül, mely hozzájárul a munkamenet tisztázásához, és egyértelművé tette, hogy pontosan ki milyen feladatokért felelős. minden nap 11 órakor volt egy daily meeting-ünk ahol elmagyaráztuk mit sikerült megoldani és milyen elakadásaink voltak. Két hetes sprintekbe dolgoztunk és mindegyik két héten tartottunk egy hosszabb gyűlést, aminek a neve sprint retrospektíva volt, ahol megbeszélük, hogy mit tapasztaltunk az előző munkamenetben és kritikákat hoztunk fel annak érdekében, hogy a következőkben jobb végeredmények szülessenek.

A dolgozat elkészítése során számos kitűzött célt sikerült elérnünk. Ugyan csapatban dolgoztunk, mindegyikünk részt vett olyan feladatoknak a megoldásában a projektben, amelyekért egyéni felelősséget vállaltunk.

Legelső célunk a kereteknek a létrehozása és a közöttük lévő navigáció megvalósítása volt, amit felhasználók bevezetése az alkalmazásba követett. Ezután szükség volt meghatározni a szerepkörököt, hogy egy nem bejelentkezett felhasználó(vendég) milyen funkciókat képes elvégezni egy bejelentkezett felhasználóval szemben. Ugyebár annak érdekében, hogy a szerepköröket valósítsuk meg, létre kellett hozzunk két külön képernyőt. Az egyik ilyen képernyő a bejelentkezés volt, amely a felhasználó azonosításáért felelős és a másik a regisztráció, amelynek a szerepe az új felhasználói fiók létrehozása.

Ezen célok elérése után, munkánk szétoszlott és a felhasználóknak a profil képernyő létrehozása az én feladatom lett, míg Andor a bicikliknek a részleteinek megjelenítését és szerkesztését a felhasználói felületen próbálta megvalósítani. Részletezve a feladatom a profil képernyőn, meg kellett jelenítsem a felhasználónak a személyes adatait(pl. név, telefonszám,..stb.) és meg kellett adjam a lehetőséget, hogy ezeket az adatokat legyen képes szerkeszteni is. Emellett meg kellett oldani, hogy a felhasználóknak a saját biciklijei és a másuktól beérkezett értékelések is szerepeljenek a képernyőn.

Következő céлом a kezdőlapnak a létrehozása volt. Ezen az oldalon, véletlenszerűen jelenítek meg különböző bicikliket a felhasználóknak és megadtam a lehetőséget, hogy bejegyzéseket tudjanak létrehozni, mások bejegyzéseit megtekinteni, kedvelni és hozzászólást írni.

Még szükség volt az alkalmazásnak a kitelepítésére találni egy megoldást és az automatikus bejelentkezés problémájának megoldására, hogy a felhasználó ne kelljen minden alkalommal újra jelentkezzen csak ha kijelentkezett mielőtt lezárta a programot.

A privát üzenetezés a felhasználók között pedig az én kezembé került, amivel sok változás járt, hiszen adatbázison kellett változtatni, profil képernyőt egy új funkcióval bővíteni, két teljesen új képernyőt hozzáadni, egyik a beszélgetések listájának a megjelenítésével foglalkozott, míg a másik a az üzenetekével, természetesen számon tartva, hogy melyik beszélgetést választotta ki az illető személy.

A következő fontosabb feladatom a GPS követőnek az implementálása volt, amihez sok új dolog járult hozzá, mint például a térképnek a megjelenítése, kezelése, a koordinátáknak az elmentése és azok értelmezése..stb. Egy teljesen új terület volt számomra, amit izgalmasnak tartottam felfedezni.

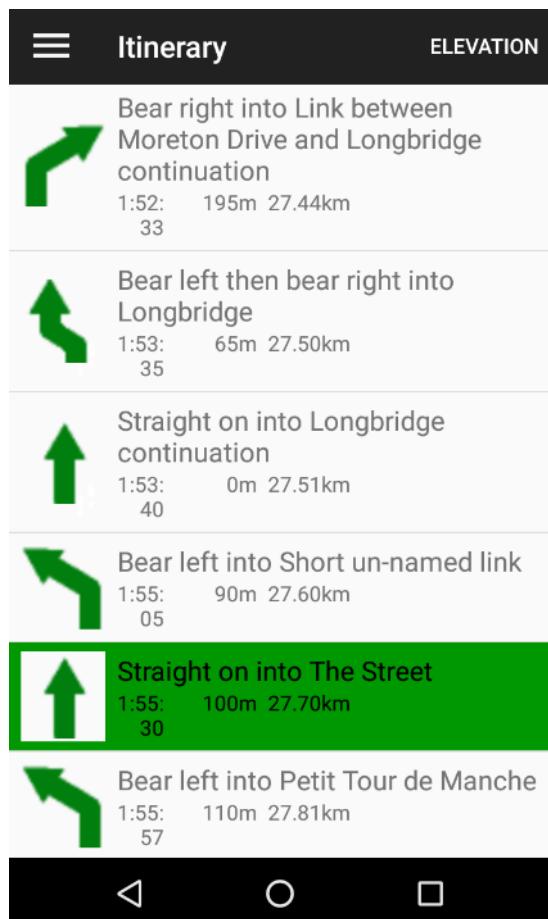
Felhasználók hitelesítésének érdekebén, kézügyben vettetem az email verifikáció implementálását és szükség esetén, lehetőséget adtam a felhasználóknak a jelszavuknak a megváltoztatására.

1.2. Hasonló alkalmazások

1.2.1. CycleStreets

A „CycleStreets” lehetőséget biztosít számodra, hogy különböző útvonalakat kérj a preferenciáidnak megfelelően(lásd 1.1 ábra). Például, ha csendesebb utat keresel mun-kába, akkor a CycleStreets kevésbé forgalmas, ugyanakkor biztonságos útvonalakat talál számodra.

Az alkalmazás számos hasznos információt nyújt, beleértve a sebességedet és megtett távolságodat. Egy élő térkép segítségével követheted az útjaidat és kaphatsz utasításokat, hogy minden pontosan láthasd, hol vagy éppen.



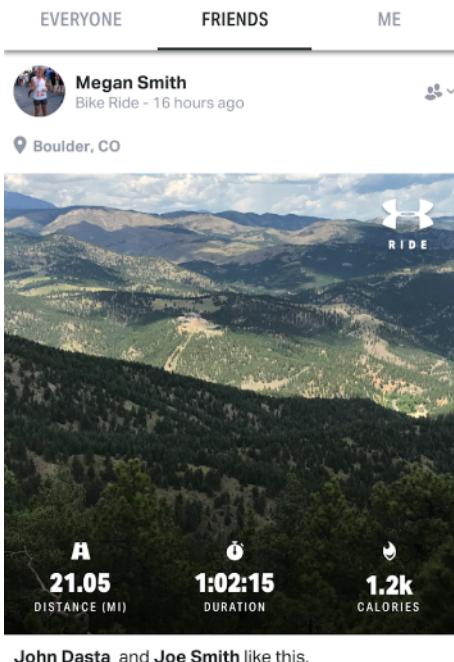
1.1. ábra. ”CycleStreets ” útvonal leírása

1.2.2. Map My Ride

A „Map My Ride” alkalmazásnak a célja, hogy ha unatkozol a minden nap útvonaladon, lehetőséget nyújtson, hogy egy újat fedezz fel. Amikor találsz egy igazán jó útvonalat, elmentheted és megosztathod barátaiddal, hogy ők is hozzáérhessenek.

Láthatod ismerőseid tevékenységeit az aktivitások listájában(lásd 1.2 ábra), csatlakozhatsz kihívásokhoz másokkal való versenyzéshez, követheted a ranglistát és nyerhetsz díjakat.

Egy másik hasznos funkció az audio visszajelzés, amely statisztikákat nyújt, például elégetett kalóriák és az utazásod sebessége tekintetében.



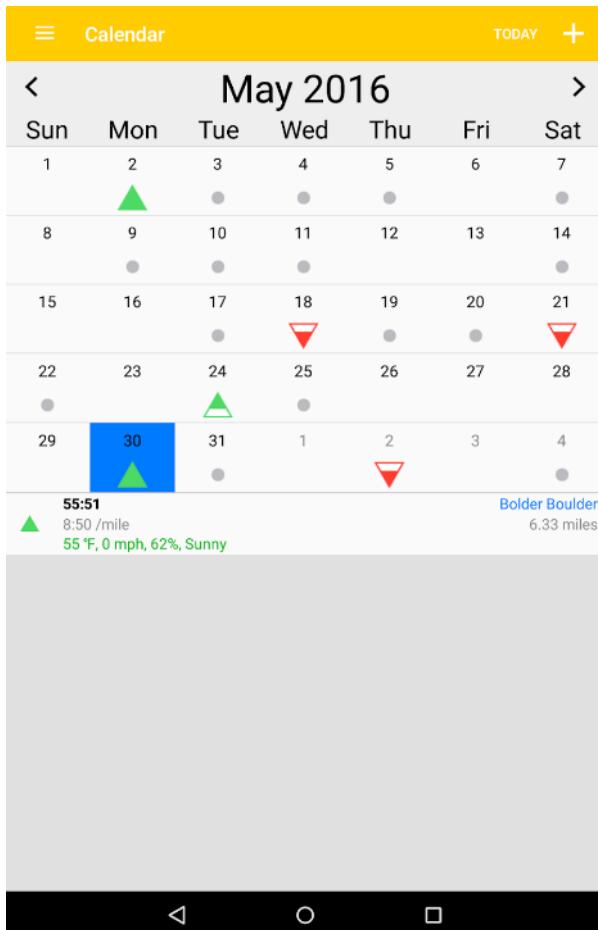
1.2. ábra. Barátok aktivitásának megjelenítése

1.2.3. Cyclemeter

A „Cyclemeter” alkalmazás tartalmaz egy naptárat, ahol nyomon követheted a kerékpározásaidat, és megtudhatod, mikor érdemes beütemezned egy edzést(lásd 1.3 ábra). Meg lehet tudni a segítségével, hogy milyen rendszerességgel bicikliztél, és megtudd az időt és távolságot.

A „Cyclemeter” bejelentésekkel rendelkezik, hogy motiváljon téged a bringázás közben, így megtarthatod a koncentrációdat, és úgy érezheted, mintha saját szurkolód lenne.

Továbbá, rengeteg ingyenes hasznos adatot is tartalmaz, mint például térképek, szakaszok, valamint oszlop- és kördiagramok, hogy összefoglalja a kerékpározási edzéseidet és biztosítja, hogy figyelemmel kísérd a fejlődésedet.

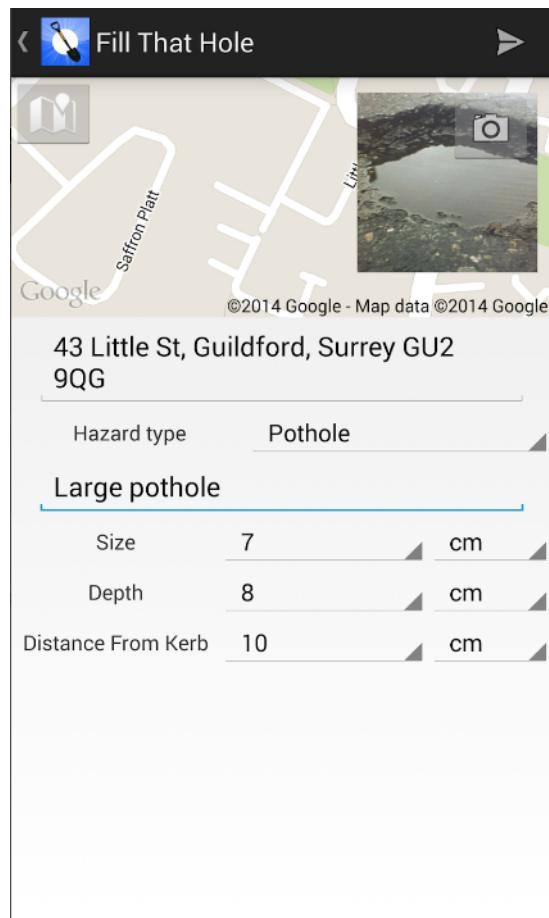


1.3. ábra. Naplár a feltöltött tevékenységekkel

1.2.4. Fill That Hole

A „Fill That Hole” alkalmazást a vezető brit kerékpáros jótékonysági szervezet, a „Cycling UK” üzemelteti. Ideális választás azoknak, akiket gyakran kátyúk és egyéb útburkolati problémák zavarják(lásd 1.4 ábra).

Ezek a problémák veszélyesek lehetnek, mivel a kátyúk gyakran vezetnek a kerékpárosok irányításának elvesztéséhez és az útra való kitéréshez. Az alkalmazás segítségével jelentheted ezeket a problémákat a helyi önkormányzatnak - egyszerűen töltse ki a formanyomtatványt, és a Fill That Hole elvégzi a többit.



1.4. ábra. Gödör jelzése a platformon

2. fejezet

Szoftver technológiák bemutatása

A BicyHub egy összetett alkalmazás, amely különböző könyvtárakat és technológiákat felhasználva lett kifejlesztve. A reszponzív felület megvalósításához a React könyvtár van alkalmazva, mely Typescript programozási nyelvben íródott, egy átlátható kód érdekében. A megbízhatóság érdekében a backend rész a Firebase szolgáltatás használatával lett implementálva. A felhasználói felület létrehozásához pedig a Gluestack könyvtár lett alkalmazva.

2.1. React

A React egy JavaScript könyvtár, amelyet a „Facebook” fejlesztett ki és a felhasználói felületek készítésére használható. Fő jellemzője a komponensalapú fejlesztés, amely lehetővé teszi a felhasználói felület logikájának elkülönítését és újrafelhasználhatóságát. A React segítségével könnyedén létre lehet hozni összetett, interaktív és reszponzív felhasználói felületeket [Gac15].

A React elsősorban webalkalmazások felületeinek kialakítására szolgál, viszont a projektben a felhasználói felülethez kapcsolódó komponensek megjelenítési logikájának kezelésében játszik szerepet. Az egyik leggyakrabban használt és rendkívül hatékony funkciója a „hook”-oknak nevezett elemek, amelyek lehetővé teszik a komponensek állapotainak kezelését és életciklus eseményeinek kezelését. Ezeknek a használata dramatikusan leegyszerűsítette a „React”-et a fejlesztőknek számára ok miatt. Az egyik az osztályok kiküszöbölése, amelyeknek az állapotok kezelése volt az elsődleges feladata. Egy másik példa az elemek elérése és hozzáférése oly módon, amely nem igényli a felsőbb rendű komponensek beavatkozását [Bug19]. Ezen belül kiemelkedő példa (2.1) a „useState hook”, amely segítségével könnyedén kezelhetjük a komponens állapotait:

```
const [date , setDate] = useState("");
const [distance , setDistance] = useState(0);
const [cost , setCost] = useState(0);
const [type , setType] = useState<logType>("ACTIVITY");
const [description , setDescription] = useState("");
```

2.1. kódrészlet. "useState hook" használata

Egy másik gyakran alkalmazott és hasznos funkció a “useEffect”, amely lehetővé teszi az aszinkron műveletek, mint például adatlekérések vagy más mellékhatások kezelését(2.2):

```
useEffect( () => {
  if (user && user.public.likes) {
    const likes = user.public.likes;
    likeCheck(likes);
  }
}, [user]);
```

2.2. kódrészlet. "useEffect hook" használata

2.2. Firebase

A Firebase egy olyan hozzáférhető és skálázható fejlesztői platform, amelyet a Google fejlesztett ki. Az elsődleges célja a fejlesztőknek a felhőalapú szolgáltatások és eszközök széles választékát biztosítani, így könnyen készíthetnek és üzemeltethetnek webes és mobilalkalmazásokat [CYGV21].

A következőkben be lesz mutatva, hogy mely Firebase szolgáltatások lettek implementálva az alkalmazásban.

2.2.1. Firebase Authentication

A Firebase Authentication e felhőalapú platform egyik szolgáltatása, amely segít a fejlesztőknek az alkalmazásuk felhasználónak azonosításában és autentikációjában. Ez a szolgáltatás lehetővé teszi a felhasználók regisztrációját, bejelentkezését és azonosítását, valamint az ezekhez kapcsolódó biztonsági funkciók kezelését [CCDN18].

Lehetőség van a Google, Facebook, Twitter, GitHub és más külső szolgáltatók segítségével történő bejelentkezésre és regisztrációra, de amint a képeken látható az email és jelszó autentikációs mód lett implementálva az alkalmazásban. A bejelentkezésnél az alábbi(2.3) metódus volt meghívva:

```
await signInWithEmailAndPassword(FIREBASE_AUTH , email , password);
```

2.3. kódrészlet. Email/jelszó alapú bejelentkezés függvény meghívása

Míg a regisztrációnál az alábbi(2.4):

```
await createUserWithEmailAndPassword(
  FIREBASE_AUTH,
  UserRegister.private.email,
  password
);
```

2.4. kódrészlet. Email/jelszó alapú regisztráció függvény meghívása

2.2.2. Firestore Database

A Cloud Firestore Database a Firebase platform egyik adatbázis szolgáltatása. Ez egy rugalmas, skálázható és valós idejű dokumentumalapú adatbázis, amely kiválóan alkalmas a modern alkalmazások fejlesztéséhez [KGT⁺²³].

A Firestore adatbázis struktúráját dokumentumok és gyűjtemények alkotják. A dokumentumok JSON-szerű objektumok, amelyek tartalmazhatnak kulcs-érték párokat, listákat vagy akár beágyazott al dokumentumokat is. A gyűjtemények csoportokként szolgálnak a dokumentumokhoz, és könnyen kezelhetők és lekérdezhetők.

A következő műveletek segítenek a dokumentumok létrehozásában, frissítésében és törlésében a Firestore adatbázisban.

1. A “setDoc()” művelet egy új dokumentum létrehozására vagy egy meglévő dokumentum teljes felülírására szolgál. Használata az alábbi(2.5) kód részletben megtekinthető:

```
await setDoc(  
    doc(FIRESTORE_DB, "users", UserRegister.private.id,),  
    UserRegister.private  
) ;
```

2.5. kód részlet. Új dokumentum létrehozatala

2. Az “updateDoc()” művelet egy meglévő dokumentum frissítésére szolgál az adatbázisban. Az adatokat összehasonlítja a meglévő dokumentummal, és csak a különbözőket frissíti(2.6).

```
await updateDoc(docRef , {  
    Reviews: updatedReviews,  
});
```

2.6. kód részlet. Létező dokumentum frissítése

3. A “deleteDoc()” művelet egy meglévő dokumentum törlésére szolgál(2.7).

```
await deleteDoc(doc(FIRESTORE_DB, "testBikes", bike.id));
```

2.7. kód részlet. Dokumentum törlése

Ugyebár az adatokhoz lévő hozzáférhetőséget biztonságossá kell tenni, erre a problémára a Firestore Security Rules a megoldás. Egyedisége abban rejlik, hogy a megírt szabályok a szerveroldalon végzik az ellenőrzést. Az alkalmazás kliensoldali kódja nem tudja megkerülni vagy módosítani ezeket a szabályokat, így a biztonság garantált. Az alábbi kód részleten(2.8) láthatóak az alkalmazásnak a biztonsági beállításai:

```

service cloud.firestore {
    match /databases/{database}/documents {
        match /{document=**} {
            allow read: if true ;
        }

        match /users/{userId} {
            allow create, read: if request.auth.uid != null ;
            allow update, delete: if request.auth.uid == resource.data.id;
        }

        match /users/{userId}/interactions/{interactionsId} {
            allow read, write: if request.auth.uid != null ;
        }

        match /testBikes/{bikeId} {
            allow write: if request.auth.uid == resource.data.ownerUid ;
            allow update: if request.resource.data.secretKey ==
                resource.data.secretKey ;
        }

        match /testPosts/{postId} {
            allow read, create: if request.auth.uid != null ;
            allow update, delete: if request.auth.uid == resource.data.ownerUid ;
        }

        match /testPosts/{postId}/interactions/{interactionsId} {
            allow read, write: if request.auth.uid != null ;
        }
    }
}

```

2.8. kódrészlet. A Firebase biztonsági beállításai

2.2.3. Firebase Storage

A Firebase Storage egy olyan felhőalapú tárhelyszolgáltatás, amelyet a Firebase platform kínál. Ez a szolgáltatás a Firestore adatbázissal együttműködve lehetővé teszi a fejlesztők számára a fájlok (képek, videók...stb.) tárolását és kezelését a Google felhő infrastruktúrájában. minden egyes eltárolt fájlnak lesz egy elérhetőségi linkje, amit az adatbázisban mentünk el [MM17]. Működése megtekinthető az alábbi kódrészleten(2.9).

```

const fetchResponse = await fetch(imageToUpload);

const theBlob = await fetchResponse.blob();

```

```

const imageRef = ref(FIRESTORE_STORAGE,
    'ProfilePics/${user.private.username}');

getDownloadURL(imageRef)
    .then(() => {
        deleteObject(imageRef);
    })
    .catch(() => {
        console.info("Nothing to delete");
    });
}

const uploadTask = uploadBytesResumable(imageRef, theBlob);

```

2.9. kódrészlet. Kép feltöltése a felhőtárhelyben

2.3. Gluestack

A „gluestack-ui” egy univerzális felhasználói felület könyvtár, amely tematizált és tematizálatlan komponenseket biztosít. Ezeket a komponenseket könnyen lehet integrálni olyan alkalmazásokba, amelyeket React, React Native, Next.js és Expo segítségével fejlesztettek.

Az alkalmazás legelső verziójában a Native-Base könyvtárhoz tartozó komponenseket használtuk. Azonban az idő elteltével a GlueStack megjelent, mely egy frissített és továbbfejlesztett változata volt a Native-Base-nek. Ennek következtében a projektnek a felhasználói felületi rétege sikeresen át lett állítva GlueStack-re.

2.4. TypeScript

A TypeScript egy olyan nyelv, amely a JavaScript fejlesztők számára készült, de kiterjeszti azt a statikus típusellenőrzés és az objektumorientált funkciók bevezetésével. Ez a Microsoft által fejlesztett nyelv egy szuperszettje a JavaScript-nek, ami azt jelenti, hogy minden érvényes JavaScript kód érvényes TypeScript kód is [JVDW16]. A fő különbség a két nyelv között a típusrendszer bevezetése(lásd az alábbi kódrészletben 2.10).

```

async function editImages(
    newBikePictures:string[],
    copyImages: string[],
):Promise<string[]>

```

2.10. kódrészlet. TypeScript típusozása

A BicyHub teljes mértékben kihasználta a TypeScript nyújtotta előnyöket. Az előre beépített statikus típusellenőrzés lehetővé teszi a fejlesztők számára, hogy már a kódírás folyamán észrevegyék és javítsák a típushibákat, ami exponenciálisan csökkenti a futási idejű hibák kockázatát és biztosítja a kód stabilitását. Emellett a TypeScript segíti a fejlesztőket a kódintelligencia terén, kényelmesebbé téve a kódkiegészítéseket és hozzájárulva a fejlesztés hatékonyságához.

Az intelligens típusok nemcsak a kódminőséget javítják, hanem megkönnyítik a kód értelmezését is. Ezáltal a fejlesztők könnyebben karbantarthatják a kódot, mivel a típusok tisztázottabbá teszik a változók és függvények várható viselkedését. Együttesen minden hozzájárul a projekt hosszú távú sikéréhez és karbantarthatóságához.

2.5. Zod

A Zod egy TypeScript alapú séma deklarációs és ellenőrző könyvtár, amely segít a fejlesztőknek definiálni és validálni különböző adattípusokat TypeScript kódban, ami ugyebár erre a projektre teljes mértékben jellemző. A könyvtár „schema” koncepcióját tágán értelmezi, magában foglalva a különféle adattípusokat, legyen az egyszerű string, szám, vagy akár összetett, egymásba ágyazott objektumok [Kna].

A Zod működésének megvalósításának érdekében, elsősorban létre kell hozni a típusbiztos séma deklarációt(lásd az alábbi kódrészletben 2.11), ahol a fejlesztők könnyen és olvashatóan definiálhatnak adattípusokat, és a Zod automatikusan generálja a típusinformációkat.

```
export const BikeSchema = z.object({
  id: z.string(),
  owner: z.string(),
  ownerUid: z.string(),
  name: z.string().min(3),
  type: z.enum(["ASPHALT", "CROSS", "TANDEM"]),
  pictures: z.array(z.string()),
  modelYear: z
    .number()
    .lt(new Date().getFullYear() + 1)
    .gt(1900),
  components: z.record(z.string().min(3), z.string().min(3)),
  description: z.string().min(1),
  prevOwners: z.array(z.string().min(1)).optional(),
  value: z.number().optional(),
  secretKey: z.string(),
  log: z
    .array(
      z.object({
        id: z.string(),
        type: z.enum(["SERVICE", "CRASH", "ACTIVITY", "CHANGE", "UPGRADE"]),
        date: z.string(),
        description: z.string().min(1),
        pics: z.array(z.string()).optional(),
        distance: z.number(),
        cost: z.number().optional(),
      })
    )
    .optional(),
});
```

2.11. kódrészlet. Zod séma felépítése

Az ezután következő, hátramaradt lépés a validáció(lásd [2.12 kódrészlet](#)), ahol a Zod a megadott objektumot validálja a séma deklarációval összehasonlítva, így biztosítva az adatok konzisztenciáját és érvényességét.

```
BikeSchema.parse(newBike);
```

2.12. kódrészlet. Objektum hitelesítése Zod-dal

2.6. Zustand

A Zustand egy könnyen használható, de erőteljes React állapotkezelő könyvtár, amely ideális választás olyan esetben, ahol egyszerűségre és hatékonyságra van szükség az állapotok kezelésében. A könyvtár széleskörű funkciókat kínál, beleértve a hook-alapú megközelítést, ami természetesen illeszkedik a React fejlesztési modellezéshez. Minimalista API-jának köszönhetően gyorsan elsajtítható és kevesebb sablonkód szükséges az állapotok kezeléséhez [[Le23](#)].

Felépítése meglehetősen átlátható(lásd az alábbi kódrészletben [2.13](#)) és lehetővé teszi a könnyű használatot és rugalmasságot. Hasonló funkcionalitásokkal rendelkezett a React-nek a „useContext()” hook-ja, de a Zustand használata mellett döntöttünk figyelembe véve, hogy globális szinten elérhetőek az így deklarált változók.

```
import { create } from "zustand";
import { User } from "../types/DataTypes";

type UserState = {
  user: User | null;
  setUser: (newUser: User | null) => void;
};

export const useUserStore = create<UserState>((set) => ({
  user: null,
  setUser: (newUser: User | null) => set({ user: newUser }),
}));
```

2.13. kódrészlet. Zustand állapotkezelő felépítése

A Zustand állapotokhoz való hozzáférést hook-ok biztosítják(lásd az alábbi kódrészletben), ahol a „useUserStore()” hook az előbb létrehozott állapotmenedzsert téríti vissza.

A kiemelkedő tulajdonságai között szerepel az egyszerű globális elérhetőség, amely lehetővé teszi az állapotok egyszerű kezelését és frissítését bármely fájlban az alkalmazásban. Ezáltal könnyen integrálható és globálisan használható az alkalmazásban.

2.7. Git verziókövető

A projektmenedzselés és az adatok elvesztésének elkerülése érdekében a „Git” verziókövetőt használtuk. A „Git” egy ingyenes és nyílt forráskódú elosztott verziókezelő rendszer, amelynek fő célja, hogy minden fejlesztőnek adjon egy teljes privát példányt a szoftvertárról, és számos módot biztosítson a verziók kezeléséhez a kontextuson belül [Spi12].

2.8. NPM - Node Package Manager

Az „NPM” (Node Package Manager) egy ingyenes, nyílt forráskódú csomagkezelő szoftver, amelyet a JavaScript nyelvű alkalmazások fejlesztéséhez használnak. A Node.js nevű szerveroldali JavaScript futtatókörnyezet részeként érhető el, és lehetővé teszi fejlesztők számára különböző csomagok egyszerű telepítését, frissítését és kezelését [Ali13].

2.9. Kódbázis minőségét javító könyvtárak

2.9.1. Prettier

A Prettier egy kódformázó, amely a fejlesztő beállításai szerint egy konzisztens kódstílussal hat a projektre, amelynek fő célja az átláthatóság és a csapatban történő munka megkönnyítése [SMP21]. minden egyes fejlesztő rendelkezik egy sajátos kód írási stílussal, habár mikor sor kerül arra, hogy egy csapattárs értelmezze azt a bizonyos programrész, akkor előfordulhatnak nehézségek. Ebből az okból kifolyólag választottuk ezt a technológiát, amely egy konstans stílusú kódbázist nyújt számunkra. Az alábbi(2.14) beállítások a kódformázót valósításák meg.

```
{  
  "arrowParens": "always",  
  "bracketSpacing": true,  
  "jsxBracketSameLine": false,  
  "jsxSingleQuote": true,  
  "quoteProps": "as-needed",  
  "singleQuote": true,  
  "semi": true,  
  "printWidth": 100,  
  "useTabs": false,  
  "tabWidth": 2,  
  "trailingComma": "es5"  
}
```

2.14. kódrészlet. Prettier konfigurálása

2.9.2. ESLint

Az ESLint egy népszerű JavaScript kódelemző eszköz, amely segít a fejlesztőknek a kód minőségének javításában és a stílus útmutatók betartásában. Az ESLint használata lehetővé teszi a fejlesztők számára, hogy meghatározzák a kód bázisuk stílusát és azonosítsák a potenciális hibákat vagy stílusproblémákat a kódjukban. Ezáltal hozzájárul a kód minőségének javításához, a konzisztens kódstílus fenntartásához és az átláthatóbb, könnyebben karbantartható kód elkészítéséhez [Tri20]. Az alábbi(2.15) programrész az ESLint konfigurálásában játszik szerepet.

```
{  
  "env": {  
    "browser": true,  
    "es2021": true  
  },  
  "extends": [  
    "standard-with-typescript",  
    "plugin:react/recommended",  
    "standard-with-typescript/hooks",  
    "prettier" ,  
    "prettier/react"  
  ],  
  "parserOptions": {  
    "ecmaVersion": "latest",  
    "sourceType": "module"  
  },  
  "plugins": [  
    "react" ,  
    "react-native"  
  ],  
  "rules": {  
  }  
}
```

2.15. kódrészlet. ESLint konfiguráció

2.10. EAS - Expo Application Service

Az Expo Application Services egy fejlett eszközkészlet és felhőalapú platform, amelyet az Expo kínál a React Native alapú mobilalkalmazások gyorsabb és hatékonyabb fejlesztéséhez. Az EAS a fejlesztőknek lehetőséget biztosít a folyamatos integrációra, tesztelésre és kiadásra, valamint számos más fejlesztői eszközre és szolgáltatásra, amelyek elősegítik az alkalmazások fejlesztését és karbantartását [GOC]. Az alábbi (2.16) kódrészletben meg lehet tekinteni, hogy milyen beállítások léteznek és szükségesek az alkalmazásnak a kitelepítéséhez.

```
{  
  "cli": {  
    "version": ">= 7.2.0"  
  },  
  "build": {  
    "preview": {  
      "android": {  
        "buildType": "apk"  
      }  
    },  
    "preview2": {  
      "android": {  
        "gradleCommand": ":app:assembleRelease"  
      }  
    },  
    "preview3": {  
      "developmentClient": true,  
      "distribution": "internal"  
    },  
    "preview4": {  
      "distribution": "internal"  
    },  
    "production": {}  
  }  
}
```

2.16. kódrészlet. Kitelepítéshez szükséges konfigurációs fájl részletei

3. fejezet

Felhasználói követelmények

Ebben a fejezetben a felhasználói követelményeket fogom részletezni, annak érdekében, hogy egy jobb rálátást lehessen nyerni az alkalmazásnak a kapacitásával kapcsolatban.

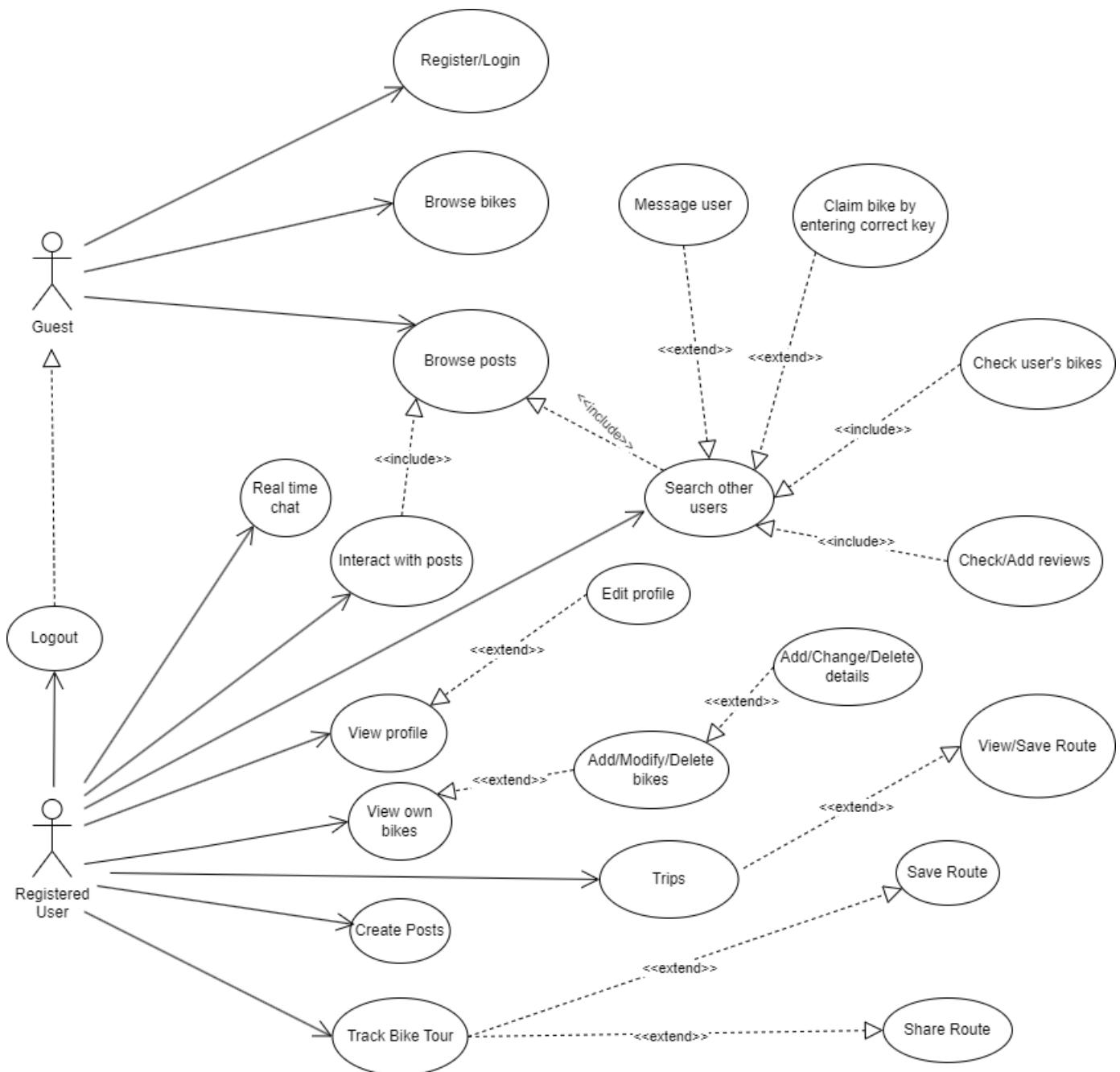
3.1. Nem funkcionális követelmények

Mivel „react native” technológiát használtunk, az alkalmazás kompatibilis, úgy „Android”, mint „iOS” eszközökkel. Az expo legújabb verziójának használata miatt, az egyetlen kikötések, hogy:

- az „Android” verzió legyen minimum 6-os
- az „iOS” verzió legyen minimum 13.4-es
- minimum 2 GB RAM

3.2. Funkcionális követelmények

A következőkben el fogom magyarázni, hogy milyen funkcionális követelményeket tartalmaz az alkalmazás. Mivel egy szociális platformról beszélünk, van egy vendég, meg egy bejelentkezett nézete az applikációnak. Az alábbi (3.1) diagram ábrázolni fogja a funkcionalitásokat rang szerint.



3.1. ábra. Use-Case diagram

3.2.1. Vendég mód

A vendég módnak a lényege, hogy egy kis ízelítőt kínáljon a felhasználónak, eltávolítva őt bármilyen interakciótól.

Ez pontosabban azt jelenti, hogy a felhasználó meg tudja tekinteni a főoldalon megjelenő bicikliket és bejegyzésekét. Továbbá, hozzáférést kap a bicikliknek a mélyebb részleteihez és a bejegyzésekkel megjelenő hozzászólásokhoz is.

Ezeket a funkciókat félretéve, a felhasználónak két módszere van a bejelentkezett módra áttérni: bejelentkezik vagy regisztrál. Regisztrálás esetén, az alkalmazás küld egy emailt az adott fiókra, amiben egy hivatkozás szerepel, amelynek célja a megadott email címnek a hitelesítése.

3.2.2. Bejelentkezett mód

Első sorban, a bejelentkezett felhasználónak elérése van azokhoz a tevékenységekhez, mint a vendég módban lévőnek, azzal a különbséggel, hogy be tudja kedvelni a bejegyzéseket, hozzászólásokat tud írni hozzájuk és ő is képes létre hozni őket.

Annak érdekében, hogy a felhasználó ne zárja ki magát a saját fiókjából, bele van építve az alkalmazásba egy jelszó visszaállító funkció, ami annyiból áll, hogy a felhasználó kap egy emailt egy hivatkozással, ami egy olyan weboldalra irányítsa, ahol meg tudja változtatni a jelszavát.

Továbbá, a regisztrálásnak hála, a felhasználó meg tudja tekinteni a saját profilját, ahol képes megváltoztatni a profil képét, a saját adatait megszerkeszteni és meg tudja nézni a róla írt értékeléseket másuktól.

Az ebben a kategóriában lévő emberek tudnak élni a baráti rendszer lehetőségével, ahol el kell küldeni egy kérést az illetőnek és annak függvényében, hogy elfogadja a kérést, hivatalosan barátok lesznek a platformon.

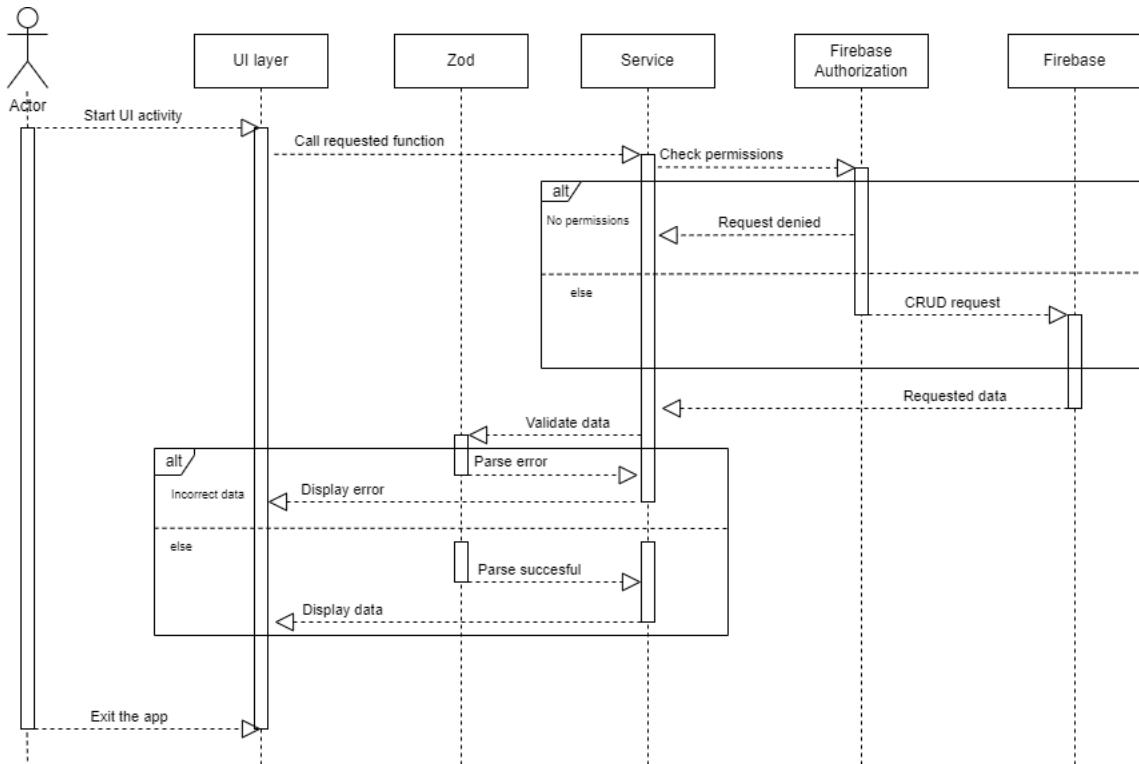
Ezek mellett, a felhasználónak hozzáférése lett a térképhez, amely segítségével tudja használni az útkövetőt, vagy meg tudja tekinteni az elmentett útvonalakat a térképen ábrázolva. Ezeket az útvonalakat meg lehet osztani mindenkinél vagy egy zártabb körnek, pontosabban a felhasználónak a barátaival. Ez ugyebár, egy kétirányú út, mivel másoknak is meg lehet tekinteni a túrázás részleteit.

A másokkal való interakciókhöz hozzátartozik a privát üzenetváltás, ahol diszkrétebb témákat lehet átbeszélni, beleértve a bicikli vásárlást, amihez szükséges az illetőhöz tartozó titkos kód.

4. fejezet

Az alkalmazás bemutatása

Ebben a fejezetben az alkalmazás használati utasításairól fogok több részletet megosztani, a könnyebb és gördülékenyebb felhasználói tapasztalat érdekében. Az implementálás, meg a háttérben történő folyamatok az alábbi diagramon (4.1) megtekinthetőek.

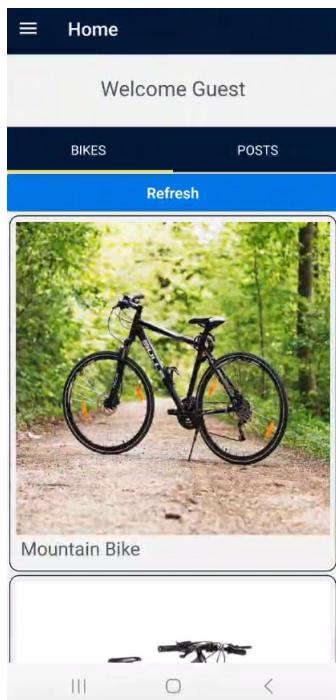


4.1. ábra. Szekvenciális diagram

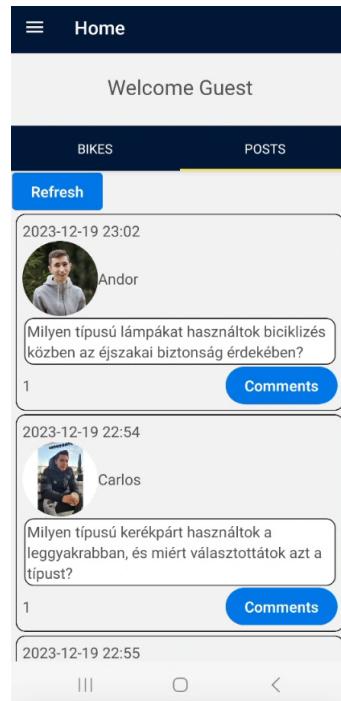
4.1. Vendég felhasználó

Abban az esetben, ha a felhasználó nincsen bejelentkezve, akkor a vendég szerepkört tölti be az alkalmazás keretén belül. Első megnyitáskor, a főoldal fog megjelenni (lásd 4.2 ábra), ahol egy üdvözlő szöveg, és egy lap nézet tárul elénk. Az első nézetben, különböző és véletlenszerűen szemléltetett bicikliket tudunk megtekinteni, folytonos görgető

gesztus segítségével. A második nézetben(lásd 4.3 ábra), dátum szerint, csökkenő sorrendben elrendezett bejegyzéseket és az azokhoz tartozó hozzászólásokat tudjuk megfigyelni, interakció nélkül természetesen.



4.2. ábra. Főoldal első nézete



4.3. ábra. Főoldal második nézete

A bal felső sarokban lévő ikonra kattintva, megjelennek azok a képernyőknek a nevei, amelyek között lehet navigálni(lásd 4.4 ábra). Rákattintva, a képernyő váltás megtörténik.



4.4. ábra. Vendég felhasználó navigációs lehetőségei

Ezt követően két lehetősége van a felhasználónak, az egyik, hogy bejelentkezzen(lásd [4.5](#) ábra), vagy ha nincs fiókja, regisztrálás(lásd [4.6](#) ábra) után, kap egy visszaigazoló email-t és utána jelentkezik be.

The screenshot shows a mobile-style login interface. At the top, it says "Login". Below that is a "Email:" field containing "email...". Below the email field is a "Password:" field containing "password...". At the bottom is a large blue "Login" button.

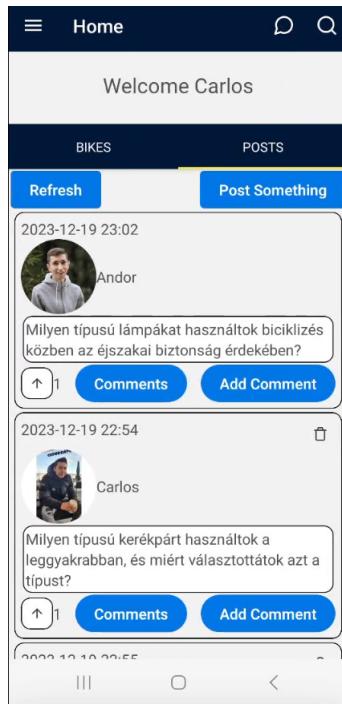
4.5. ábra. Bejelentkezési oldal

The screenshot shows a mobile-style registration interface. At the top, it says "Register". Below that is a "First Name:" field with "ex: John (min. 2)". Below it is a "Last Name:" field with "ex: Doe (min. 2)". Then there are fields for "Country:", "City:", "Username:", "Email:", "Password:", "Confirm Password:", and "Phone Number:". Each of these fields has an example value in parentheses. At the bottom is a large blue "Register" button.

4.6. ábra. Regisztrálási oldal

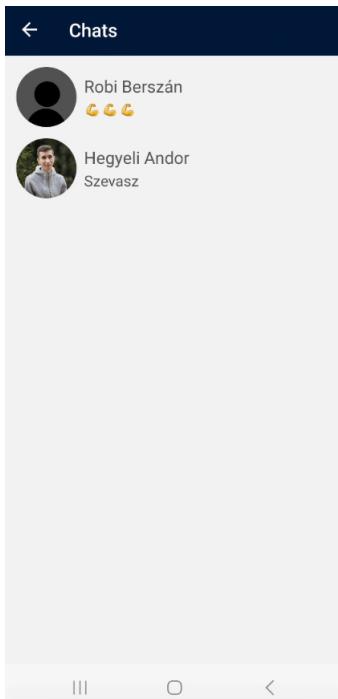
4.2. Bejelentkezett felhasználó

Amint láthatjuk, ([4.7](#) ábra) bejelentkezve, az üdvözlő szöveg a regisztrált nevén köszönti a felhasználót, megjelenít két ikont a jobb felső sarokban és még van a lehetőségünk, hogy bejegyzéseket hozzunk létre, like-olunk vagy hozzászólásokat írunk.

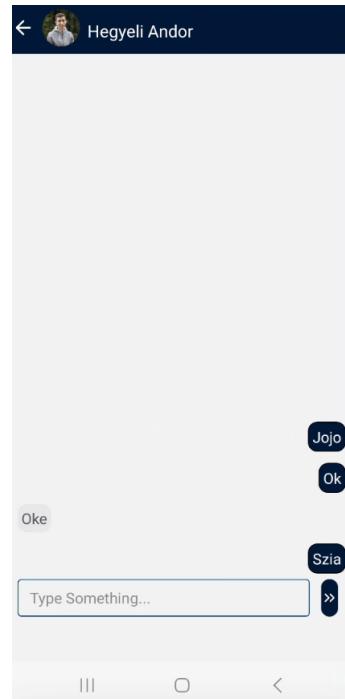


4.7. ábra. Főoldal bejelentkezett felhasználó esetén

A következőkben, térjünk ki az ikonok jelentőségeire. A jobb szélén lévő jelkép más felhasználóknak a keresésére szolgál. A mellette lévőnek a ránkattintásával átnavigálunk a „Chat” képernyőre(lásd 4.8 ábra), ahol bárkivel privát beszélgetést(lásd 4.9 ábra) tudunk folytatni.

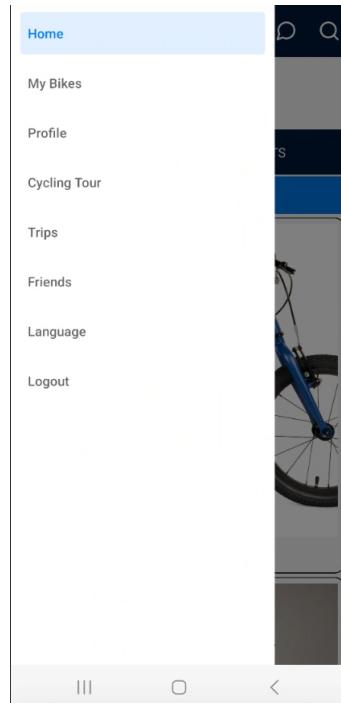


4.8. ábra. Üzenetlista



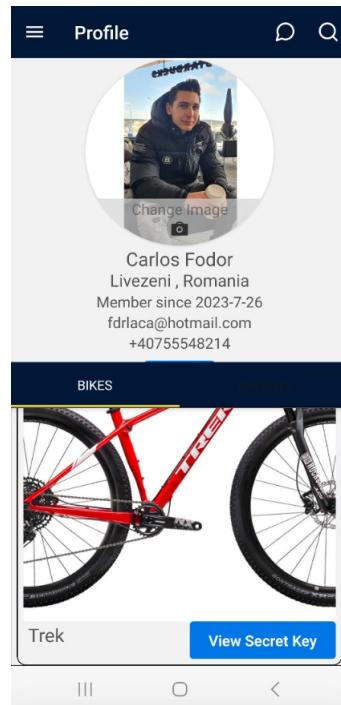
4.9. ábra. Üzenetezés

Mivel említettem az előző alfejezetben, a bal felső sarokban lévő jelképnek már ismerjük a célját, viszont észre lehet venni, hogy bejelentkezve több képernyőhöz van hozzáférésünk (lásd 4.10 ábra).



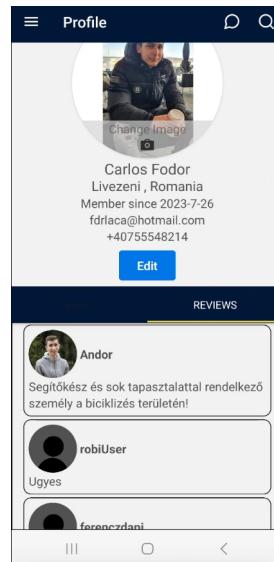
4.10. ábra. Bejelentkezett felhasználónak a navigáció lehetőségei

A következő képernyő, amiről szó fog kerülni a „Profile” (lásd 4.11 ábra), ami nevéből érthetően, egy felhasználónak a saját profilját szemlélteti. Az első dolog ami a szemnek fel tűnik a nagy, kör alakú kép lenne, ami a felhasználónak a profilképe. Az alatta lévő szövegek az illető személynek a személyes adatai, amelyeket megadott a regisztrálásnál. Ismét láthatunk egy lap nézetet, ahol, az első nézetben, meg lehet tekinteni az illetőnek a biciklijeit. Mindegyik biciklit meg lehet nézni lefele görgetve és rájuk kattintva átnavigál egy olyan oldalra, amely a biciklinek a technikai részleteit tartalmazza. Emellett észrevehető a képek mellett egy gomb, „View Secret Key” néven. A gombnak a megnyomásával megjelenik egy kis ablak ami tartalmaz egy titkos kulcsot, amihez csak a bicikli tulajdonosának van hozzáférése. Ha nem a saját profilunkon böngészünk, akkor egy másik gombhoz lesz hozzáférésünk „Claim bike” néven. Abban az esetben, ha egy bicikli vásárlás történt a háttérben, az eladó megossza a titkos kulcsot a vevővel, amit(a „Claim bike” gomb megnyomása után) beír a megfelelő szövegmezőbe és helyes kulcs beírása esetén, az illető lesz a kerékpárnak az új tulajdonosa az alkalmazás felületén is.



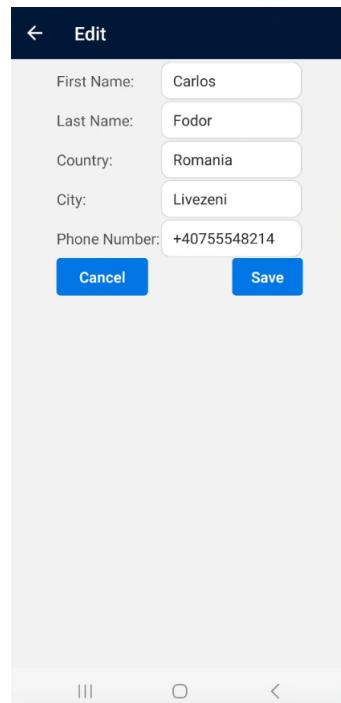
4.11. ábra. Profil oldal első nézete

A második nézetben, felbukkanak a más felhasználók által megírt kiértékelések a hitelesség érdekében(lásd 4.12 ábra). Ugyebár, bárkinek lehetnek kérdései a biciklizés területén és az alkalmazás célja, hogy egy megbízható közösséget kínáljon mindenkinnek. Emiatt, ezek a kiértékelések egy mérceként szolgálnak, hogy bárkinek legyen egy kis felfogása, hogy mennyire bízhat meg az illetőben, aki választ adott a kérdésére.



4.12. ábra. Profil oldal második nézete

A felhasználó adatai alatt szereplő „Edit” gomb egy olyan oldalra vezet, amely az illető adatok megszerkesztéséért felelős(lásd 4.13 ábra).



4.13. ábra. Profil szerkesztését szolgáló oldal

A következő képernyő, amit be fogok mutatni a „Cycling Tour” lesz (lásd 4.14 ábra). Az első doleg amit észre fogunk venni egy térkép lesz, ami elfoglalja a képernyő nagy részét és fölötté két gombot fogunk találni.



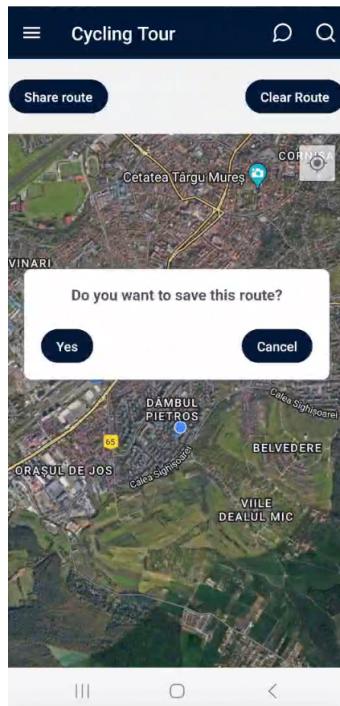
4.14. ábra. Cycling Tour képernyő

A „Start tracking” gombnak a célja az, hogy tárolja el az útvonalat, ami mentén biciklizett a felhasználó. Miután meg lett nyomva ez a gomb, az alkalmazás a háttérben megjegyzi a koordinátákat minden irányváltásnál és megjelenít egy új gombot „Stop tracking” néven, amivel meg lehet állítani az útvonal követőt(lásd 4.15 ábra).

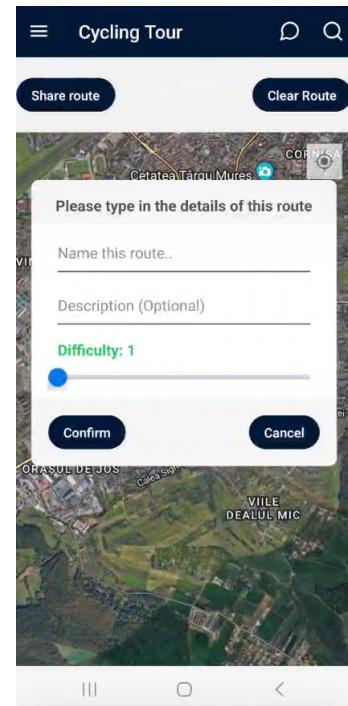


4.15. ábra. Útvonal követő elindítva

Miután meg lett állítva a követő, meg fog jelenni egy kis ablak amely megkérdezi, hogy el akarjuk-e menteni a lekövetett útvonalat (lásd 4.16 ábra). Ha a „Yes” opción választottuk, elő fog ugrani még egy ablak, amelynek a túrának a részleteire lesz szüksége. Meg fog kelleni adjunk egy nevet, ami szerint fogja tudni azonosítani az alkalmazás az útvonalat, a nehézségi szintjét az adott biciklitúrának egy 1-től 10-es skálán és akár egy leírást, ami opcionális (lásd 4.17 ábra).

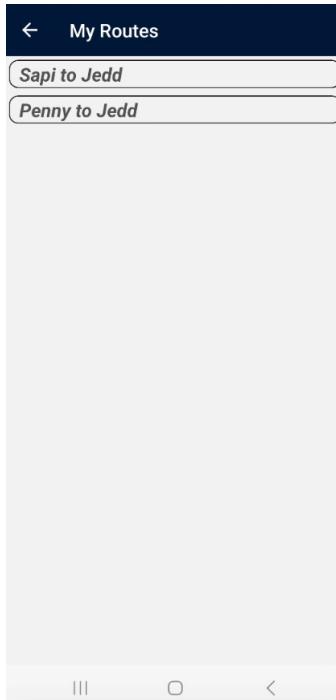


4.16. ábra. Útvonal elmentésének lehetősége



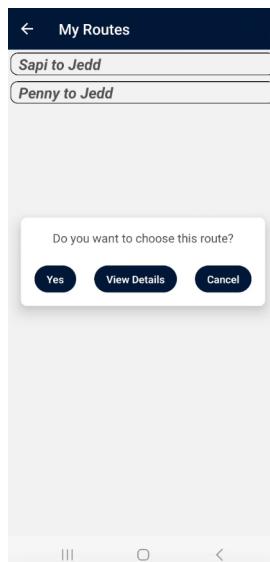
4.17. ábra. Túra részleteinek elmentése

Ha sikeresen elmentettük az útirányt, a „View Routes” gombon keresztül el tudjuk érni.(lásd 4.18 ábra)



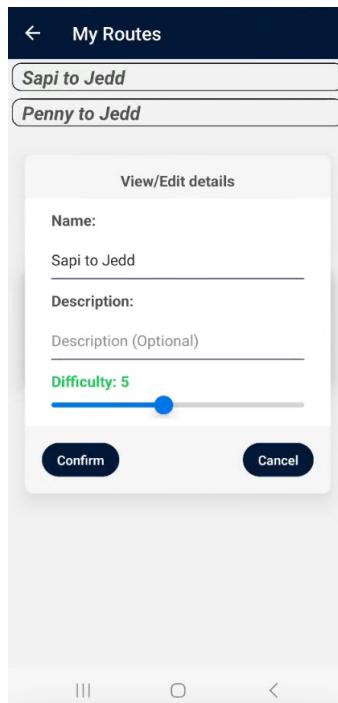
4.18. ábra. Elmentett útvonalak

Ha kiválasszuk az egyiket, ismét megjelenik egy ablak, amely megkérdezi, ha meg akarod tekinteni a térképen a biciklitúrát vagy csak a részleteire vagy kíváncsi(lásd 4.19 ábra). Hosszú gomblenyomással ki lehet törölni a kiválasztott entitást.



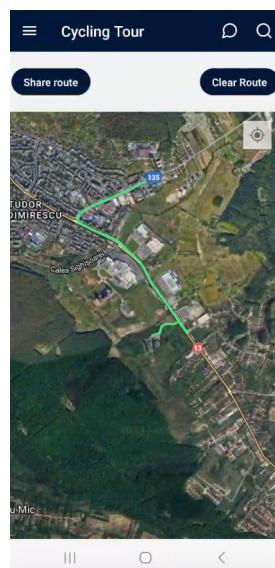
4.19. ábra. Választás a térképen való megjelenítés és a részletek megtekintése között

Ha a középső gombot válasszuk ki, meg lehet figyelni az adott részleteket, de egyúttal, meg van a lehetőség azok módosítására is(lásd 4.20 ábra).



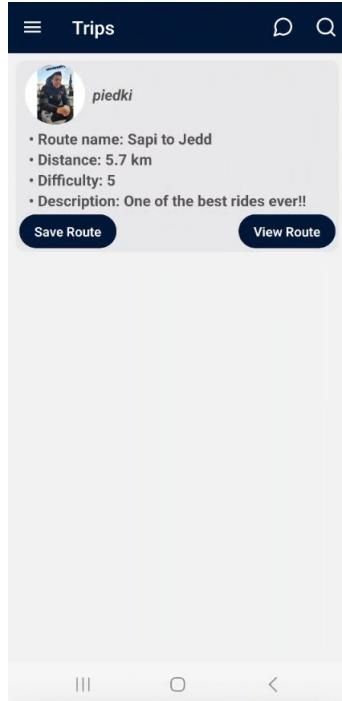
4.20. ábra. Biciklitúra részletek

Ha a „Yes” gombot nyomtuk meg, akkor az alkalmazás visszavezet a térképhez és egy zöld vonallal ábrázolja a kiválasztott útvonalat(lásd 4.21 ábra). Ez a funkció azért lett implementálva, hogy ne vesszen kárba egy olyan biciklitúra, ami a felhasználónak nagyon tetszett, azonban egy útmutató hiányában, nem tudná újraélni az egészet.



4.21. ábra. Biciklitúra koordináták térképen

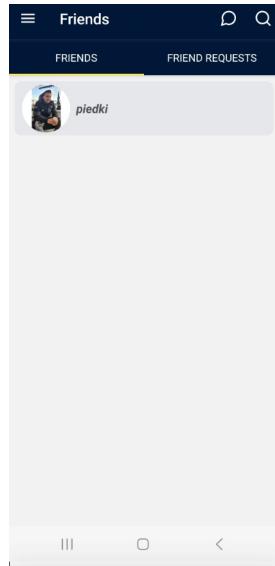
Ismét látható, hogy változás történt a gombok megjelenítésében. Két új gombunk van, a „Clear route” aminek a célja, hogy eltüntesse az útmutatót a térképről és a „Share Route”, ami megossza ezt az útvonalat más felhasználókkal(lásd 4.22 ábra), ameddig nem nyomjuk meg a „Clear route” gombot.



4.22. ábra. Trips képernyő

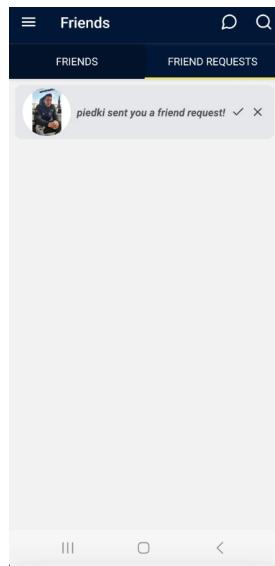
A „Trips” képernyő, azért lett létrehozva, hogy lehetőséget adjon más személyeknek új utakat kipróbálni biciklijükkel azáltal, hogy megoszták egymás között tapasztalataikat. Két opció áll mindenkinél a rendelkezésére, megtekinteni az útvonalat a térképen a „View Route” gomb lenyomásával vagy elmenteni a „Save Route” gomb által.

A „Friends” képernyő az alkalmazáson lévő barátoknak a megjelenítéséért szolgál. Az alábbi képen (4.23) meg lehet tekinteni a képernyőnek az első nézetét.



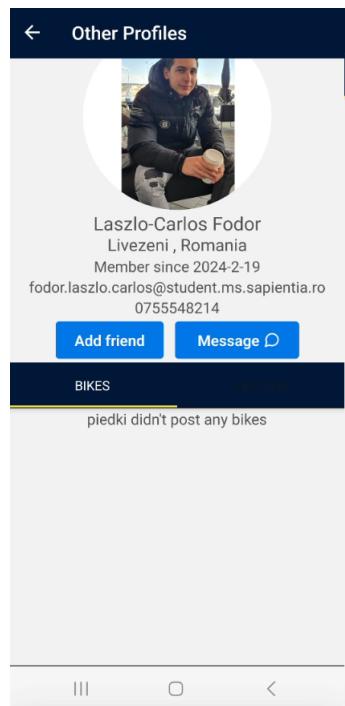
4.23. ábra. "Friends" képernyő első nézete

A második nézetben (lásd 4.24 ábra) szerepelnek az aktív barátsági kérések. Észrevehető két ikon a kérésnek a jobb oldalán, a pipával el lehet fogadni, míg az „x” ikon elérintésével el lehet utasítani a bejelölést.



4.24. ábra. "Friends" képernyő második nézete

Az alábbi ábra (4.25) egy útmutatóként szerepel, aminek a célja, hogy ábrázolja a módját annak, hogy hogyan lehet ismerősöket bejelölni. Az „Add friend” gomb megnyomásával, a kérés el lesz küldve az illető felhasználónak.



4.25. ábra. Ismerős bejelölésének az ábrája

Összefoglaló

A dolgozatom legfőbb célja egy olyan szociális platform létrehozása volt a biciklis közösség számára, amelyen keresztül barátságok alakulhatnak ki és amely edukatív jellegű. Ezen a platformon a felhasználók megoszthatják egymással tapasztalataikat, hogy mindenki számára jobb biciklizési élményt biztosítsanak.

A bevezetőben röviden bemutattam a projektemnek a fontosabb részeit, amelyek megvalósításáért én voltam felelős, valamint megismertettem a célkitűzéseimet és a projekt munkamenetét.

A dolgozat további részében részletesen ismertettem az alkalmazásban használt technológiák jelentőségét és alkalmazását. Kiemelten foglalkoztam a React Native technológiával, amely az alkalmazás felületén megjelenő komponensekért volt felelős.

A Firebase-t használtam a szöveges adatok, képek és felhasználói információk tárolására, míg az Expo lehetőséget biztosított az alkalmazásnak a futtatására és annak kitelepítésére. Következőkben a felhasználói követelmények kerültek témára, ahol részletesen leírtam, hogy a különböző felhasználói státuszok szerint milyen tevékenységeket lehet elvégezni a platformon.

Továbbfejlesztési lehetőségek implementálnák az alkalmazásba egy olyan rendszert, amely bicikli hírekkel kapcsolatos cikkeket jelenítene meg, vagy egy kialakítanák egy bicikli kölcsönző rendszert, ahol a felhasználók ki tudnák adni bérbe a biciklijüköt.

A cél az volt, hogy minden felhasználó számára személyre szabott és könnyen használható felületet biztosítsak, amely támogatja a közösségi interakciókat és az egyéni fejlődést egyaránt.

Gitlab link: <https://git.edu.codespring.ro/bicyhub/bicyhub-app.git>

Ábrák jegyzéke

1.1. "CycleStreets " útvonal leírása	13
1.2. Barátok aktivitásának megjelenítése	14
1.3. Naptár a feltöltött tevékenységekkel	15
1.4. Gödör jelzése a platformon	16
3.1. Use-Case diagram	28
4.1. Szekvenciális diagram	30
4.2. Főoldal első nézete	31
4.3. Főoldal második nézete	32
4.4. Vendég felhasználó navigációs lehetőségei	32
4.5. Bejelentkezási oldal	33
4.6. Regisztrálási oldal	33
4.7. Főoldal bejelentkezett felhasználó esetén	34
4.8. Üzenetlista	34
4.9. Üzenetezés	34
4.10. Bejelentkezett felhasználónak a navigáció lehetőségei	35
4.11. Profil oldal első nézete	36
4.12. Profil oldal második nézete	36
4.13. Profil szerkesztését szolgáló oldal	37
4.14. Cycling Tour képernyő	38
4.15. Útvonal követő elindítva	38
4.16. Útvonal elmentésének lehetősége	39
4.17. Túra részleteinek elmentése	39
4.18. Elmentett útvonalak	40
4.19. Választás a térképen való megjelenítés és a részletek megtékinése között	40
4.20. Biciklitúra részletek	41
4.21. Biciklitúra koordináták térképen	41
4.22. Trips képernyő	42
4.23. "Friends" képernyő első nézete	43
4.24. "Friends" képernyő második nézete	43
4.25. Ismerős bejelölésének az ábrája	44

Irodalomjegyzék

- [Ali13] Juzer Ali. *Instant Node Package Manager*. Packt Publishing Ltd, 2013.
- [Bug19] Daniel Bugl. *Learn React Hooks: Build and refactor modern React.js applications using Hooks*. Packt Publishing Ltd, 2019.
- [CCDN18] Nilanjan Chatterjee, Souvik Chakraborty, Aakash Decosta, and Asoke Nath. Real-time communication application based on android using google firebase. *Int. J. Adv. Res. Comput. Sci. Manag. Stud*, 6(4), 2018.
- [CYGV21] Pankaj Chougale, Vaibhav Yadav, Anil Gaikwad, and B Vidyapeeth. Firebase-overview and usage. *International Research Journal of Modernization in Engineering Technology and Science*, 3(12):1178–1183, 2021.
- [Gac15] Cory Gackenheimer. *Introduction to React*. Apress, 2015.
- [GOC] SAMUEL GOZNÍK. Device control system and mobile application.
- [JVDW16] Remo H Jansen, Vilic Vane, and Ivo Gabe De Wolff. *TypeScript: Modern JavaScript Development*. Packt Publishing Ltd, 2016.
- [KGT⁺23] Ram Kesavan, David Gay, Daniel Thevessen, Jimit Shah, and C Mohan. Fi-restore: The nosql serverless database for the application developer. 2023.
- [Kna] Matej Knazík. Paper plane: Openapi based mocking tool.
- [Le23] Thai Le. Comparing state management between redux and zustand in react. 2023.
- [MM17] Laurence Moroney and Laurence Moroney. Cloud storage for firebase. *The definitive guide to firebase: build android apps on google's mobile platform*, pages 73–92, 2017.
- [SMP21] Adam D Scott, Matthew MacDonald, and Shelley Powers. *JavaScript Cookbook*. " O'Reilly Media, Inc.", 2021.
- [Spi12] Diomidis Spinellis. Git. *IEEE software*, 29(3):100–101, 2012.
- [Tri20] Huy Trinh. A practical approach to javascript testing. 2020.