

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Kaupo Kurvits 18964IADB

# **Rakendus matkaradade seisukorra hindamiseks**

Bakalaureusetöö

Juhendaja: Einar Kivisalu  
Magistrikraad

Tallinn 2023

## **Autorideklaratsioon**

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Kaupo Kurvits

15.05.2023

## **Annotatsioon**

Käesoleva töö eesmärgiks on luua rakendus, mis võimaldab kasutajatel saada paremat ülevaadet erinevatest matakaraja seisukordadest, võimaldades lihtsasti otsida erinevaid matkaradasid ning lisada radade kohta erinevaid pilte ning kommentaare oma kogemustest.

Antud töös kirjeldatakse milliseid raamistikke ja teeke kasutatakse tulemuse saamiseks nii mobiilirakenduses kui tagarakenduses.

Töö tulemusena valmis rakendus, mis töötab iOS kui Android operatsioonisüsteemil.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 17 leheküljel, 4 peatükki, 12 joonist, 2 tabelit.

## **Abstract**

### **Application for Assessing the Condition of Hiking Trails**

The objective of this Thesis was to create an application that provides users with better information about current conditions of various hiking trails. Allowing to search easily various hiking trails along with the ability to upload photos and share comments about their experience.

In this work is described which frameworks and libraries are used to get the desired outcome for both the mobile application and the backend system.

As a result of this work, an application has been created that is compatible on both iOS and Android platforms.

The thesis is in Estonian and contains 17 pages of text, 4 chapters, 12 figures, 2 tables.

## Lühendite ja mõistete sõnastik

Android	Google mobiiliseadmete operatsioonisüsteem
API	Arvutisüsteemide instituut
C	Programmeerimiskeel
C#	Programmeerimiskeel
Dart	Programmeerimiskeel
Flutter	Google loodud raamistik mobiilirakenduste arendamiseks
GPS	Üleilmne asukoha määramise navigatsiooni süsteem
HTTP	Võrguprotokoll, mis aitab veebilehitsejatel serveritega suhelda veebis
iOS	Apple mobiiliseadmete operatsioonisüsteem
Java	Programmeerimiskeel
JWT	JSON Web Token on standard andmete loomiseks koos digitaalalkirja ja krüpteerimisega
Kotlin	Programmeerimiskeel
Python	Programmeerimiskeel
REST	Tarkvaraarhitektuuri laad, mis seab veebirakenduse loomisele kindlad piirid
SQL	Struktureeritud päringukeel, mida kasutatakse andmebaasisüsteemides
Swift	Programmeerimiskeel

## Sisukord

1 Sissejuhatus .....	9
1.1 Probleem.....	9
1.2 Eesmärk .....	9
1.3 Olemasolevad rakendused .....	10
2 Rakenduste funktsionaalsus.....	11
2.1 Rakendusele esitatavad nõuded .....	11
2.2 Mobiilirakenduse valik.....	12
2.3 Flutter.....	13
2.4 Flutteri teegid.....	14
2.4.1 Flutter map teek.....	14
2.4.2 Dio teek .....	15
2.4.3 Carousel slider teek .....	15
2.5 Tagarakendus.....	16
2.5.1 Keeled.....	17
2.5.2 SQL Alchemy teek .....	17
2.5.3 Alembic teek.....	19
2.5.4 PyJWT teek .....	19
3 Loodud rakendus .....	21
3.1 Edasiarenduse võimalused.....	24
4 Kokkuvõte .....	25
Kasutatud kirjandus .....	26
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	27

## Jooniste loetelu

Joonis 1. Näide dio konfigureerimisest .....	15
Joonis 2. Näide dio päringust .....	15
Joonis 3. Näide liuglevate piltide esitamiseks .....	16
Joonis 4. Näide SQL Alchemy mudel .....	19
Joonis 5. Näide käskudest Alembic migratsiooni koostamiseks ja käivitamiseks .....	19
Joonis 6. Näide JWT koostamiseks .....	20
Joonis 7. Näide privaat- ja avaliku võtme lugemiseks rakenduses.....	20
Joonis 8. Näide rakenduse otsingust.....	21
Joonis 9. Näide mullile peale vajutamisest.....	21
Joonis 10. Näide avatud detail vaatest.....	22
Joonis 11. Näide suurendatud detail vaatest.....	23
Joonis 12. Näide sisselogimise vaatest .....	23

## **Tabelite loetelu**

Tabel 1. Funktsionaalsed nõuded .....	11
Tabel 2. Mittefunktsionaalsed nõuded .....	12



# **1 Sissejuhatus**

Üha enam inimesi otsib aktiivseid ja loodussõbralikke puhkusevõimalusi. Matkamine on muutunud populaareks tegevuseks, mis võimaldab inimestel avastada loodust, parandada füüsilist tervist ja vähendada stressi. Matkarajad on loodusturismi ja matkamise lahutamatu osa.

## **1.1 Probleem**

Inimestel kellele meeldib käia matkaradadel teevad tihti valiku, kas ilma järgi minna jalutuskäigule või mitte. Rajale kohale jõudes võib tekkida olukord, kus ollakse oma valikus pettunud või on valitud vale riietus matkaraja läbimiseks.

On olemas küll erinevaid matka rakendusi, kuid neil puudub tihtipeale värske teave matkaraja kohta. Mõni rakendus laseb lisada ise erinevaid radu, kuid sellised rakendused on enamjaolt multifunktsionaalsed ning pole mõeldud ainult matkaradade jaoks. See omakorda raskendab radade otsimist ning ei saa olla kindel, et valitud rada on tõesti mõeldud matkarajana.

## **1.2 Eesmärk**

Rakendus annaks matkajatele kiire ja lihtsa ülevaate matkaradade seisundist võimaldades neil paremini planeerida oma matka ja vältida ootamatuid takistusi nagu üleujutused, lumeolud või muud tingimused, mis võivad teekonna keeruliseks muuta. Luua võimalused kasutajate tagasiside kogumiseks ja jagamiseks, et parandada matkaradade teavet. Võimalikult uudse info lisamine teiste kasutajate poolt annab kasutajatele nendest ohtudest hea ülevaate.

Kasutajatele nähtav rakendus peaks olema kättesaadav läbi nutitelefone ning erinevatel mobiili platvormidel. Kuna suurimad ja tuntumad tegijad on Android- ja iOS seadmed, siis peaks see kättesaadav olema vähemalt neis seadmetes.

Matkaradade asukoha kuvamiseks tuleb kuvada need kaardil ning neid peab ka olema võimalik otsida. Kasutajatel on võimalus radade kohta teha pilte ning lisada kommentaare.

Tagarakendus tegeleb andmete väljastamisega ning äri loogikaga, samuti andmebaasi suhtlusega. Kogu süsteem peab võimaldama kasutajate autentimist ja autoriseerimist.

Kokkuvõttes peaks matkaraja rakendus toimima mitmekülgse ja kasutajasõbraliku platvormina, mis aitab parandada matkamise kogemust, tõsta keskkonnateadlikust ja tagada ohutu matkamine.

### **1.3 Olemasolevad rakendused**

Enamus olemasolevad rakendused pakuvad loetelu erinevatest matkaradadest mida saab otsida. Need sisaldavad infot raja asukoha kohta ning annavad ülevaate sellest. Sisaldades paremal juhul pilte matkarajast mis on tehtud raja lisamisel süsteemi, kuid puudub ülevaade hetke seisundi kohta [1].

On olemas multifunktsionaalseid rakendusi, mis pakuvad võimalusi kus saab sisestada erinevaid matkaradu, jooksuradu ja rattaradu nagu näiteks Wikiloc [2]. See võimaldab ühest rakendusest leida erinevaid liiki radu, kuid samas teeb otsingu ebamugavamaks kui üritada otsida kindlat tüüpi rada.

Ramblr pakub võimalust salvestada ja jagada oma liikumist, olgu see siis matkamine, rattasõit, suusatamine, kanuumatk, automatk, pikem reis või mis tahes muu matk [2].

## 2 Rakenduste funktsionaalsus

Rakenduste funktsionaalsus on kriitilise tähtsusega olles aluseks kasutajakogemusele, tõhususele, skaleeritavusele ning efektiivsusele. Kasutajakogemus sõltub suurel määral rakenduse funktsionaalsusest. Kui rakendus on kasutajasõbralik, suurendab see kasutajate rahulolu. Lisaks ootab kasutaja, et rakendus oleks sujuv ja vastaks kiiresti.

### 2.1 Rakendusele esitatavad nõuded

Matkarakendus peaks olema varustatud mitme olulise funktsionaalsusega. Kaardifunktsioon on üks neist, võimaldades kasutajatel vaadata ja navigeerida matkaradade vahel. Kasutajale peaks olema võimalus lisada pilte, mis kujutavad matkaraja seisukorda ja kommentaare, mis võimaldab jagada oma mõtteid teiste kasutajatega. Otsingu loetelu funktsioon peaks võimaldama hõlpsalt leida konkreetseid matkaradu. Rakendus peaks olema avatav nii Android kui iOS operatsioonisüsteemil.

Autentimise funktsioon on samuti oluline, et tagada kasutajate konto turvalisus. See võiks olla traditsiooniline kasutajanime ja parooli kombinatsioon, aga võib ka sisaldada täiendavad turvameetmeid. Funktsionaalsed nõuded on esitatud tabelis 1.

Tabel 1. Funktsionaalsed nõuded

Number	Funktsionaalsete nõuete sisu
FN1	Rakendusel peab olema kaardifunktsioon
FN2	Rakendusel peab olema piltide lisamise võimalus
FN3	Rakendusel peab olema kommentaaride lisamise võimalus
FN4	Rakendusel peab olema otsingu tulemuste loetelu
FN5	Rakendus peab olema avatav nii Android kui iOS operatsioonisüsteemil
FN6	Rakenduses peab olema autentimine

Mittefunktsionaalsed nõuded matkarakenduses hõlmavad aspekte, mis ei ole seotud konkreetse rakenduse funktsioonidega, vaid pigem puudutavad kasutaja kogemust, jõudlust ja süsteemi usaldusväärsust. Rakendus peaks olema kasutajasõbralik ja lihtne kasutada isegi neile, kes pole varem matkarakendust kasutanud. Kasutajate parooli ning JWT krüpteerimisel tuleb kasutada turvalisi algoritme ja võtmepikkusi. Rakendust peaks

olema võimalik edasi arendada ja lisada uut funktsionaalsust. Mittefunktsionaalsed nõuded on esitatud tabelis 2.

Tabel 2. Mittefunktsionaalsed nõuded

Number	Mittefunktsionaalsete nõuete sisu
MF1	Rakendus peab olema lihtsasti kasutatav
MF2	Rakendusel peab kaarti saama suumida sõrmedega
MF3	Rakendusel peab saama avada asukohta kaardil sõrme vajutusega
MF4	Rakendusel peab olema võimalus suumida pilte sõrmedega
MF5	Rakenduses peab olema võimalik liikuda kommentaaride vahel sõrme liigutusega

## 2.2 Mobiilirakenduse valik

Mobiilirakenduste arendamisel on peamiselt kaks lähenemist: tavalised ehk põlisrakendused ja hübriidrakendused. Mõlemal lähenemisel on oma eelised ja puudused.

Tavalised mobiilirakendused on loodud konkreetse platvormi jaoks nagu Android või iOS. Androidi puhul kasutatakse Java või Kotlin ja iOS puhul C või Swift programmeerimiskeelt. Iga platvorm nõuab eraldi koodibaasi ja arendusprotsessi. Kuna põlisrakendused on loodud spetsiifiliselt konkreetse platvormi jaoks, pakuvad nad tavaliselt paremat jõudlust ja sujuvamat kasutajakogemust. Nad on optimeeritud oma platvormi riist- ja tarkvarale. Lisaks võimaldavad arendajatel kasutada kõiki seadme riist- ja tarkvarafunktsioone nagu GPS, andurid, kaamera jne.

Hübriidrakendused pakuvad üht koodibaasi, mida saab kasutada nii Androidi kui ka iOS-i platvormide jaoks. See vähendab arenduse aega kuna pole vaja luua eraldi rakendusi iga platvormi jaoks. Hübriidrakendused jäävad tavaliselt alla põlisrakenduste jõudlusele, kuna nende kood jookseb enamjaolt veebilehitseja abil loodud ümbrises. See tähendab, et hübriidrakenduste jõudlus on mõnevõrra aeglasem. Neil on tavaliselt piiratud juurdepääs seadme riist- ja tarkvarafunktsioonidele [3].

On ka olemas platvormivahelised raamistikud, mis võimaldavad luua rakendusi mitmele platvormile. See lähenemine on midagi vahepealset põlis- ja hübriidrakendusest. Erinevalt hübriidrakendusest kasutavad platvormivahelised raamistikud oma

renderdamis mootorit ja kompileerivad koodi otse platvormispetsiifiliseks masinkoodiks, mis tagab parema jõudluse ja kasutajakogemuse.

Koostatav rakendus peab töötama nii Android kui iOS-i peal, seega jääb valikusse kas hübriidrakendus või platvormivaheline rakendus. Kuna platvormivaheline rakendus on kiirem ning sellel on suurem ligipääs erinevatele seadme riist- ja tarkvarafunktsioonidele, siis osutub see suhteliselt kindlaks valikuks.

Kuna olen varem kokku puutunud Flutteriga, siis otsustasin selle kasuks.

## 2.3 Flutter

Flutter on avatud lähtekoodiga kasutajaliidese raamistik, mille on loonud Google, pakkudes arendajatele võimalust luua Androidi ja iOS-i rakendusi ühe koodibaasi abil. Selle eesmärk on muuta arendusprotsess kiireks, tõhusaks ja paindlikuks, et luua erinevatele seadmetele mõeldud kvaliteetseid rakendusi. Lõputöö rakenduse arendamisel on mitmeid põhjuseid, miks valida Flutteri raamistik.

Esiteks pakub Flutter kiiret arendusprotsessi tänu *hot reload* funktsioonile [4]. See võimaldab arendajatel näha koodis tehtud muudatusi reaajas ilma rakendust taaskäivitatama, mis aitab oluliselt kiirendada arendusprotsessi ja säästa väärtuslikku aega.

Teiseks kasutab Flutter Dart programmeerimiskeelt, mis on optimeeritud kiireks jõudluseks. Rakendus kompileeritakse otse masinkoodiks, mis tagab sujuva kasutajakogemuse ja väikese ressursikasutuse [5]. Flutterile on loodud suurel hulgal teeke, mis võimaldavad arendajatel kiiresti lisada erinevaid funktsioone ja integreerida kolmandate osapoolte teenuseid. Läbi selle on võimalik mugavalt kasutada rakenduses mitmekülgset funktsionaalsust, mis vastab kasutajata vajadustele.

Kolmandaks pakub Flutter paindlikku disaini ja komponentide valikud, mis võimaldab luua erinevate platvormide jaoks sobivat disaini. Lisaks toetab Flutter Google Material ja Apple Cupertino disainisüsteeme, mis annab rakendusele professionaalse välimuse ja tagab, et rakendus näe välja loomulik igal toetatud platvormil.

Lõpuks on Flutteri populaarsuse tõttu selle kasutajaskond suur, mis tähendab, et arendajad saavad tugineda laiale teadmiste pagasile ja ressurssidele [6]. Sellest tulenevalt on probleemide lahendamine ja tõrgete eemaldamine lihtsam kuna arendajatel on võimalus kogukonnaga suhelda, küsimusi esitada ja vajalik teave leida. Aktiivne kogukond ja suur hulk õppematerjale muudavad Flutteri arenduskeskkonna arendajatele meeldivaks ja tõhusaks.

Kokkuvõtteks, kui otsida loodava rakenduse jaoks tulevikukindlat ja paindlikku arendusraamistikku, on Flutter suurepärase valik. Selle abil saab luua kvaliteetseid ja mitmekülgseid rakendusi, mis töötavad sujuvalt erinevatel platvormidel. Pakkudes samal ajal suurepärase kasutajakogemust ja kiiret arendusprotsessi. Flutteri kasutamine võimaldab keskenduda rakenduse põhifunktsioonidele ja väärtustele, samal ajal tagades platvormide ülese toimimise ja kättesaadavuse.

## **2.4 Flutteri teegid**

Matkaraja mobiilirakenduse jaoks on valitud mitmed teegid, mis lihtsustavad ja kiirendavad rakenduse loomist. Eesmärgiks on aja säästmine ning ei pea leiutama iga funktsionaalsuse jaoks midagi uut vaid teekide abil saab lisada rakendusse funktsionaalsust. Teegid on sageli loodud kogunud arendajate poolt ning tõenäoliselt on loodud kood läbi testitud. Flutteri ökosüsteemis on saadaval palju erinevaid teekide funktsioone nagu võrguühendus, animatsioonid, kaardid, pilditöötlus ja palju muud. Teegid võimaldavad ka hõlpsalt integreerida kolmandate osapoolte teenuseid ja API-sid. Järgnevalt tooksingi välja peamised kasutatavad teegid matkaraja mobiilirakenduses.

### **2.4.1 Flutter map teek**

Flutter map võimaldab arendajatel lisada interaktiivseid kaarte Flutteri mobiilirakendustesse. See põhineb Leaflet-i kaardiraamistikul mis on avatud lähtekoodiga [7]. Flutter map võimaldab kasutada erinevaid kaarditüüpe nagu OpenStreetMap, Google Maps, MapBox jt. See võimaldab arendajatel valida parima kaardiallika oma rakenduse vajadustele. Kaardile on võimalik lisada markereid, jooni ja muid kujundeid, mis on kasulikud erinevate geograafiliste andmete esitamiseks ja kasutajatele kuvamiseks. Flutter map võimaldab jälgida sündmusi nagu kaardi klõpsamine, markeri lohistamine, kaardi suurendamine ja muid interaktiivseid funktsioone.

## 2.4.2 Dio teek

Dio võimaldab lihtsalt suhelda erinevate veebiteenuste ja API-dega (Joonis 2), pakkudes samal ajal rohkem paindlikkust kui Darti sisseehitatud „http“ teek. See võimaldab arendajatel muuta ja filtreerida võrgupäringuid ning nende vastuseid. Näiteks saab seda funktsionaalsust kasutada autentimiseks, puhverdamiseks ja vigade haldamiseks. Dio pakub ka palju konfigureerimisvõimalusi (Joonis 1), mis võimaldavad määrata rakendusele sobilikke seadistusi nagu päised, baas-URL, konverterid jms.

```
dio.options = BaseOptions(  
  baseUrl: 'http://10.0.2.2:8000',  
  connectTimeout: const Duration(seconds: 5),  
  receiveTimeout: const Duration(seconds: 3),  
  headers: {  
    'Content-Type': 'application/json',  
  },  
);
```

Joonis 1. Näide dio konfigureerimisest

```
Future<T> get(T Function(Map<String, dynamic>) fromJson,  
String uuid) async {  
  try {  
    Response response = await dio.get('$endpoint/$uuid');  
    return fromJson(response.data);  
  } on DioError catch (e) {  
    throw Exception('Failed to load data: ${e.message}');  
  }  
}
```

Joonis 2. Näide dio päringust

## 2.4.3 Carousel slider teek

Carousel slider on populaarne teek Flutteri rakenduse jaoks, mis võimaldab lisada mitmest elemendist koosnevat keritavat galeriid (Joonis 3). See pakub erinevaid kohandamisvõimalusi ja animatsioone, et pakkuda atraktiivset ja sujuvat kasutajakogemust. Teeki on võimalik mitmeti konfigureerida nagu näiteks elementide vaheline kaugus, automaatne kerimine, animatsioonide kasutus jne. Üks suurimaid plusse piltide kuvamisel on võimalus laadida pilte vastavalt vajadusele. Selle tulemusel ei pea rakendus igat pilti eelnevalt alla laadima ning elemente välja renderdama.

```
CarouselSlider.builder(  
  options: CarouselOptions(  
    height: carouselHeight,
```

```

        initialPage: 0,
        viewportFraction: 1,
        aspectRatio: 16 / 9,
        padEnds: false,
        enableInfiniteScroll: false,
        onPageChanged: (index, reason) {
            _currentIndexNotifier.value = index;
        },
    ),
    carouselController: _carouselController,
    itemBuilder: (BuildContext context, int itemIndex,
int pageIndex) {
        return FutureBuilder(
            future: _getImage(pictures[itemIndex].uuid),
            builder: (BuildContext context, AsyncSnapshot
snapshot) {
                if (snapshot.connectionState ==
ConnectionState.waiting) {
                    return const Center(child:
CircularProgressIndicator());
                } else if (snapshot.hasError) {
                    return const Center(child: Text('Error
loading image'));
                } else {
                    return _buildImageContainer(pictures,
itemIndex, snapshot);
                }
            },
        );
    },
    itemCount: pictures.length,
);

```

Joonis 3. Näide liuglevate piltide esitamiseks

## 2.5 Tagarakendus

Tagarakenduse valiku puhul oleksid programmeerimiskeelte valikuteks Python, Java ning C# kuna nendega olen kõige rohkem kokku puutunud. Koostava rakenduse puhul tundub mõistlik valik Python. Pythoni valisin selle tõttu, kuna selle peal töötavaid raamistikke on kerge ülesse seada ning väiksemate projektide jaoks on see piisav. Arendajate seas on see populaarne ning teekide valik on suur. Pythoni raamistikke mis sobivad tagarakenduse arendamiseks on mitmeid nagu näiteks FastApi, Flask, Django jne. Parim valik neist matkaraja tagarakenduse tegemiseks tundub FastApi. FastApi põhineb Starlette-i asünkroonse raamistiku ja Pydantic andmete



valideerimise teegi peal [8] mis võimaldab kiiret, paindlikku ja tõhusat REST API arendust. REST API on veebiteenuste arhitektuuri stiil, mis võimaldab süsteemide vahelist suhtlust, kasutades standardseid HTTP protokolle ja meetodeid [9]. Valitud raamistik pakub selgeid ja kompaktsid süntakseid, mis muudavad koodi lihtsalt loetavaks ja kergemini hooldavaks.

Andmebaasiks sai valitud PostgreSQL kuna see on vabavara [10] ning on väga populaarne valik arendajate seas kes otsustavad kasutada enda projektis relatsioonilist andmebaasisüsteemi. Selle üles seadmine tagarakendusega suhtluseks on võrdlemisi lihtne kasutades FastApi raamistikku.

### **2.5.1 Keeled**

Java, Python ja C# on tuntud ja laialt kasutatavad programmeerimiskeeled, kuid neil on erinevad omadused, eelised ning puudused.

Java on tuntud selle poolest, et see töötab praktiliselt igal platvormil ilma, et oleks vaja platvormil viia läbi olulisi muudatusi. Sellel on laiapõhine teekide, raamistike ja tööriistade valik. Miinuseks võiks välja tuua süntaksi, mis on mõnevõrra pikk.

Python on interpreteeritav keel ning tuntud oma lihtsa ja loetava süntaksi poolest, mis võimaldab kiirendada alguses arendusprotsessi. Seda saab kasutada võrdlemisi paljudes valdkondades nagu veebiarendus, andmete andus, tehisintellekt ja palju muud. Python on võrdlemisi aeglane võrreldes teiste keeltega nagu C# või Java [11].

C# on kompileeritud keel mis on tihedalt integreeritud Microsofti ökosüsteemiga ning seda saab kasutada erinevates valdkondades nagu veebiarendus, mobiilirakendused ja mängude arendamisel. Viimasel ajal on C# muutunud rohkem platvormide vahelisemaks, kuid siiski jääb alla Java-le selles osas.

### **2.5.2 SQL Alchemy teek**

SQL Alchemy on Python teek, mis on mõeldud andmebaasi interaktsiooni lihtsustamiseks ja efektiivsuse tõstmiseks. See võimaldab SQL päringuid teha rohkem

Python keele moodi kui SQL lausetena välja kirjutades. SQL Alchemy SQL väljendid on funktsionaalne programmeerimisstiilis SQL-i abstraktsioon.

Samuti on mugav hallata andmebaasi transaktsioone pakkudes transaktsiooni alustamise, kinnitamise ja tühistamise funktsioone. See haldab andmebaasiühendusi automaatselt ning optimeerib ressursikasutust.

SQL Alchemy toetab mitmesuguseid andmebaasisüsteeme nagu PostgreSQL, MySQL, Oracle ja teisi [12].

Objekt-relatsiooniline kaardistamine (ORM) võimaldab luua andmebaasi vahelised seosed, muutes andmebaasi andmetega töötamise lihtsamaks. See vähendab vajalikust kirjutada keerulisi SQL-päringuid ja võimaldab lihtsamalt töötada Pythoni objektidega (Joonis 4). Samuti kergendab selle teegi kasutusele võtmine olukorra, kus tahetakse hiljem andmebaasisüsteemi välja vahetada. Nimelt saab väga vähe koodi muudatustega minna üle teisele andmebaasisüsteemile kuna päringute kood jääb samasuguseks. On vaja muuta ainult andmebaasi ühenduse sõne ning muuta teek, mis tegeleb andmebaasi ning rakenduse omavahelise suhtlusega.

```
class Track(BaseMeta):
    __tablename__ = 'track'

    country: CountryType = Column(Enum(CountryType),
    nullable=False, comment='Country of Track')
    county: str = Column(String(255), nullable=False,
    comment='County of Track')
    latitude: float = Column(Numeric(precision=12, scale=8,
    asdecimal=True), nullable=False, comment='Latitude')
    longitude: float = Column(Numeric(precision=12, scale=8,
    asdecimal=True), nullable=False, comment='Longitude')
    name: str = Column(String(255), index=True, nullable=False)
    info: str = Column(String(512))
    length: Float = Column(Numeric(precision=8, scale=2))
    length_type: LengthType = Column(Enum(LengthType))

    files: Mapped[List['File']] = relationship('File',
    back_populates='track')
    comments: Mapped[List['TrackComment']] =
    relationship('TrackComment', back_populates='track')

    __table_args__ = (
        UniqueConstraint('name', 'county', 'country',
    name='unique_track_name_county_country'),
```

```
    {'schema': SCHEMA_NAME}  
)
```

Joonis 4. Näide SQL Alchemy mudel

### 2.5.3 Alembic teek

Alembic on teek mis on loodud andmebaasi migratsioonide haldamiseks. See võimaldab hallata andmebaasi skeemi muutusi ajas, jälgides erinevaid versioone andmebaasi struktuurist. Selle kaks peamist funktsionaalsust on migratsioonid ja versioonikontroll. Migratsioonid on andmebaasid struktuurid muudatuste kirjeldused mis võimaldavad andmebaasi skeemi muuta ilma olemasolevate andme kadumiseta. Kuigi iga migratsiooni tuleks skeptiliselt üle vaadata, et ta teeb seda mis on mõeldud. Alembic võimaldab kirjutada migratsioone Python koodis, kasutades SQL Alchemy väljendeid. See muudab migratsioonid lihtsamaks ja loetavamaks aidates kaasa arendusprotsessile. Migratsioone on võimalik käivitada ka käsurealt (Joonis 5).

Versioonikontroll Alembic võimaldab hoida järjekorda rakenduse andmebaasi struktuuri muudatustes. Alembic kasutab versioonikontrolli süsteemi, mis võimaldab tagasi pöörduda varasemate andmebaasi skeemide juurde, kui on vaja teha muudatusi või parandada vigu. Tagasi pöördumisega tuleb olla ettevaatlik kuna võib tekkida andmekadu.

Alembicu peamine eelis on see, et see vähendab andmebaasiga seotud vigade riski kuna sellega on võimalik planeerida ja testida muudatusi enne, kui need reaalsesse süsteemidesse lisatakse.

```
alembic revision --autogenerate -m "Create tables"  
alembic Upgrade head
```

Joonis 5. Näide käskudest Alembic migratsiooni koostamiseks ja käivitamiseks

### 2.5.4 PyJWT teek

PyJWT on Pythoni teek, mis on loodud JSON Web Tokenit (JWT) loomiseks (Joonis 6). Sellega saab samuti JWT kontrollida ja dekodeerida. JWT kasutatakse andmete turvaliseks edastamiseks eri osapoolte vahel. Matkaraja rakenduse puhul tagarakenduse ja mobiilirakenduse omavahelise suhtluse jaoks. See muudab selle sobivaks teegiks rakendusele kuna andmete turvalisus, privaatsus ja autentimine on oluline.

JWT loomisel võimaldab teek lisada selle sisse andmeid ning peale seda kasutada salajast võtit andmete allkirjastamiseks, et olla kindel andmete valiidsuses ning ka JWT sisu ei oleks vahepeal muutunud. Lisaks saab määrata aja kui kaua on see valideeritud.

Matkaraja rakenduses kasutatakse ES512 krüpteerimisalgoritmi, mis põhineb elliptilistel kõvertel ning see pakub võrreldes traditsiooniliste RSA-algoritmidega suuremat turvalisust [13]. ES512 tähistab elliptiliste kõverate digitaalallkirja algoritmide (ECDSA) varianti, mis kasutab elliptilise kõvera krüptograafiat, mille bitipikkus on 512 bitti.

See kasutab privaat- ja avaliku võtme krüptograafiat kus privaat võtmega krüpteeritakse JWT ning selle ehtsust kontrollitakse avaliku võtmega (Joonis 7). Avalikku võtit võib jagada kui tahetakse, et teisel osapoolel oleks võimalik kontrollida JWT autentsust.

```
def create_access_token(user: User):
    to_encode = {"sub": user.username}
    expire = datetime.datetime.utcnow() +
datetime.timedelta(minutes=30)
    to_encode.update({"exp": expire})
    encoded_jwt = jwt.encode(to_encode, private_key,
algorithm="ES512")
    return encoded_jwt
```

Joonis 6. Näide JWT koostamiseks

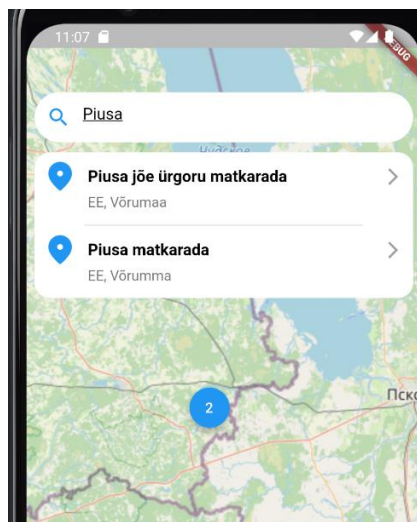
```
with open("cert/private_key_ec.pem", "rb") as key_file:
    private_key = serialization.load_pem_private_key(
        key_file.read(),
        password=None,
        backend=default_backend()
    )
```

```
with open("cert/public_key_ec.pem", "rb") as key_file:
    public_key = serialization.load_pem_public_key(
        key_file.read(),
        backend=default_backend()
    )
```

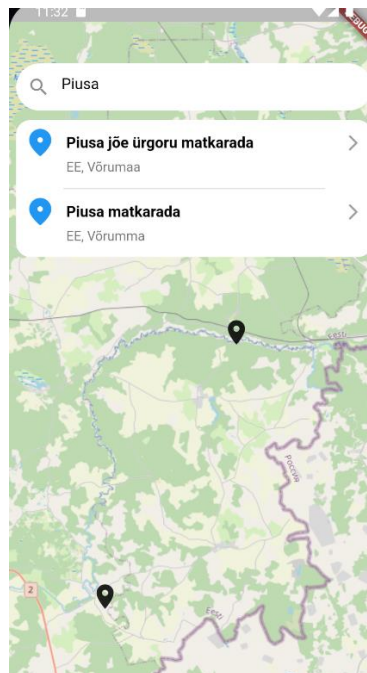
Joonis 7. Näide privaat- ja avaliku võtme lugemiseks rakenduses

### 3 Loodud rakendus

Loodud rakendus võimaldab otsida erinevaid matkaradasid kasutades otsimiseks tagarakenduse andmebaasis olevate radade nimesid. Nimede otsing toimub sõltumata sellest, kas otsingu tekstis kasutatakse suuri või väikeseid tähti. Otsingu tulemused kuvatakse rakenduses loeteluna ning samal ajal kuvatakse kaardile nende asukohad (Joonis 8). Kui asukohad on piisavalt ligidal, siis kuvatakse nad mullina mille sees on kogu arv piirkonna matkarajadest. Sellele vajutades suureneb kaart ning ekraanile ilmuvad rajad eraldiseisvalt (Joonis 9).

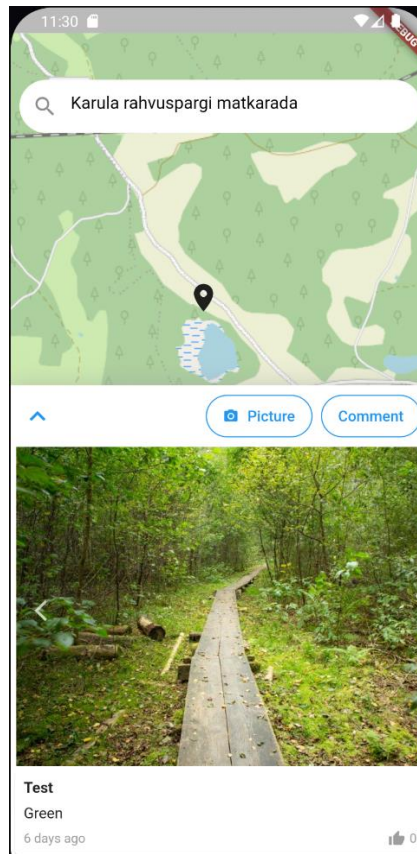


Joonis 8. Näide rakenduse otsingust



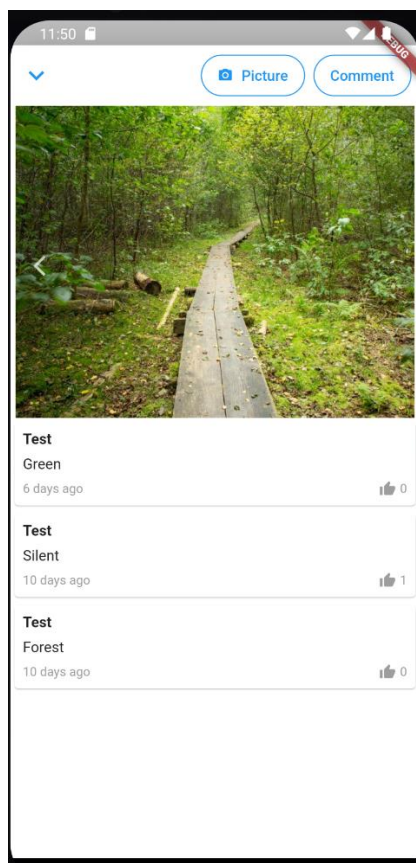
Joonis 9. Näide mullile peale vajutamisest

Kui vajutada asukoha ikoonile, siis avaneb detailne vaade kust on võimalik näha lisatud pilte ja kommentaare (Joonis 10). Pilte on võimalik vahetada vajutades edasi või tagasi nupule, mis on kuvatud kui ikoonid või lihtsalt liigutades näppu ekraanil paremale või vasakule. Kui on jõutud viimase pildini, siis pole võimalik edasi liikuda ning edasi liikumise ikoon kaob. Erinevate kommentaaride lugemiseks on võimalus neid kerida kas üles või alla poole liigutades näppu ekraanil. Lisaks sellele on seal olemas nupud mille kaudu saab lisada nii pilti kui kommentaari.



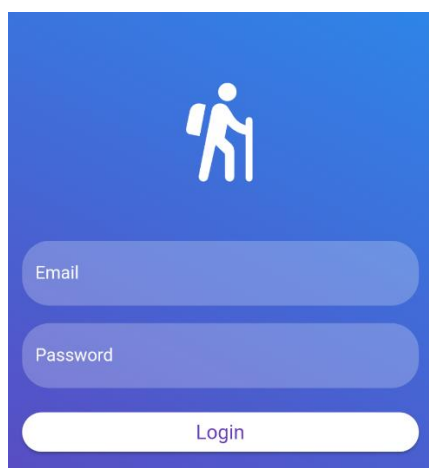
Joonis 10. Näide avatud detail vaatest

Vajutades vasakul serval olevale ikoonile detailvaates on võimalik seda suurendada ekraani täis suuruseni (Joonis 11). Pildile vajutades avaneb uus vaade mille kaudu on võimalik pilti suurendada.



Joonis 11. Näide suurendatud detail vaatest

Rakendusel on olemas sisselogimise vaade kus kasutaja peab sisestama oma email ning parooli (Joonis 12). Sisselogimise õnnestumisel tagastab tagarakendus JWT ning ebaõnnestumise korral kuvab veateate.



Joonis 12. Näide sisselogimise vaatest

### **3.1 Edasiarenduse võimalused**

Rakenduse loomise ajal on tekkinud autoril mõtteid kuidas arendada loodud rakendust edasi. Lisaks piltidele ja kommentaaridele võiks olla rakenduses rohkem tutvustavat infot matkaraja kohta. Info võiks sisaldada vähemalt teavet selle kohta, mis lisavõimalused on matkarajal ning ka väike lühitutvustus. Kasutajal oleks mugav kui saaks kuvada ka ilmastiku infot selles piirkonnas kus rada asub.

Detailandmete avamisel võiks kaart ka kuvada matkaraja teekonda, millel võiksid olla ikoonid erinevatest raja võimalustest nagu lõkkeplats, torn, matkamaja ja muud sellist. Kasutajale võiks kuvada tema asukohta rakenduse kasutamise hetkel, et oleks võimalik jälgida kas kaldutakse rajalt kõrvale või kus parasjagu asutakse.



## 4 Kokkuvõte

Eesmärgiks oli luua matkaraja rakendus milles oleks võimalik jälgida matkaraja hetkeseisukorda teiste kasutajate abil. Enamus rakenduse funktsionaalsusest sai tehtud ning rakendust on võimalik kasutada. Tagarakenduses jäid tegemata testid ning seda ei jõudnud kuskile üles paigaldada. Flutter mobiilirakendus pole veel kasutajatele läbi rakenduste poe kättesaadav. Arvatust keerulisemaks osutus erinevate matkarajade asukoha leidmine, kuna autor ei leidnud kuskilt sobivat nimistud kus oleks loetelu paljudest matkarajadest.

Loodud rakenduses saab kasutaja logida sisse, otsida matkaradasid ja jälgida piltide ning kommentaaride abil raja võimaliku värsket olukorda. Lisada pilte läbi oma mobiili ning sisestada kommentaare.

Töö käigus sai uuritud sarnaseid rakendusi ning nendest sai inspiratsiooni loodud rakenduse jaoks. Võrreldi erinevaid programmeerimiskeeli ning erinevaid võimalusi mobiilirakenduse tegemiseks. Sai välja toodud milliseid põhi teeke kasutati nii tagarakaneduse kui ka mobiilirakenduse loomiseks ning mille jaoks neid vaja oli. Valminud rakendust saab edasi arendada ja lisada funktsionaalsust.

## Kasutatud kirjandus

- [1] „Loodusega koos,“ [Võrgumaterjal]. Available: <https://www.loodusegakoos.ee/et>.
- [2] [Võrgumaterjal]. Available: <https://baltictrails.eu/et/coastal/apps>.
- [3] Latitude Technolabs, „Hybrid Mobile App Development Advantages and Disadvantages,“ [Võrgumaterjal]. Available: <https://latitudetechnolabs.medium.com/hybrid-mobile-app-development-advantages-and-disadvantages-a8b1f4c7252b>.
- [4] Google, „Flutter Hot reload,“ [Võrgumaterjal]. Available: <https://docs.flutter.dev/development/tools/hot-reload>.
- [5] [Võrgumaterjal]. Available: <https://dart.dev/overview>.
- [6] Nomtek, „Flutter vs. React Native in 2023 — Detailed Analysis,“ [Võrgumaterjal]. Available: <https://www.nomtek.com/blog/flutter-vs-react-native>.
- [7] [Võrgumaterjal]. Available: [https://pub.dev/packages/flutter\\_map](https://pub.dev/packages/flutter_map).
- [8] [Võrgumaterjal]. Available: <https://fastapi.tiangolo.com/features/>.
- [9] [Võrgumaterjal]. Available: <https://www.ibm.com/topics/rest-apis>.
- [10] [Võrgumaterjal]. Available: <https://www.postgresql.org/about/licence/>.
- [11] [Võrgumaterjal]. Available: <https://www.python.org/doc/essays/comparisons/>.
- [12] [Võrgumaterjal]. Available: <https://www.sqlalchemy.org/features.html>.
- [13] Sectigo, „What Are the Differences Between RSA, DSA, and ECC Encryption Algorithms?,“ [Võrgumaterjal]. Available: <https://sectigo.com/resource-library/rsa-vs-dsa-vs-ecc-encryption>.

# Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>

Mina, Kaupo Kurvits

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Rakendus matkaradade seisukorra hindamiseks", mille juhendaja on Einar Kivisalu
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

15.05.2023

---

<sup>1</sup> Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.