TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Kristjan Klimušev 185686IADB

Eesti keele õigekirja õppimise rakenduse arendamine Android platvormile

Bakalaureusetöö

Juhendaja: Aleksei Talisainen
magistrikraad

Tallinn 2022

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: Kristjan Klimušev

6.01.2022

Annotatsioon

Antud lõputöö raames loodi mobiilirakendus Androidi operatsioonisüsteemile, mis abistaks kasutajatel eesti keele õigekirja lihtsasti ja mugavalt õppida.

Rakenduses saab kasutaja õppida eesti keele õigekirja erinevaid liike, näiteks kokku- ja lahkukirjutamine, suur ja väike algustäht ning võõrsõnade õigekiri. Lisaks on kasutajal võimalik liituda privaatse ruumiga või luua oma privaatne ruum kohandatud sõnavaraga, kus kõik ruumiga liitunud kasutajad saavad oma skoori edetabelisse salvestada.

Töös tuuakse välja rakenduse eeldused ja tehnilised nõuded ja kirjeldatakse arendatava rakenduse arhitektuur, arenduse protsess ja kasutusel olevad tehnoloogiad.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 32 leheküljel, 5 peatükki, 12 joonist, 0 tabelit.

Abstract

Estonian Grammar Learning Mobile Application Development on the Android Platform

The main purpose of this thesis was to create an Android application, which allows users to study Estonian spelling conveniently.

The application allows the user to study different categories of Estonian grammar, for example captialization, compound words and foreign words. Secondly, the application allows users to join private rooms or create their own with custom vocabularies, where all joined users can save their scores to the leaderboard.

The thesis points out the prerequisites and the techical requirements of the mobile application. The architecture, development process and technologies used to create the application are also described.

The thesis is in Estonian language and contains 32 pages of text, 5 chapters, 12 figures, 0 tables.

Lühendite ja mõistete sõnastik

Android	Tarkvarakomplekt peamiselt mobiiltelefonidele.
Gradle	Rakenduste ehitamise automatiseerimiseks loodud tööriist.
Flutter	Flutter on Google'i loodud avatud lähtekoodiga raamistik.
SDK	Tarkvaraarenduspakett, tarkvaraarenduse tööriistade kogumik.
Apple	Ettevõte, mis arendab ja toodab riistvara ja tarkvara.
SQL	Andmebaasi relatsiooniline päringukeel.
Widget	Element graafilises kasutajaliideses, milleks võib olla näiteks nupp või kerimisriba.
Firebase	Mobiili- ja veebirakenduste arendamise platvorm.
React Native	Avatud lähtekoodiga mobiilirakenduste raamistik.
JavaScript	Objektorienteeritud programmeerimiskeel.
iOS	Apple'i arendatav mobiilsete seadmete operatisioonisüsteem.
Dart	Google poolt välja töötatud programmeerimiskeel, mis on loodud veebi ja mobiilirakenduste loomiseks.
Cloud Firestore	Skaleeritav andmebaas mobiili- ja veebirakenduste jaoks, mis on loodud Google ja Firebase poolt.
Tagarakendus	Veebiteenus, mis serveerib kasutajaliidesele andmeid.
IDE	Integreeritud programmeerimiskeskkond.
Visual Studio Code	Integreeritud programmeerimiskeskkond, mis toetab erinevaid programmeerimiskeeli.

Sisukord

Jooniste loetelu

1 Sissejuhatus

Tänapäeval on tavaks saanud, et meili või kirjatööd kirjutades vaadatakse pidevalt internetist järgi, kuidas mingit sõnapaari õige kirjutada on. Sellise võimaluse puudumisel kasutatakse raskemate sõnapaaride või lausete asemel lihtsamaid väljendeid, et vigu vältida ning selle tõttu muutub keel labasemaks, kuna ei kasutata ära keele kogu potentsiaali. Eesti keele õigekirja oskus peaks olema eestlasele elementaarne, kuid tegelikkuses on näha eesti rahva madalat eesti keele õigekirja oskust. Kuna inimeste aeg on nende jaoks väga tähtis, siis nad ei ole motiveeritud oma vabast ajast oma eesti keele õigekirja oskusi parandama.

Eelnevast lähtudes on lõputöö eesmärgiks mobiilirakenduse loomine Android platvormile, mis aitaks lihtsasti ja mugavalt eesti keele õigekirja oskusi parandada. Kuna ma olen ise sellise rakenduse kasulikkusest ja kasutamisest huvitatud, siis eeldan, et selline rakendus tuleb kasuks õpilastele, tudengitele ning ka teistele huvilistele.

Kõigepealt analüüsitakse töö teoreetilises osas juba olemasolevaid sarnaseid lahendusi, et leida mõtteid, mida saab paremini teha. Praktilises osas tuuakse välja rakenduse skoop ning kasutatud vahendid, mida arenduskäigus vaja läks. Samuti kirjeldatakse detailselt kõiki arendusprotsessi etappe ning testimist.

2 Taust

Antud peatükis kirjeldab autor probleemi, mida lahendatakse ning kirjeldab lõputöö eesmärki ja nõudeid.

2.1 Probleem

Eesti keelt peetakse teiste keeltega võrreldes raskeks, aga vaatamata sellele tahavad enamik eestlasi osata korrektset Eesti keelt. Kõik Eesti õpilased, kes tahavad oma haridusteed väärikalt läbida, peavad omama kõrget Eesti keele oskuse taset. Väljakutsed algavad juba põhikoolis, kus valmistatakse õpilasi ette gümnaasiumite vastuvõtukatseteks.

Oma lemmikusse ülikooli saamiseks peavad õpilased edukalt läbima Eesti keele riigieksami. Riigieksamil mängib suurt rolli hea õigekirja oskus, aga selle omandamine vajab suurt vaeva.

Õpilastel on vähe vaba aega, sest enamus kulub koolis käimisele, kodutöödele ning erinevatele hobidele. Inimesed ei taha vaba aega kulutada õigekirja harjutamisele, aga seda rakendust saab kasutada siis, kui muud tegevust niikuinii ei toimu, näiteks bussis istudes või siis kui voodis lebades und ei tule.

Paberil saab õigekirja harjutada ainult ühe korra ja vastuse kontrollimine ei ole tõhus.
Arvutis õigekirja harjutades on vaja internetiühendust ja peab olema kohas, kus saab arvutit kasutada. Arvutis on lisaks sellele veel palju segavaid faktoreid nagu näiteks arvutimängud ja sotsiaalmeedia.

Loodud mobiilirakendust saaks kasutada mugavalt igal pool ja ilma internetiühenduseta.

2.2 Ülesande püstitus

Käesoleva töö eesmärgiks on arendada lihtsasti ja mugavalt kasutatav mobiilirakendus Android platvormile, mida saab kasutada eesti keele õigekirja õppimiseks ükskõik kus

parajasti asudes. Rakendus töötab nii, et kõigepealt ilmuvad ekraanile valikud, mis sorti õigekirja kasutaja õppida tahab. Peale valiku tegemist tekib juhuslik sõna, mis võetakse valitud sõnastikust, ekraanile ja kasutaja peab hakkama arvama, kas sõna on õigesti või valesti kirjutatud. Peale igat valikut annab rakendus tagasisidet, et kasutaja saaks koheselt oma vigadest õppida. Mobiilirakendus peab olema lihtsasti kasutatav ja kaasahaarav ning töötama ka ilma internetiühenduseta. Eelnevalt mainitul on kaks positiivset külge – esiteks saab rakendust kasutada ilma segavate faktoriteta ja teiseks on alati võimalus rakendust kasutada vaatamata interneti olemasolule. Rakendust on kasutajal võimalik kasutada ilma sisse logimata ja ilma õpitava sõnavara loomiseta, kuna rakenduses on olemas vaikimisi sisseehitatud sõnavara. Valmis rakendus peab sisaldama võõrsõnade õigekirja, kokku- ja lahkukirjutamise ning suure ja väikese algustähe õppimise võimalusi. Kasutaja peab saama vajadusel oma sõnavara luua või siis lisada vaikimisi sõnavarasse uusi sõnu või neid kustutada. Kasutaja peab saama luua privaatse ruumi ja liituda privaatse ruumiga ning seal oma tulemuse edetabelisse salvestada. Selleks peab rakenduses olema võimalus kasutajat autentida. Autentimist on vaja selleks, et kasutaja saaks andmeid pilve salvestada ja salvestatud andmeid pilvest alla laadida.

2.3 Rakenduse nõuded

Siin toob autor välja kõik rakenduse funktsionaalsed ja mittefunktsionaalsed nõuded.

2.3.1 Rakenduse funktsionaalsed nõuded

- Kasutaja peab saama rakenduses sisse logida.
- Kasutajal peab olema valik, mis õigekirja kategooriat ta õppida tahab.
- Uue sõnastiku loomise võimalus.
- Uute sõnade sõnastikku lisamise võimalus.
- Sõnade eemaldamise võimalus sõnastikust.
- Andmete pilve salvestamine.
- Privaatse ruumi loomise ja liitumise võimalus.
- Privaatse ruumi edetabelisse tulemuse salvestamine.

2.3.2 Rakenduse mittefunktsionaalsed nõuded

- Rakenduse põhifunktsionaalsuse kasutamine ilma internetita.
- Rakendus peab olema intuitiivse disainiga.
- Rakendus peab olema eesti keeles.
- Rakendus peab töötama Android operatsioonisüsteemil.

2.3.3 Rakenduse vaikimisi sõnavara nõuded

- Sõnavaras on vähemalt 100 sõna.
- Sõnad peavad põhinema erinevatele õigekirja reeglitele.
- Sõnad peavad olema ühtepidi mõistetavad.
- Sõnad on päriselus aktuaalsed.
- Sõna vale variant peab olema õigega sarnane.

3 Metoodika

Autor kirjeldab antud peatükis töö sooritamiseks valitud meetodeid ning nende alternatiive.

3.1 Tehnoloogilised valikud

Mobiilirakenduse arendamiseks sai valitud Flutteri [1] raamistik. Flutter on Google'i loodud avatud lähtekoodiga kasutajaliidese tarkvaraarenduskomplekt. Seda kasutatakse platvormidevaheliste rakenduste arendamiseks Androidile, iOS-ile, Linuxile, Macile, Windowsile ja veebile ühest koodibaasist. Flutter sai valitud sellepärast, et see on kiirem kui alternatiivid ning annab võimalused kasutada autentimist ja väga lihtsasti pilve salvestamist Firebase platvormi abil. Tänu Flutteri kihilisele arhitektuurile saab kergemini luua ilusa disainiga mobiilirakendusi. Lisaks on Flutter kiire, sest see kasutab riistvara kiirendamise 2D graafilist liidest Skia [2], mis toetab ka Chrome'i ja Androidi.

3.2 Raamistiku alternatiivid

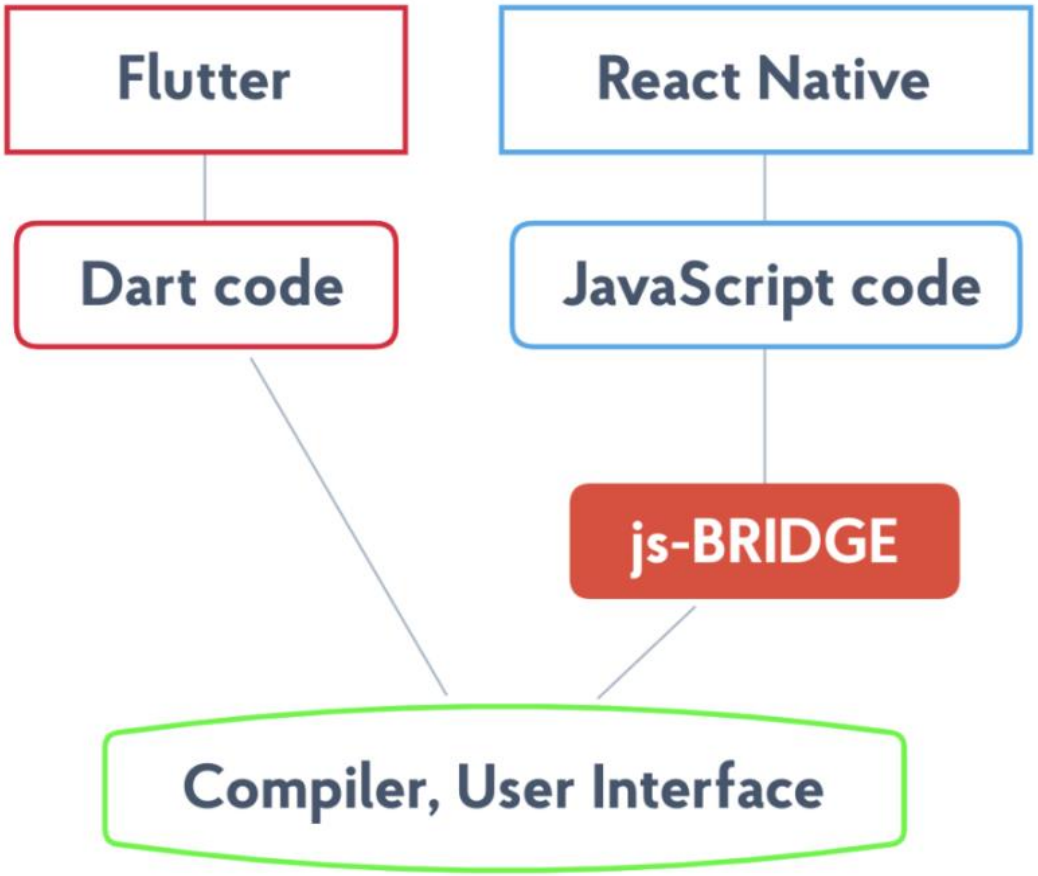
Mobiilirakenduse tõhusaks arenduseks on vaja kasutada õigeid tööriistu ja tarkvaraarenduskomplekte. Kõige tähtsamaks tööriistaks peetakse tihtipeale mobiiliarenduse raamistikku. Modernsed mobiilirakenduste arendamise raamistikud toetavad konkreetses keskkonnas rakenduste väljatöötamist. See võimaldab arendajal kasutada palju erinevaid valmis teeke, et arendaja ei peaks täiesti nullist alustama. Selle tõttu pakub raamistik meile palju sisseehitatud eeliseid nagu kiire ja tõhus arendusprotsess ja koodi taaskasutus. [3]

Autor otsustas võrreldavaks raamistikuks valida React Native [4], sest see on arendajate seas väga populaarne valik.

3.2.1 Jõudlus

Väga tähtis aspekt, mida tuleb vaadata raamistikku valides on selle kiirus. Lähtekoodi käivitamine React Native raamistikku kasutades vajab „silla" loomist JavaScript koodi ja seadme kohaliku keskkonna vahel, et need kaks erinevat keskkonda saaksid omavahel

informatsiooni vahetada. „Sild“ kahe keskkonna vahel on põhimõtteliselt interpretaator, mis tõlgib JavaScripti koodi seadme kohalikuks programmeerimiskeeleks ja ka vastupidi.



Joonis 1. Raamistike suhtlus kompilaatoriga

See „sild“ funktsioneerib lisa puhvrina, mis teeb kogu rakendust aeglasemaks, sest iga lisa komponent vajab aega ja ressursse, et funktsioneerida. Selle tõttu tekib React Native rakendustes tavaliselt pudelikaela efekt. Flutter kompileerib aga x86 ja ARM kohalikkudeks teekideks ilma vahekihtideta ning selle tõttu tegutseb kiiremini ja kasutab vähem ressursse koodi käivitamiseks. Lisaks sellele kasutab Flutteri raamistik C++ graafika renderdusmootorit, mis kujutab pildi otse ekraanile. Tänu sellele on animatsioonid kiiremad ja sujuvamad ning vajavad vähem koodi ridu. See ei tähenda, et React Native nende asjadega hakkama ei saa, lihtsalt Flutter teeb neid mobiiliseadmetes kiiremini ja tõhusamalt. Eelnevalt mainitu tõttu on Flutter mobiiliseadmetes 2-15 korda kiirem (olenevalt testidest) kui React Native. Kuigi siin on tegemist kõigest testidega, siis kiiruse vahe on ka päris rakendustes keskmiselt samasugune. [5]

3.2.2 Disain

Mõlemal raamistikul on suurepärased graafilised funktsioonid, aga need kasutavad täiesti erinevaid tehnikaid kasutajaliidese kujutamiseks. React Native rakendus näeb Androidi ja iOS platvormide peal natuke erinev välja, aga Flutteri raamistikku kasutades näeb rakendus kõigi seadete peal samasugune välja, vaatamata seadme operatsioonisüsteemi versioonile või seadme mudelile.

3.2.3 Dokumentatsioon ja kogukond

Google on tuntud oma detailsete ja hästi struktureeritud dokumentatsioonide poolest. Lisaks tavalisele dokumentatsioonile on arendajal võimalus vaadata õpetlikke videosid, mis on loodud Google tiimi poolt või teha praktilisi harjutusi Codelabsis. Eelnevalt mainitud ressursid on kõik pakutud ametlikult ning lisaks neile leiab veel sadu kursuseid Udemy, Udacity ja muudelt veebilehtedelt.

Kuigi Flutter on küllaltki uus raamistik, ei ole sellel puudust kogukonna toetusest, sest see kogub väga kiiresti populaarsust. 2020. aasta seisuga näitas Stack Overflowi uuring, et Flutter oli raamistike populaarsuse seas kolmandal kohal ning React Native kümnendal kohal [6]. 2021. aasta seisuga oli Flutter tõusnud teisele kohale ning React Native üheksandale [7]. Flutteri kogukond on väga suur erinevates sotsiaalvõrgustikes nagu Medium, Stack Overflow, Discord, Reddit. Lisaks on Darti programmeerimiskeele kogukonnad ka väga populaarsed. [8]

3.3 Flutter ja Dart

Flutter on avatud lähtekoodiga tarkvara, mis on loodud Google poolt *native* rakenduste loomiseks Android ja iOS platvormidele. Flutter on suhteliselt uus, sest see esitati ametlikult 2018. aasta detsembris. Kuna native rakendused on ehitatud kindla seadme operatisoonisüsteemi peale, siis rakendusel on võimalik kasutada seadmele spetsiifilist riistvara ja tarkvara. [9]

Flutter lihtsustab rakenduste arendamist ja samal ajal pakub native rakendustega samaväärset jõudlust. Flutteri raamistik annab meile võimaluse säilitada visuaalse sarnasuse sõltuvata platvormist, millele rakendust arendatakse. Dart programmerimiskeel oli originaalselt mõeldud JavaScripti asenduseks. Flutter on avatud lähtekoodiga ja täiesti tasuta kasutatav. Flutter on arendajate seas suhteliselt sama populaarne GitHubis ja Stack

Overflowis võrreldes vanemate raamistikega nagu React Native. Lisaks sellele on Google Play Store'is juba üle 50000 Flutteri rakenduse ja see number ainult kasvab. Alibaba Group, eBay, Groupon ja muud populaarsed elektroonilise kaubanduse ettevõtted kasutavad Flutterit, et pakkuda kasutajale samaväärne kasutajakogemus olenevalt hetkel kasutatavast platvormist. Flutteri raamistik, mis on kirjutatud Dart programmeerimiskeeles, kasutab Flutteri mootorit ja widgeteid. Widget on taaskasutatav jupp koodi, mis kirjeldab, kuidas peaks rakenduse kasutajaliides välja nägema. Flutteri rakenduse arendusprotsess erinebki teistest oma deklaratiivse kasutajaliidese kirjutamise tõttu. Siin on vajadus alustada lõpust, mis tähendab seda, et enne kasutajaliidese arendamist peab arendaja teadma, milline see kasutajaliides välja peaks nägema. Enamik arendajaid peavad sellist sorti kasutajaliidese loomist lihtsamaks, aga see võib alguses keeruline olla. Flutteri põhiidee on see, et arendajad saavad luua tervikliku kasutajaliidese lihtsalt erinevaid widgeteid kombineerides. Rakenduse liides koosneb erinevatest pesastatud widgetitest, mis võivad olla ükskõik mis objektid. Objekt võib olla nupp, tekst, *padding*, konteiner. Widgeteid kombineerides saab arendaja rakendust lihtsasti kohandada. Widgetid saavad üksteist mõjutada ja kasutada sisse-ehitatud funktsioone, et reageerida rakenduse oleku välistelemuutustele. Widgetid on väga olulised elemendid kasutajaliideses ja vastavad Androidi, iOS ja veebirakenduste disaini nõuetele. [10]

3.4 Firebase

Firebase on Google'i mobiilirakenduste arendamise platform, mis aitab arendajal oma rakendust luua, pakkudes arendajale erinevaid teenuseid.

3.4.1 Firebase Authentication

Kasutaja autentimiseks on vaja kõigepealt tema kredentsiaale. Kredentsiaalideks võivad olla kasutaja meiliaadress ja parool või siis OAuth *token* mõne födereeritud identiteedi pakkuja käest nagu näiteks Google Identity. Kredentsiaalid saab edasi saata Firebase Authenication SDK-le ja selle tagarakenduse teenused kinnitavad kredentsiaalid ning saadavad kasutajale vastuse. Pärast edukat sisselogimist saab ligipääsu kasutaja üldisele informatsioonile ja arendaja saab ligipääsu andmetele, mis on salvestatud muudesse Firebase teenustele. Antud kontekstis on see väga kasulik, sest autoril on vaja saada ligipääsu Firebase'i Cloud Firestore andmetele.

Enamik ajast on vaja teada parasjagu rakendust kasutava isiku identiteeti. Kasutaja identiteeti teades on võimalik rakendusel kasutaja andmeid pilve turvaliselt salvestada ja pakkuda talle sama personaliseeritud kasutajakogemust erinevate kasutaja seadmete peal.

Firebase Authenication pakub *backend* teenuseid, kergesti kasutatavaid tarkvaraarenduskomplekte ja UI komplekte, et kasutajaid rakenduses autentida. Firebase Authentication toetab autentimist erinevatel viisidel: telefoni number, Google, Facebook, Twitter, kasutajanimi ja parool.

Firebase Authentication on tihendalt integreeritud muude Firebase teenustega ja see kasutab tööstusstandardeid nagu OAuth 2.0 ja OpenID Connect ning on tänu sellele lihtsasti integreeritav ka kohandatud tagarakendusega. [11]

3.4.2 Firebase Cloud Firestore

Cloud Firestore on paindlik, lihtsasti skaleeritav andmebaas mobiili, veebi ja serveri arenduseks, mis on loodud koostöös Firebase ja Google Cloudi vahel. Cloud Firestore hoiab andmeid sünkroonis kõikide kasutajate rakenduste vahel, mis võimaldab luua andmetele reaalajas reageerivaid mobiilirakendusi.

Cloud Firestore on NoSQL andmebaas, millele saab mobiilirakendus ligi otseselt läbi kohalike SDK-de. Cloud Firestore NoSQL andmete mudel on selline, et andmed salvestatakse dokumentidesse, kus on väljad, millel on väärtused. Need dokumendid on omakorda salvestatud kollektsioonidesse, mis on konteinerid dokumentide jaoks, mida saab kasutada andmete organiseerimiseks ja päringute tegemiseks. Dokumendid toetavad erinevaid andmetüüpe, lihtsatest sõnedest kuni komplekssete objektideni.

Päringute tegemine on Cloud Firestores tõhus ja paindlik. Võimalik on teha pealiskaudsemaid päringuid dokumendi tasemel ilma tervet kollektsiooni pärimata, et ressursse säästa. [12]

3.5 Arenduskeskkonna seadistamine

Arenduskeskkonna korrektse töötamise jaoks on vaja installida järgmised vahendid:

- Visual Studio Code 1.62 [13]
- Gradle 4.1 [14]
- Flutter 2.14.3

Firebase Authenticationi ja Firebase Cloud Firestore andmebaasi oma rakenduses kasutamiseks tuleb kõigepealt oma Flutteri projekt veebis Firebase arendaja konsooli [15] lisada. Firebase oma projektile lisades saab veebist konfiguratsioonifaili *google-services.json*, mille tuleb lisada oma projekti kausta. Konfiguratsioonifaili kasutamiseks on vaja *build.gradle* failidesse lisada mõned read, et Grade saaks konfiguratsioonifaili projekti sisse laadida.

Project-level build.gradle (<project>/build.gradle):

```
buildscript {
  repositories {
    // Check that you have the following line (if not, add it):
    google()  // Google's Maven repository
  }
  dependencies {
    ...
    // Add this line
    classpath 'com.google.gms:google-services:4.3.10'
  }
}

allprojects {
  ...
  repositories {
    // Check that you have the following line (if not, add it):
    google()  // Google's Maven repository
    ...
  }
}
```

Joonis 2. build.gradle faili seadistamine projekti tasemel.

App-level build.gradle (<project>/<app-module>/build.gradle):

```
apply plugin: 'com.android.application'
// Add this line
apply plugin: 'com.google.gms.google-services'

dependencies {
  // Import the Firebase BoM
  implementation platform('com.google.firebase:firebase-bom:29.0.1')

  // Add the dependencies for the desired Firebase products
  // https://firebase.google.com/docs/android/setup#available-libraries
}
```

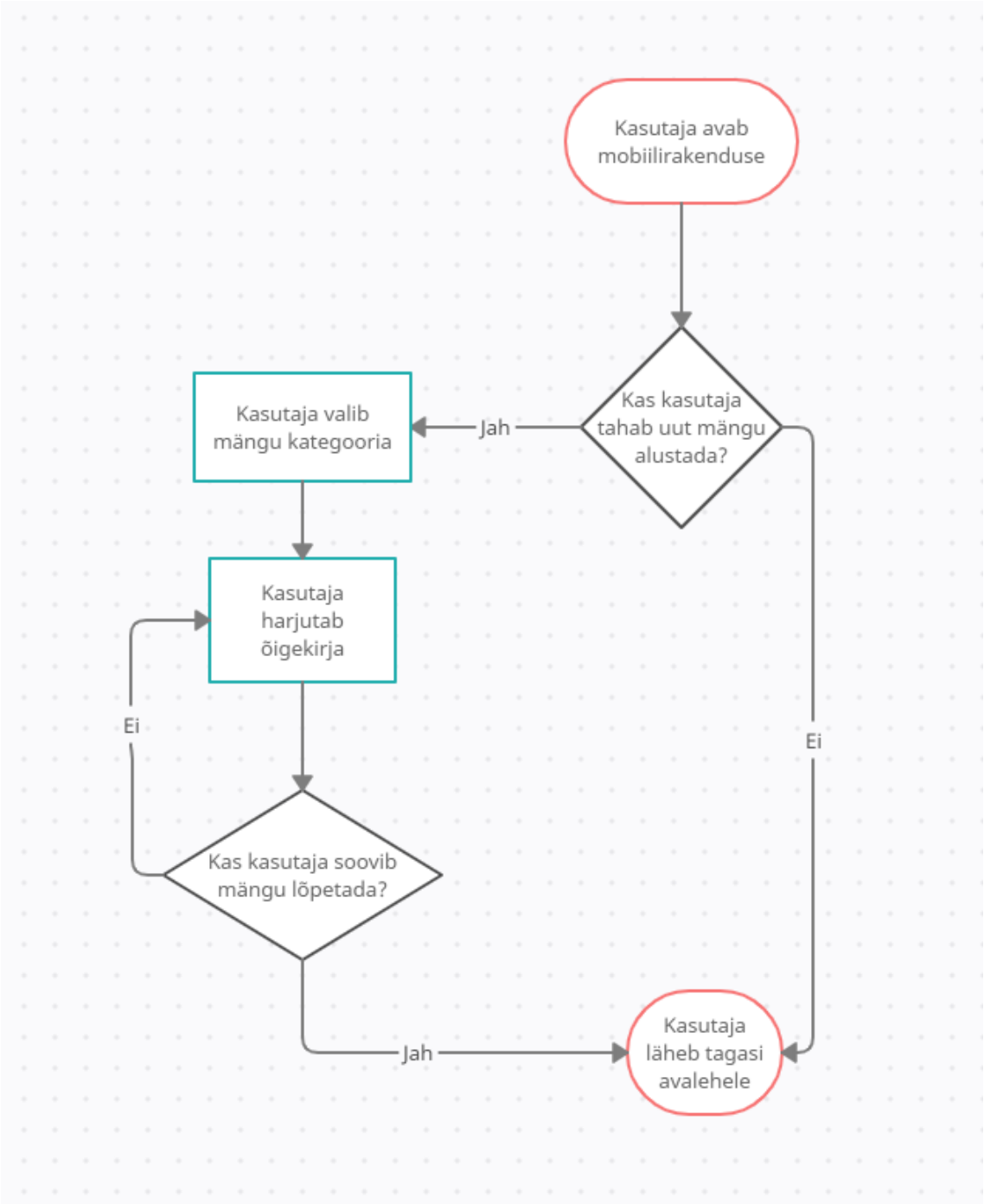
Joonis 3. build.gradle faili seadistamine rakenduse tasemel.

Peale oma rakenduse lisamist Firebase'i ja konfiguratsioonifaili seadistamist saab kasutada erinevaid Firebase poolt pakutavaid võimalusi nagu Firebase Authentication ja Firebase Firestore.

3.6 Rakenduse arenduse lihtsustamiseks koostatud skeemid

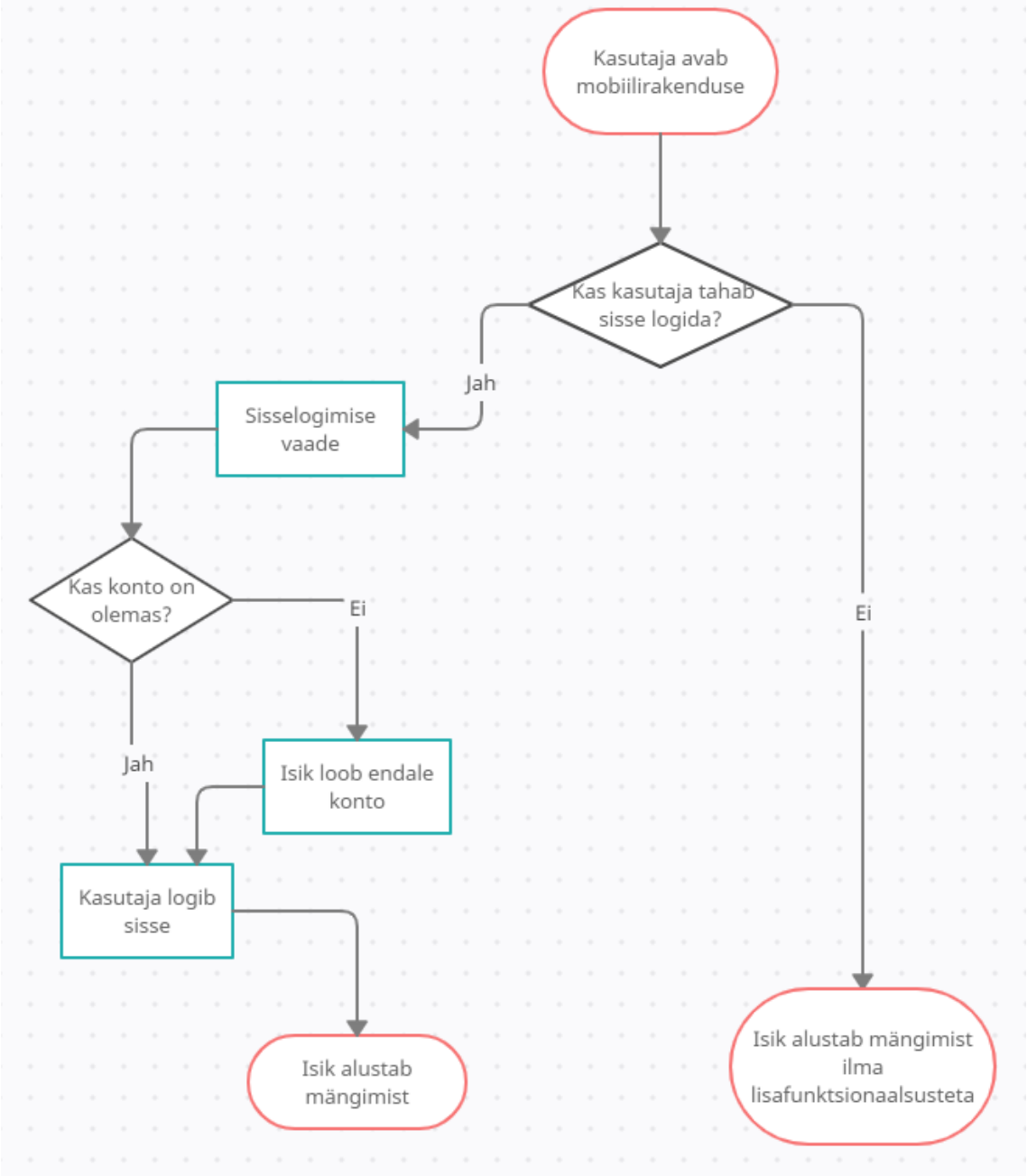
Kasutajaloode diagrammide koostamine on väga kasulik tegevus, et visualiseerida, mida peaks täpselt tegema efektiivse rakenduse loomise jaoks. Fookuseks on kasutaja, seades rõhku protsessi mitmele iteratsioonile ja pidades silmas kasutaja tagasisidet.

Kõige alguses oli vaja luua mobiilirakenduse alus ilma igasuguste lisafunktsionaalsusteta:



Joonis 4. Rakenduse vooskeem lisafunktsionaalsuseta.

Siis lisas autor rakendusele kasutaja autentimise võimaluse:

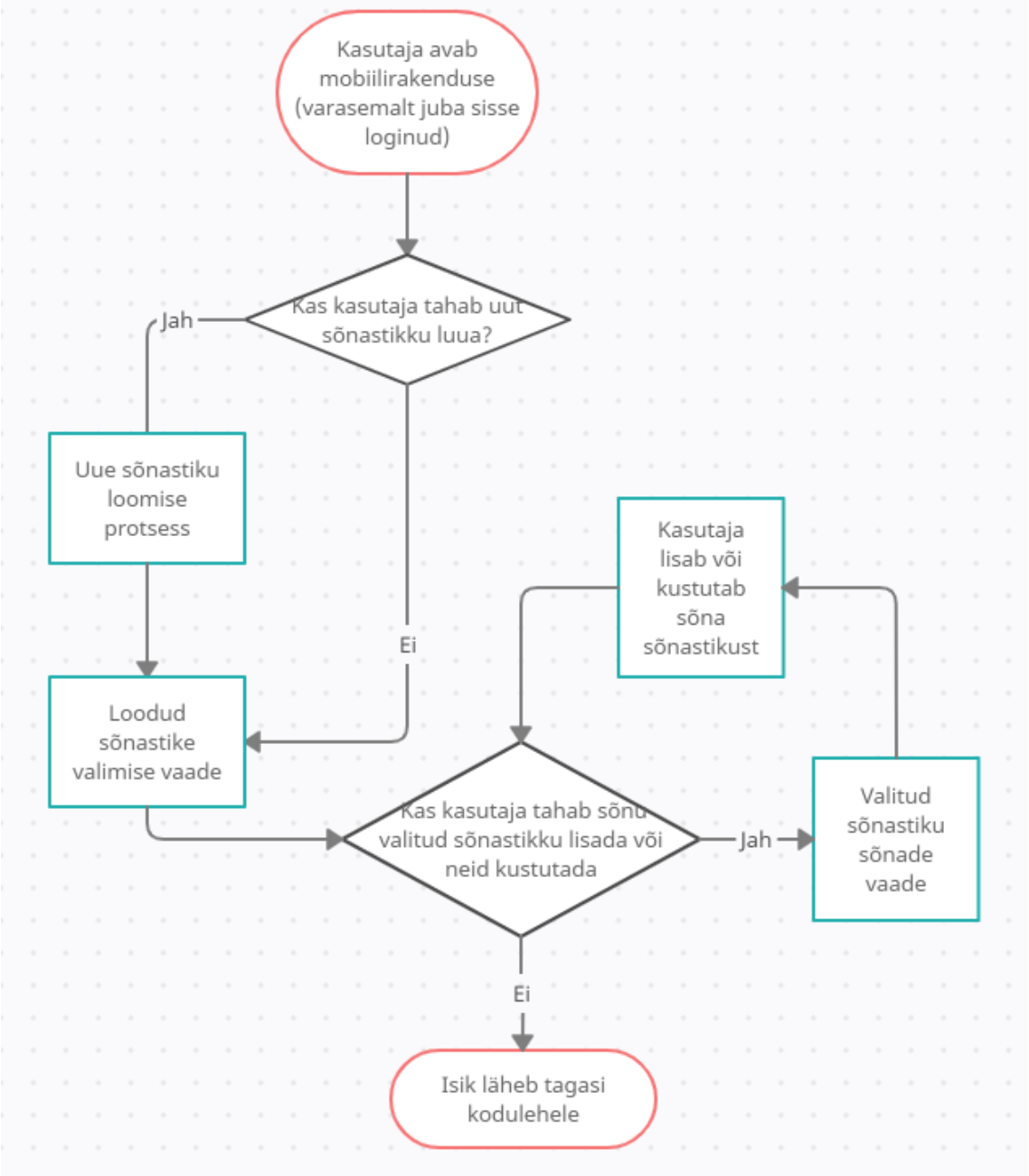


Joonis 5. Kasutaja autentimise vooskeem.

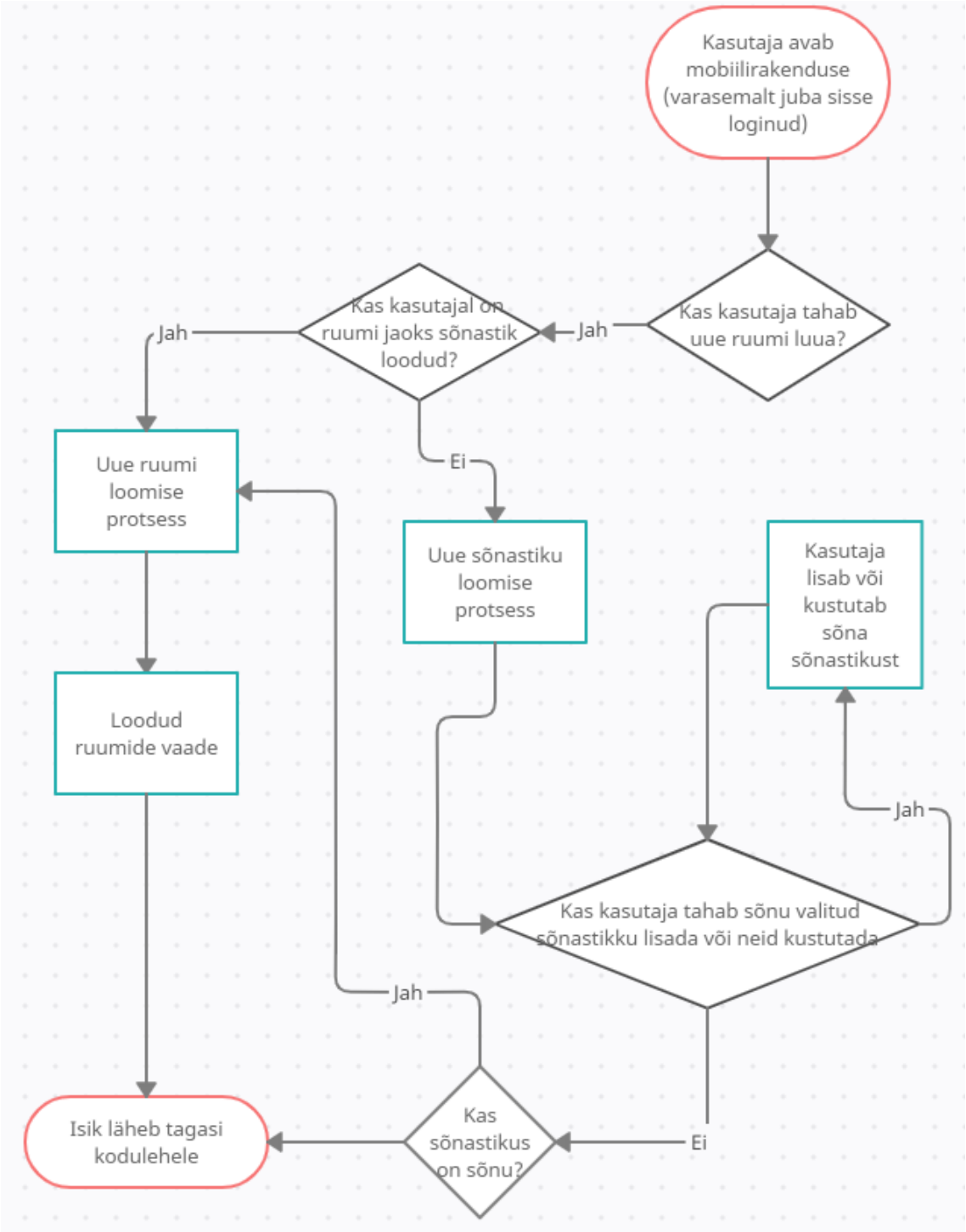
Kasutaja autentimist on vaja selleks, et kasutaja saaks kasutada rakenduse lisafunktsionaalsust nagu andmete pilve salvestamine, privaatsete ruumide loomine ja kohandatud sõnavara loomine.

Nüüd saab mõelda, kuidas peaks välja nägema kohandatud sõnavara loomine ja peale seda privaatse ruumi loomine, sest privaatse ruumi loomiseks on vaja valida sõnastik,

mida ruumis olles kasutama hakatakse:



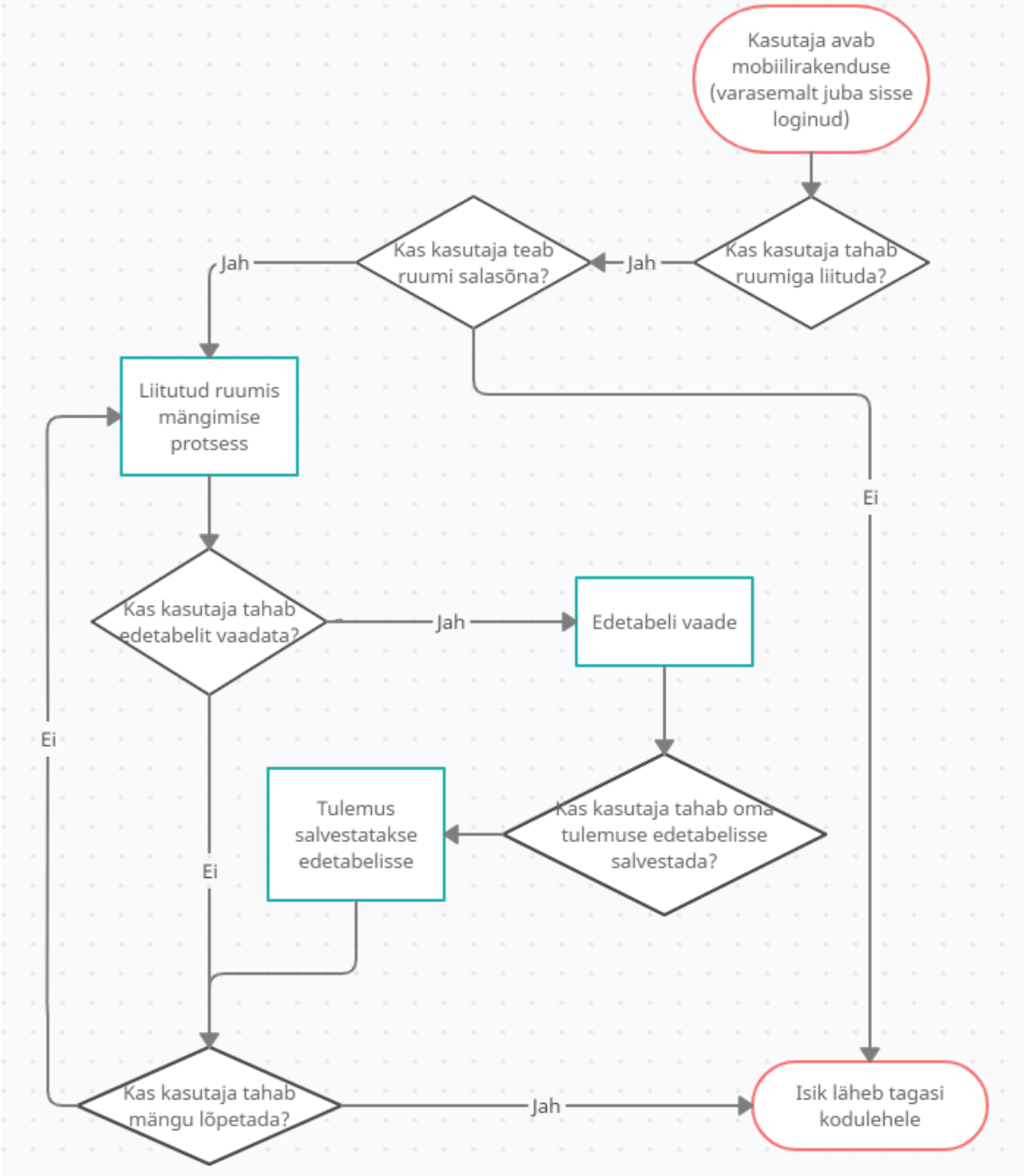
Joonis 6. Kohandatud sõnastiku loomise ja sõnade lisamise/kustutamise vooskeem.



Joonis 7. Privaatse ruumi loomise vooskeem

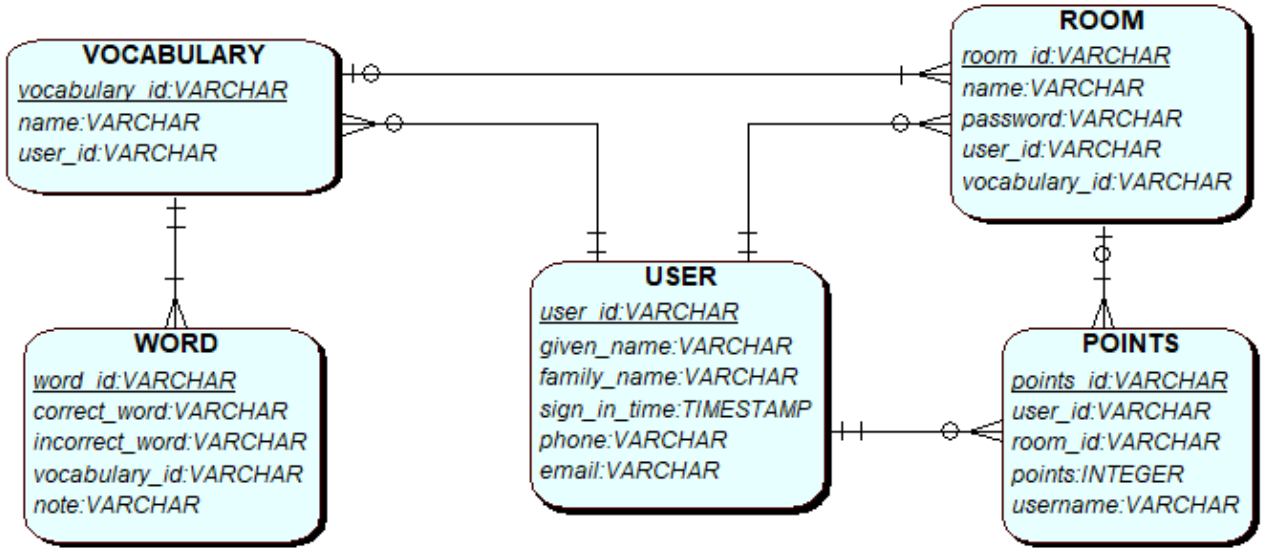
Siin skeemil on välja toodud, kuidas peab välja nägema privaatse ruumiga liitumine, seal

mängimine, punktide salvestamine ning edetabeli vaatamine:



Joonis 8. Privaatse ruumiga liitumise, tulemuse salvestamise edetabelisse vooskeem.

3.6.1 Andmebaasi relatsiooniline mudel



Joonis 9. Andmebaasi relatsiooniline mudel.

4 Testimine

Rakenduse testimine on tähtis ja kasulik osa arendusprotsessist, mis aitab veenduda koodi korrektsuses.

Testimine annab arendajale teada, mis osad rakendusest töötavad hästi ja millised vajavad parandamist. Kõige tähtsam on rakendust testida erinevate seadmete peal, kuna need kasutavad erinevaid operatsioonisüsteemide versioone ning see võib tuua kaasa erinevaid probleeme.

Kui rakenduse algne versioon sai valmis, andsin seda oma pereliikmetele ja sõpradele proovida ning küsisin, mida võiks paremini teha ja mis olid rakenduse puudujäägid. Sellist sorti testimist nimetatakse manuaalseks testimiseks ja seda on vaja, et leida vigu enamjaolt kasutajakogemuse valdkonnas. Peale tagasiside saamist tegin rakenduses parandusi ja siis seda palusin uuesti testida. Alguses ilmnes palju erinevaid vigu ja probleeme, aga korduvatel testimistel jäi vigu järjest vähemaks.

Sain palju soovitusi edaspidiseks arenduseks ning autoril on endal ka mõned ideed olemas.

4.1 Edaspidine arendus

- Kohandatud sõnastike ja nendesse lisatud sõnade kustutamine võiks toimuda ilma navigatsiooni muutmiseta ja kustutada võiks saada mitu rida korraga.
- Rakendus võiks töötada ka iOS operatisioonisüsteemil, et Apple kasutajad saaksid ka rakendust kasutada.
- Privaatse ruumi edetabelisse saab salvestada mitu tulemust, aga võiks saada salvestada ainult parima tulemuse.
- Rakenduse disain võiks olla ilusam.
- Arendada rakenduse turvalisust.

5 Kokkuvõte

Lõputöö eesmärgiks oli luua mobiilirakendus, mis võimaldab Eesti keele õigekirja lihtsasti ja mugavalt õppida.

Lõputöö käigus pandi paika rakenduse funktsionaalsed ja mittefunktsionaalsed nõuded ning arendati välja nõuetekohane mobiilirakendus, mida saab kasutada ka teiste keelte õppimiseks.

Projekti arendades õppisin, kuidas luua iseseisvalt *full-stack* mobiilirakendus algusest lõpuni. Kasutasin ära saadaval olevaid ja kasulikke lahendusi, mida Flutteri raamistik ja Firebase mobiilirakenduste arendamise platvorm pakub.

Lõputöö käigus valminud mobiilirakendus ei ole kindlasti perfektne, aga sellele vaatamata on kõik välja toodud nõuded täidetud ning kuna töötav rakendus on edukalt valminud, siis on sinna lihtsam uut funktsionaalsust lisada.

Kasutatud kirjandus

[1] Flutter, [Võrgumaterjal]. Available: https://github.com/flutter/flutter. [Kasutatud 1 11 2021].

[2] „Skia: The 2D Graphics Library,“ [Võrgumaterjal]. Available: https://skia.org/. [Kasutatud 1 11 2021].

[3] Explorate Global, „Explorate Global,“ [Võrgumaterjal]. Available: https://www.explorateglobal.com/blog/choose-mobile-app-development-framework/. [Kasutatud 3 12 2021].

[4] Facebook, „React Native,“ [Võrgumaterjal]. Available: https://reactnative.dev/.

[5] NIX United, „NIX United,“ [Võrgumaterjal]. Available: https://nix-united.com/blog/flutter-vs-react-native/. [Kasutatud 3 12 2021].

[6] Stack Overflow, „2020 Developer Survey,“ [Võrgumaterjal]. Available: https://insights.stackoverflow.com/survey/2020. [Kasutatud 4 1 2022].

[7] Stack Overflow, „2021 Developer Survey,“ [Võrgumaterjal]. Available: https://insights.stackoverflow.com/survey/2021. [Kasutatud 4 1 2022].

[8] Altexsoft, „Altexsoft,“ [Võrgumaterjal]. Available: https://www.altexsoft.com/blog/engineering/pros-and-cons-of-flutter-app-development/. [Kasutatud 2 1 2022].

[9] A. S. Gillis, „TechTarget,“ [Võrgumaterjal]. Available: https://searchsoftwarequality.techtarget.com/definition/native-application-native-app. [Kasutatud 20 12 2021].

[10] NIX United, „NIX United,“ NIX United, [Võrgumaterjal]. Available: https://nix-united.com/blog/the-pros-and-cons-of-flutter-in-mobile-application-development/. [Kasutatud 20 12 2021].

[11] Google, „Firebase Authentication,“ [Võrgumaterjal]. Available: https://firebase.google.com/docs/auth. [Kasutatud 2 11 2021].

[12] Firebase, „Firebase Documentation,“ [Võrgumaterjal]. Available: https://firebase.google.com/docs/firestore. [Kasutatud 3 12 2021].

[13] Microsoft, „Visual Studio Code,“ [Võrgumaterjal]. Available: https://code.visualstudio.com/. [Kasutatud 2 1 2022].

[14] Gradle Inc., „Gradle Build Tool,“ [Võrgumaterjal]. Available: https://gradle.org/. [Kasutatud 2 1 2022].

[15] Firebase, „Firebase Console,“ [Võrgumaterjal]. Available: https://console.firebase.google.com/u/0/. [Kasutatud 3 12 2021].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

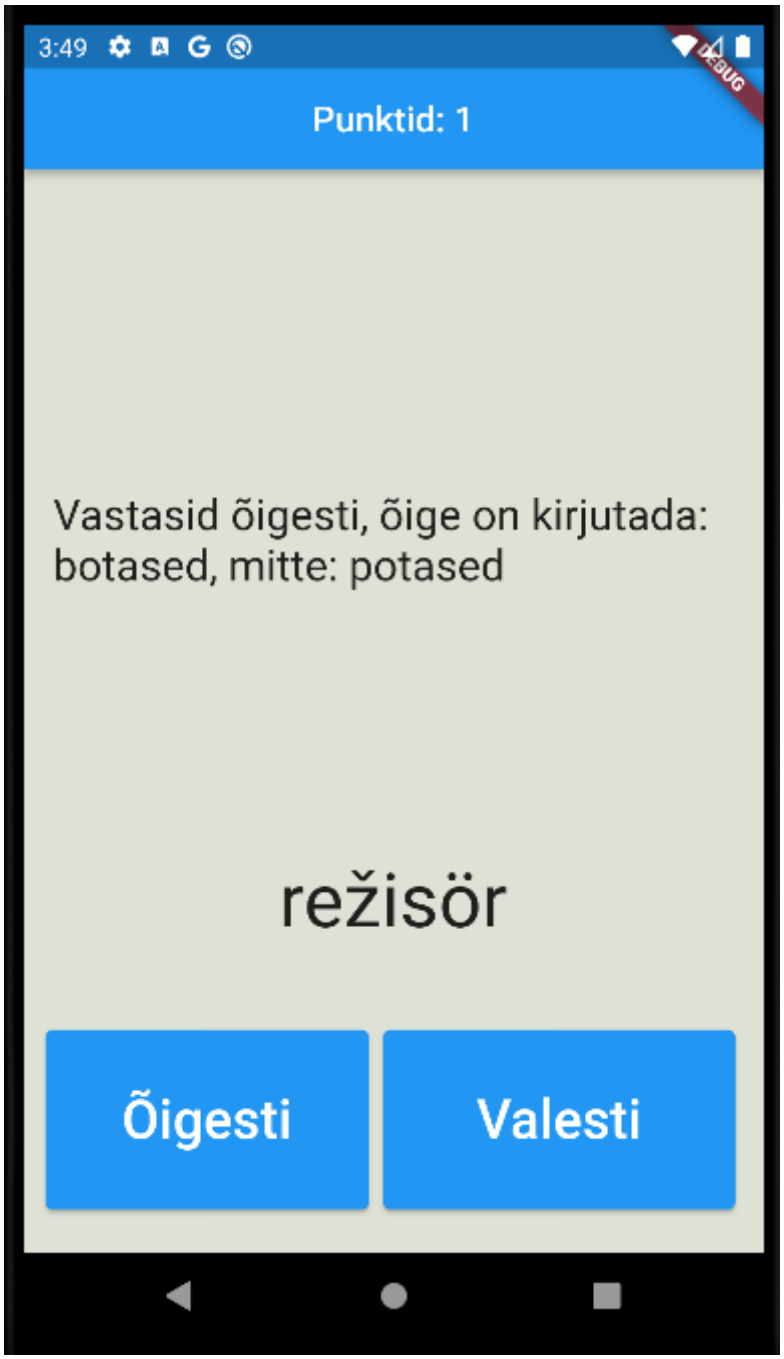
Mina, Kristjan Klimušev

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Eesti keele õigekirja õppimise rakenduse arendamine Android platvormile", mille juhendaja on Aleksei Talisainen
   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.

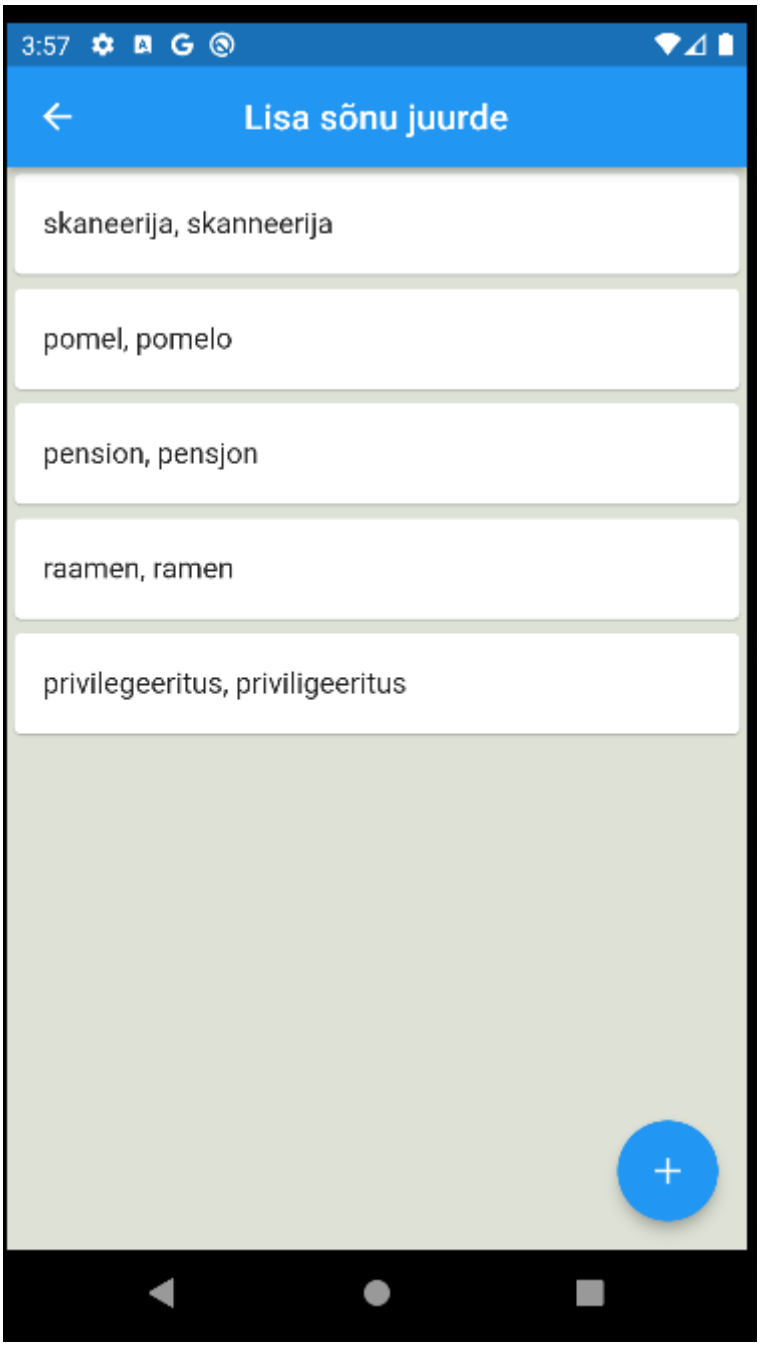
6.01.2022

1 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 loomingulise 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.

Lisa 2 – Valminud rakendus ja selle mõned vaated



Joonis 10. Vaikimisi sõnastikuga mängu vaade.



Joonis 11. Valitud sõnastiku sõnade vaade.

Joonis 12. Näidis kohandatud sõnadega mängust ja lisatud sõna märkmest.

Lisa 3 – Mobiilirakenduse allalaadimise link

https://tinyurl.com/2p8by47e