

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Hans Joosep Lahe 143075IAPB

MAASTIKUMÄNGE TOETAV MOBIILIRAKENDUS

Bakalaureusetöö

Juhendaja: Tanel Tammet
PhD

Tallinn 2017

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: Hans Joosep Lahe

22.05.2017

Annotatsioon

Nutitelefoni populaarsuse tõusuga on avanenud uued võimalused virtuaalne maailm siduda reaalse maailmaga. Käesoleva bakalaureusetöö raames on loodud maastikumänge toetav mobiilirakendus Androidi platvormile. Androidi rakenduse ülesehitamisel lähtuti MVP arhitektuuri põhimõtetest. Mobiilirakenduse andmete haldamiseks ning mängu organiseerimiseks on loodud ka esialgne veebirakendus. Serveri ja kliendi vaheliseks suhtluseks kasutatakse RESTful teenust. Töö esimeses pooles uuritakse olemasolevaid lahendusi ja koostatakse funktsionaalsed nõuded pakutavale rakendusele. Teises pooles antakse ülevaade üldisest arhitektuurisest lahendusest ja kirjeldatakse detailsemalt mobiilirakenduse disaini ning ülesehitust. Tulemuseks on Androidi mobiilirakendus, mis kombineerib orienteerumise ja mõistatuste lahendamise elemendid.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 28 leheküljel, 4 peatükki, 13 joonist, 1 tabelit.

Abstract

Mobile application to support treasure hunts

With the rise of the smartphone new possibilities to connect the virtual world with the real one have risen. The purpose of the thesis was to create a mobile application with the intention to support treasure hunts and encourage people to explore the places around them. The operating system of choice for the application was Android due to the author's experience with the platform. The Android application followed the MVP architecture blueprint and used the Package By Feature package structure in order to ease the management of the source code. The thesis also involved the development of a web application with the purpose to mediate dataflow between the database and the Android application. The communication follows REST principles. The first part of the thesis analyzed existing solutions and compared them to the solution proposed by the author. The result of the thesis is an Android application that combines elements of puzzle-solving with orienteering, source code is publicly available at <https://Hans.Lahe@gitlab.pld.ttu.ee/Hans.Lahe/thesis.git> .

The thesis is in Estonian and contains 28 pages of text, 4 chapters, 13 figures, 1 table.

Lühendite ja mõistete sõnastik

<uri>	Serveri ip aadress või domeeni nimi.
<i>Activity</i>	Androidi API klass, mis vastutab kasutajaliidese kuvamise eest kasutajale.
Android	Android on tarkvarakomplekt elektroonikaseadmetele, mis hõlmab operatsioonisüsteemi, vahetarkvara ja peamisi rakendusi.
API	<i>Application Programming Interface</i> ehk rakendusliides, mis on disainitud konkreetsete reeglite järgi suhtlema teiste programmidega, võimaldades kolmandatel osapooltel kasutada valmis lahendusi
Django	Tasuta ja avatud lähtekoodiga Pythonis programmeerimiskeeles kirjutatud veebirakenduste arendamiseks mõeldud raamistik.
<i>Fragment</i>	Androidi API klass, mis vastutab ühe tükki kasutajaliidese kuvamise eest kasutajale.
GPS	<i>Global Positioning System</i> ehk sateliitnavigatsiooni süsteem.
GSON	Tuntud ka kui <i>Google Gson</i> , GSON on avatud lähtekoodiga Java teek JSONi dekodeerimisega Java objektideks ja vastupidi.
HTTP	<i>Hypertext Transfer Protocol</i> ehk hüperteksti edastusprotokoll on andmevahetuse protokoll.
iOS	Apple poolt loodud operatsioonisüsteem mobiilsetele seadmetele.
Jackson	Avatud lähtekoodiga Java teek JSONi dekodeerimisega Java objektideks ja vastupidi.
LARP	<i>Live Action Role-Playing game</i> ehk rollimäng, milles osalejad püüavad kehastada tegelasi neid füüsilises maailmas välja mängides.
MVC	<i>Model-view-controller</i> on tarkvara arhitektuurimuster, mis jagab tarkvararakenduse kolmeks omavahel seotud osaks.
MVP	<i>Model View Presenter</i> on kasutajaliideste loomise jaoks mõeldud tarkvaraarhitektuuri muster.
NFC	<i>Near Field Communication</i> on kogum kommunikatsiooni protokolle, mis võimaldavad kahel seadmel lühikese (4 cm või vähem) distantisi üle infot vahetada.

ORM	<i>Object-relational mapping</i> on andmete muutmise ühest vormist teise objekt orienteeritud programmeerimiskeeltes.
PBF	<i>Package By Feature</i> on viis, kuidas jagada rakenduse lähtekood osadeks jagamist funktsionaalsuse järgi.
PBL	<i>Package By Layer</i> on viis, kuidas jagada rakenduse lähtekood osadeks jagamist kihtide järgi.
Pokemon Go	Pokemon Go on ettevõtte Niantic poolt arendatud asukohapõhine liitreaalsuse mobiilimäng.
PostgreSQL	PostgreSQL on relatsiooniline andmebaasihaldur mõningase objektorienteeritud kallakuga.
QR-code	<i>Quick Response Code</i> ehk ruutkood on Jaapanis loodud kahemõõtmeline maatrikskood, mis võimaldab skaneerida infot mobiiltelefoniga, kus mobiilirakendus selle dekodeerib.
REST	<i>Representational State Transfer</i> on tarkvaraarhitektuuri stiil, mis koosneb juhenditest ja reeglitest, eesmärgiga koostada skaleeritav veebiteenus.
RxJava	Java teek, mis toob lisa võimalusi reaktiivsete programmeerimise võtete kasutamiseks Javas.
SDK	<i>Software Development Kit</i> ehk arendustarkvara, võimaldab programmeerijal luua rakendusi konkreetsele platvormile.
<i>Selective availability</i>	<i>Selective availability</i> oli avaliku GPS signaalide tahtlik degradeerumine julgeoleku kaalutlustel.

Sisukord

1 Sissejuhatus.....	11
2 Taust.....	12
2.1 TTÜ Meeskonnaprojekt Narrative Trip Platform.....	12
2.1.1 Veebirakendus.....	12
2.1.2 Mobiilirakendus.....	13
2.2 Olemasolevad lahendused.....	14
2.2.1 Go Team - Go hunt.....	15
2.2.2 Goosechase.....	15
2.2.3 Klikaklu.....	16
2.2.4 Scavify.....	16
2.2.5 Actionbound.....	17
2.2.6 Strayboots.....	17
2.2.7 Loquiz.....	18
3 Funktsionaalsed nõuded.....	20
4 Arhitektuur.....	23
4.1 Andmebaas.....	24
4.2 Kliendi suhtlus serveriga.....	25
4.3 Veebirakendus.....	25
5 Mobiilirakenduse kirjeldus.....	26
5.1 Kasutajaliides.....	26
5.1.1 Sisselogimise vaade.....	27
5.1.2 Mängude listivaade.....	28
5.1.3 Mängu tutvustamise vaade.....	28
5.1.4 Algava mängu vaade.....	29
5.1.5 Mängu statistika vaade.....	29
5.1.6 Mängu vaade.....	30
5.2 Rakenduse arhitektuur.....	32

5.3 Rakenduse struktuur.....	35
6 Kokkuvõte.....	38
Kasutatud kirjandus.....	39
Lisa 1 – Narrative Trip Platform andmemudel.....	41
Lisa 2 – API kirjeldus.....	42
Lisa 3 – Administratsiooni paneeli üldvaade.....	48
Lisa 4 – Administratsiooni paneeli piirava ala lisamise vaade.....	49

Jooniste loetelu

Joonis 1: Tegevustediagramm mängu positiivseks läbimiseks.....	22
Joonis 2: Komponentide loogiline vaade.....	23
Joonis 3: Andmemudel.....	24
Joonis 4: Navigatsioonidiagramm.....	26
Joonis 5: Sisselogimise vaade.....	27
Joonis 6: Mängu tutvustamise vaade.....	28
Joonis 7: Algava mängu vaade.....	29
Joonis 8: Punktivaade.....	30
Joonis 9: Teadetevaade.....	31
Joonis 10: Vastusevaade.....	32
Joonis 11: Model View Presenter diagramm.....	33
Joonis 12: Mudeli osa detailne vaade.....	34
Joonis 13: Rakenduse struktuur.....	37

Tabelite loetelu

Tabel 1. Sarnaste lahenduste funktsionaalsuste võrdlus.....	19
---	----

1 Sissejuhatus

Arvutite muutumisega mobiilseks on avanenud head võimalused mängude sidumiseks reaalse asukohtadega maailmas. Esimene laialt levinud asukohapõhine mäng oli 2000. aastal alguse saanud ning siiani populaarne geopeitus [1]. Viimaste aastate jooksul on suurt populaarsust kogunud Ingress¹ ja tema järglane PokemonGo².

Maastikumängud koosnevad asukohtadest, mida on vaja läbida. Asukohaga võib olla seotud ka lahendust vajav ülesanne, mille lahendamine on mängus edasi liikumiseks vajalik. Ülesannete liigid võivad varieeruda küsimusele vastamisest kuni piltide tegemiseni. Enamasti toimuvad need mängud tiimides ja kindlatel aladel. Sellist liiki maastikumängude jaoks ei ole autor hästisobivat ja laialt levinud mobiilirakendust leidnud; täpsemalt peatükis "Olemasolevad lahendused".

Esialgne idee sellist liiki maastikumängude jaoks sobiva rakenduse loomiseks tekkis - autoriga mitte seotult - TTÜ magistrantidel aine Tarkvaraarenduse meeskonnaprojekti raames. Nende esmaseks sihtgrupiks olid peamiselt LARPi ehk rollimängu ürituste korraldajad, kuna info edastamine mängu ajal käib paberilt lugemise või mängujuhile helistamisega.

Käesoleva bakalaureusetöö eesmärgiks on toetada maastikumängude läbiviimist mobiilirakendusega, mis on ette nähtud laiema projekti ühe osana.

Lõputöö teises peatükis annab autor ülevaate sarnastest lahendustest ning kirjeldab käesoleva töö inspiratsiooniks olnud projekti. Kolmandas peatükis koostatakse funktsionaalsed nõuded, sellele järgnev peatükk räägib veebirakenduse ja andmebaasi osast. Viies peatükk keskendub mobiilirakenduse realiseerimisele.

¹ <https://www.ingress.com/>

² <http://www.pokemongo.com/>

2 Taust

Erinevate kultuuri- ja huviasutuste kampaaniaid, töökollektiivide tiimiüritusi ning muid ettevõtmisi, mis viivad inimesed õue liikuma ja panevad ümbritseva keskkonnaga tutvuma, leiavad aset üle maailma regulaarselt. Kuigi konkreetsed reeglid võivad natukene erineda (nt. aja peale, ülesanded punktides, piiratud ala jne) on üldine kontseptsioon sarnane.

2.1 TTÜ Meeskonnaprojekt Narrative Trip Platform

Sellist probleemi märkasid ja esmalt püüdsid sellele ühtset lahendust leida TTÜ magistrandid aine Tarkvaraarenduse meeskonnaprojekt raames. Eesmärgiga luua keskkond LARP mängu korraldajatele ja mängijatele. Kuigi nutitelefonid on leidnud tee pea iga inimese tasku ning lihtsustanud inimeste elu mitmel rindel, on sellest LARPi harrastajad mängides vähe kasu tunda saanud. Hetkel leiab nutitelefon harrastajate hulgas kasutust, kui mängujuht helistab või sõnumi saadab. Siin nägid tudengid head võimalust LARPi entusiastidele keskkonna loomise näol. Kavandati teha veebiliides mängu loomiseks ja jälgimiseks ning Androidi mobiilirakendus mängijatele. [2] [3]

2.1.1 Veebirakendus

Veebirakenduse jaoks seati ette 4 funktsionaalset nõuet [2] :

1. *Mängu looja saab lisada, muuta ja kustutada legendi.*
2. *Mängu looja saab lisada, muuta ja kustutada mängu. Mängu loomise ja muutmise käigus saab kasutaja määrata ja kustutada kontrollpunkte, siduda ja lahti siduda nendega varem loodud legende (muuta legende?).*
3. *Mängujuht saab alustada mängu, aktiveerida ja deaktiveerida mängijaid.*

- 4. Mängujuht saab kogu mängu aja informatsiooni mängijate paiknemisest ja pärast mängu lõppu ka aruande mängu ajal toimunud (jäi 3. iteratsiooni skoobist välja).*

Esimesed kolm nõuet saadi veebiliidese puhul edukalt täidetud. Legendide lisamine hõlmas lehel kuvatud kaardile klikkimist ning seejärel tekstiaknas legendile nime andmist. Sarnaselt sai ka legendi muuta. Mängu lisamine, muutmine ja kustutamine tähendas vastavale nupule vajutamist. Mängude alustamine oli selge - mängu lehel kuvati suur vastava tekstiga nupp, mida vajutades mäng algas. Lisaks on võimalik lehe keelt muuta. Samuti on sisselogimine mugav kasutades Facebooki või Google'i kontot.

Lehel on ka puudujääke. Keele muutmisel jäävad osad elemendid tõlkimata (nt. nupud). Samuti esineb segadusse ajavat kasutajaliidest nt. „Add Legend” nupule vajutades avaneb dialoog küsimusega, kas antud mängu seisundit tuleks muuta.

Veebirakendus oli üles ehitatud kasutades Spring raamistikku. Spring on populaarne ning väga mitmekülgne raamistik veebirakenduste loomiseks. Seega oli tehnoloogia valik igatpidi väga sobilik. Nende loodud andmemudel on nähtav Lisas 1.

2.1.2 Mobiilirakendus

Mobiilirakenduse jaoks seati endale 3 funktsionaalset nõuet [2] :

- 1. Mängija saab identifitseerida ennast kokkulepitud mängus.*
- 2. Mobiiliäpp edastab mängija asukoha ja kuvab mängijale kontrollpunktis saadud legendi.*
- 3. Mängija saab küsida täiendavat informatsiooni mängu ülesannete lahendamiseks ning avaldada soovi mängust lahkumiseks (jäi 3. iteratsiooni skoobist välja).*

Sarnaselt veebirakendusele katavad seatud funktsionaalsed nõuded LARPi põhivajadusi.

Rakenduse käivitamisel soovitakse kasutajalt saada mängukoodi ja kasutaja soovitud kasutajanime selleks mänguks. Samuti on nupp eelmise mängu jätkamiseks. Seega

esimene funktsionaalne nõue on täidetud. Mängu sisenedes kuvatakse kasutajale lähima legendi pealkiri ja kirjeldus teksti kujul. Rohkem funktsionaalsust rakenduses ei olnud. Mobiilirakendus oli väga minimaalne ning funktsionaalsus jäi suuresti täitmata.

Rakendus oli loodud Androidi platvormile, kirjutatud Java programmeerimiskeeles ning kasutusel oli Androidi SDK.

2.2 Olemasolevad lahendused

Aaretejahid ja maastikumängud leiavad aset tihti, sellest räägib juba suur hulk teenuseid, mis selle korraldamist pakuvad. Eriti rohkesti leiavad kasutust äriklientidele suunatud aaretejahid, mis toimuvad tiimides. Sellised lahendused reklaamivad ennast edukate motivatsiooni ning meeskonnatöö võimenditena. Olemasolevate lahenduste funktsionaalsusi võrreldakse tabelis 1.

2.2.1 Go Team - Go hunt

Go Team - Go hunt [4] teenuse eesmärgiks on arendada tiimide sisest kommunikatsiooni. Nad pakuvad varasemalt väljatöötatud asukohapõhiseid aaretejahte ettevõtetele. Mängus osalev grupp jaotatakse tiimideks, igale tiimile antakse tahvelarvuti või soovi korral installitakse vastav rakendus mõne tiimiliikme omanduses olevasse tahvelarvutisse. Rakendus kuvab vihjeid ning suunab tiime läbima erinevaid asukohti. Mängus liigutakse punktist punkti järgides suunaviite.

Õigesse punkti jõudes kasvatab tiim skoori. Olulisel kohal on jooksvalt küsimuste vastamine punktist-punkti liikumise ajal. Mängu lõppedes kuvatakse edetabel, skoori järgi reastatult.

- Platvormid: Android ja iOS
- Hind: Sõltub asukohast ja mängijate arvust
- Kättesaadavus: Välja valitud riigid üle maailma k.a. Eesti
- Link:<http://www.goteam.guru/about/go-hunt>,
<https://play.google.com/store/apps/details?id=fi.taz.goteam.client>

2.2.2 Goosechase

Goosechase [5] pakub võimalust mäng ise koostada. Iga mäng koosneb missioonidest. Soovi korral on võimalik mängijate arvu piiramiseks anda mängule parool. Mängus olevad missioonid võivad varieeruda mingi kindla sisuga pildi tegemisest, mingisse punkti minemisega või mingi tegevuse filmimisega. Teised saavad seejärel hinnata tehtud pilte ja videoid ning mängu võidab tiim, kes kogus kõige rohkem punkte. Mängud kestavad kuni 7 päeva.

- Platvormid: Android ja iOS

- Hind: Tasuta - Piiratud funktsionaalsus ning lubatud ainult pere või sõpradega, max 5 tiimi. Vastasel juhul alates 499\$/mäng.
- Kättesaadavus: Globaalne
- Link:<https://www.goosechase.com/> , <https://play.google.com/store/apps/details?id=com.goosechaseadventures.goosechase>

2.2.3 Klikaklu

Klikaklu [6] mängud koosnevad vihjetest, vihjed omakorda geograafilistest koordinaatidest ja piltidest (nt. omapärased elemendid linnapildis). Esmalt peab mängija asukoha üles leidma, seejärel kuvatakse neile pilt. Nüüd peab mängija nuputama, kus täpsemalt pilt tehti ning seejärel tegema võimalikult sarnase pildi. Sarnase pildi tuvastab kliendi poolel pildituvastusalgoritm. Organiseerija võib mängu üles seada nii, et mängu alustades on kõik vihjed mängijatele nähtavad, samuti on võimalik panna paika järjekord ning kuvada mängijale järgmist vihjet alles eelneva täitmisel. Rakendus on saanud positiivset tagasisidet ning Apple on rakendust esile tõstnud 6 kuud järjest.

- Platvormid: iOS
- Hind: Tasuta kuni kahele mängijale ja kümnele punktile. Edasi tasuline, hind sõltub mängijate ja punktide arvust, võimalik on maksta mängu või päeva kaupa.
- Kättesaadavus: Globaalne
- Link:<http://www.klikaklu.com/>
<https://itunes.apple.com/us/app/klikaklu/id541735944?ls=1&mt=8>

2.2.4 Scavify

Scavify [7] mängud koosnevad ülesannetest. Ülesande sisuks võib olla pildi tegemine, küsimusele vastamine või kindlasse punkti minek. Mängus tehtud pilte julgustab rakendus üles laadima sotsiaalvõrkustikesse. Mängijate tehtud pildid mängu vältel on nähtavad teistele mängijatele hindamiseks ja kommenteerimiseks. Nende klientide

hulgas on olnud nii Google, IBM, Yale'i ülikool, ExxonMobil kui ka mitmed loomaaiad.

- Platvormid: Android ja iOS
- Hind: Tasuline
- Kättesaadavus: Ülemaailmne
- Link: <https://www.scavify.com/> , <https://play.google.com/store/apps/details?id=com.scavify.app>

2.2.5 Actionbound

Actionboundi [8] esmane eesmärk on harida kasutajaid. Kasutajad saavad luua teekondi, mis koosnevad punktidest. Punktide juures on võimalik skaneerida QR kood või vastata mõnele küsimusele, seejärel rakendus kuvab täiendavat informatsiooni antud asukoha kohta.

- Platvormid: Android ja iOS
- Hind: Tasuta piiratud funktsionaalsusega eratarbeks. Tasuline suurema mängu korraldamiseks.
- Kättesaadavus: Ülemaailmne
- Link: <https://en.actionbound.com/>,
<https://play.google.com/store/apps/details?id=de.actionbound>

2.2.6 Strayboots

Strayboots [9] pole avalikult saadaval, vaid üritus-by-üritus kaupa. Ürituse korraldamiseks tuleb nende müügimeeskonnaga ühendust võtta, seejärel koostatakse vastavalt ettevõtte soovidele maastikumäng. Tiimid laevad alla äpi, mille värvid ja logo on vastavalt ettevõtte soovidele varasemalt paika pandud. Äpis olevaid juhiseid järgides läbivad tiimid asukohti, nendes asukohtades ja tee peal vastavad erinevatele küsimustele, mille eest saavad punkte. Reaalajas on võimalik suhelda läbi rakenduse

teiste osalejatega ja jagada pilte. Samuti on reaalajas võimalik sirvida edetabelit ja pärast ürituse lõppu tehtud pilte sotsiaalmeedias jagada.

- Platvormid: Android ja iOS
- Hind: Tasuline, alates 650\$/mäng
- Kättesaadavus: Valitud USA linnades
- Link: <https://www.strayboots.com/>

2.2.7 Loquiz

Loquiz [10] ideeks on tuua lauamäng toast välja, mängunuppudeks on tiimid, kes liiguvad ühest kohast teise, igas punktis lahendades mingi ülesande. Võimalik on näha edetabelit, kui hästi läheb sinul võrreldes teiste tiimidega. Mängus on näha koheselt kõik punktid ning võimalik on luua strateegiat ehk määrata ära, millist teekonda mööda punkte läbitakse. Sarnaselt Go Team - Go hunt lahendusele on mängijatel kasutada tahvelarvuti, mille järgi punktist-punkti liikuda. Loquiz on Eestis välja töötatud lahendus ning muuhulgas on saanud EASist kahel korral toetust, kogusummas 38800€ [11].

- Platvormid: Loquiz annab oma seadme, millel mängida. Seadme operatsioonisüsteemiks on Android.
- Hind: Tasuline
- Kättesaadavus: Ülemaailmne
- Link:<http://loquiz.com/index.html>

Table 1: Sarnaste lahenduste funktsionaalsuste võrdlus

Lahenduse nimetus Funktsionaalsus	Go Team – Go Hunt	Goosechase	Klikakatu	Scavify	Actionbound	Strayboots	Loquiz
Analüütika	Ei	Jah	Ei	Jah	Jah	Jah	Jah
Asukohtade kirjeldustes videod	Ei	Jah	Ei	Ei	Jah	Ei	Ei
Boonuspunktide süsteem	Jah	Jah	Ei	Jah	Ei	Jah	Jah
Kasutajate autentimine	Ei	Jah	Jah	Jah	Ei	Jah	Jah
Kliendile kohandatud rakenduse disain	Ei	Jah	Ei	Jah	Ei	Jah	Jah
Kõik ülesanded asukohapõhised	Ei	Ei	Jah	Ei	Jah	Ei	Ei
Mängu loomine mobiilirakenduses	Ei	Ei	Jah	Ei	Jah	Ei	Ei
Mänguala defineerimise võimalus	Ei	Ei	Ei	Ei	Ei	Ei	Ei
Mängujuht näeb mängijate asukohti	Ei	Ei	Jah	Ei	Ei	Jah	Jah
Punktide läbimine suvalises järjekorras	Ei	Jah	Jah	Jah	Ei	Ei	Jah
Punkti tüüpide arv	1	1	2	1	2	2	1
Reaalajas edetabel	Ei	Jah	Ei	Jah	Ei	Jah	Jah
Saadaval mitmes keeles	Jah	Ei	Ei	Ei	Jah	Ei	Jah
Sarnase pildi tuvastamine	Ei	Ei	Jah	Ei	Ei	Ei	Ei
Sisenemine mängukoodiga	Ei	Ei	Ei	Ei	Ei	Ei	Ei
Sotsiaalne element	Ei	Jah	Ei	Jah	Ei	Jah	Ei
Ülesande tüüpide arv	3	4	2	4	2	2	2
Vestlus mängujuhiga läbi rakenduse	Ei	Ei	Ei	Ei	Ei	Jah	Ei
Vestlus teiste tiimidega mängu ajal	Ei	Ei	Ei	Ei	Ei	Jah	Ei

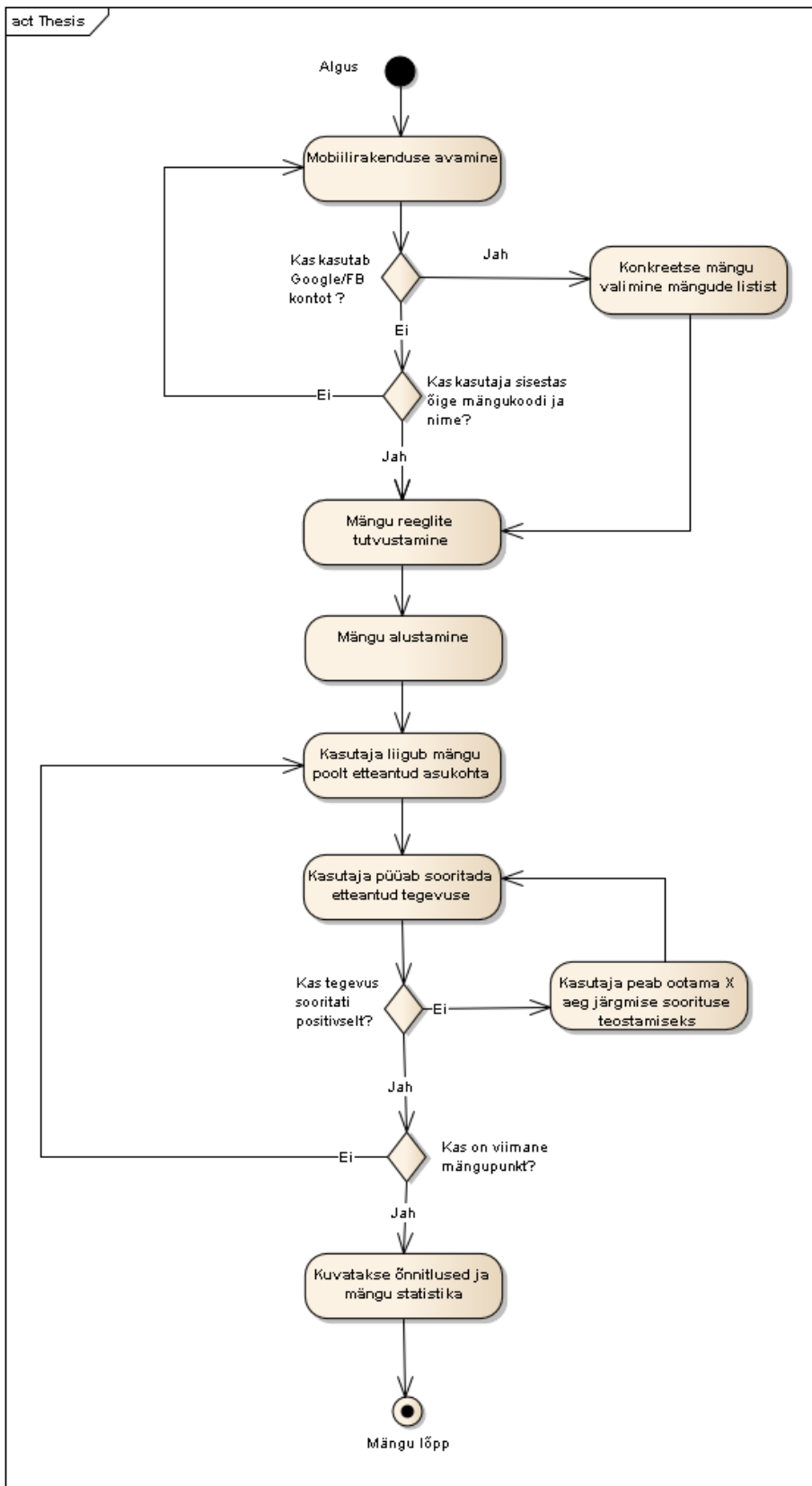
3 Funktsionaalsed nõuded

Eesmärgiga pakkuda võimalikult paindlikku lahendust on mobiilirakendusele ette seatud järgnevad funktsionaalsed nõuded:

1. Autentimine kasutades Google või Facebooki kontot, kuid see pole kohustuslik. Võimalik on otse mängu siseneda ka kasutades mängukoodi ning valides endale kasutajanime. Esimesel juhul peab olema võimalik sirvida mängude loetelu, mis sisaldab käimas olevaid, tulemas ja möödunud mänge. Teisel juhul see võimalus puudub.
2. Mängija saab mängu ajal näha järgmise punkti leidmise jaoks asjakohast informatsiooni. Kuvatav informatsioon sõltub punkti tüübist. Ühel punktil peab olema vähemalt üks tüüp. Punktide võimalikud tüübid on järgmised:
 - a) Kaugus: Mängijat hoitakse kursis punkti ja tema vahelisest distantsist. Distants on antud meetrites.
 - b) Vihje: Kuvatakse mõistatus või vihje, mis on seotud otsitava punktiga.
 - c) Kompass: Mängija saab teada punkti suuna enda suhtes.
 - d) Kaart: Punkt kuvatakse kaardil. Kuigi kaart ja mängija asukoht sellel on alati nähtav, siis otsitava punkti asukoht on nähtav ainult siis, kui kaardi punktitüüp on otsitava punktiga seotud.
3. Iga punktiga on seotud üks ülesanne. Iga punktiga on seotud üks kohustuslik ülesanne. Vale vastuse korral tuleb oodata võimalust uuesti vastata. Ooteaeg on defineeritud mängulooja poolt, kuid võib ka puududa. Ülesannete tüübid on järgmised:

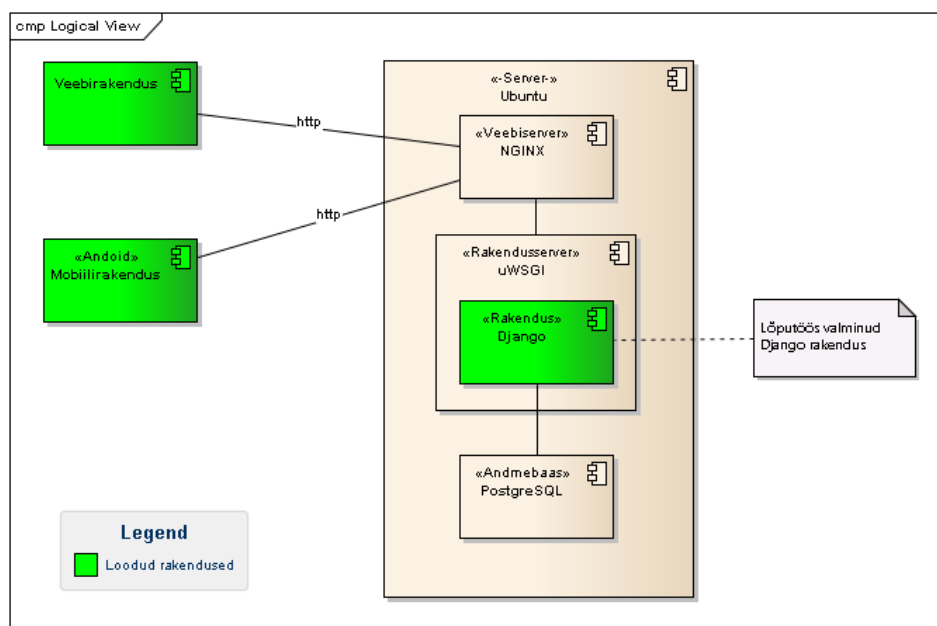
- a) NFC: Mängija peab leidma peidetud NFC kiibi.
 - b) Asukoht: Piisab mängija jõudmisest punkti
 - c) Küsimus: Kuvatakse valikvastustega küsimus.
 - d) Kaamera: Vastavalt etteantud kirjeldusele on vaja teha pilt.
 - e) QR: Mängija peab skaneerima punktis olevat QR-koodi.
4. Mängijate asukoht salvestatakse teatud ajavahemike järel serverisse. Kui mängija uus asukoht on mängujuhi poolt defineeritud alal, siis saadetakse vastava alaga seotud sõnum mängijale.
 5. Mängijal on võimalus rakenduse siseselt vestelda mängujuhiga.
 6. Mängu lõppedes on võimalik näha statistikat mängu kohta.
 7. Mängu loomine läbi rakenduse (edasine arendus).
 8. Sotsiaalne element - võimalik jagada tehtud pilte, statistikat, reaajas edetabel mängijatele nähtav (edasine arendus).
 9. Saadavus mitmes keeles (edasine arendus).

Kasutajatepoolne põhitegevustejada mängu lõpetamiseks, mis on antud projekti lõpetamiseks realiseeritud on kuvatud joonisel 1. Funktsionaalsuses võrdlus sarnaste lahenduste vahel ning väljapakutud lahenduse vahel on nähtav tabelis 1.



Joonis 1: Tegevustediagramm mängu positiivseks läbimiseks

4 Arhitektuur



Joonis 2: Komponentide loogiline vaade

Näidisserver koosneb neljast komponendist, mis jooksevad Ubuntu 16.10 operatsioonisüsteemil (Joonis 2). Veebiserverina on kasutusel NGINX¹. Rakendusserverina kasutatakse uWSGI-d² ja veebirakendus ise on kirjutatud Pythoni programmeerimiskeeles kasutades Django³ raamistikku. Kuna veebiserveril pole võimalust otse Pythoni rakendusi jooksutada oli tarvis kasutada liidest, mis teeks veebiserverile need rakendused mõistetavaks, selleks kasutati uWSGI-d. Django raamistik valiti, kuna sellel on taga tugev kogukond, palju pistikprogramme, hea dokumentatsioon ning raamistiku disaini filosoofiad pakkusid autorile suurt huvi. Teised valikud põhinevad heal liidestatavusel Djangoga ja on avatud lähtekoodiga tasuta saadaval. [19]

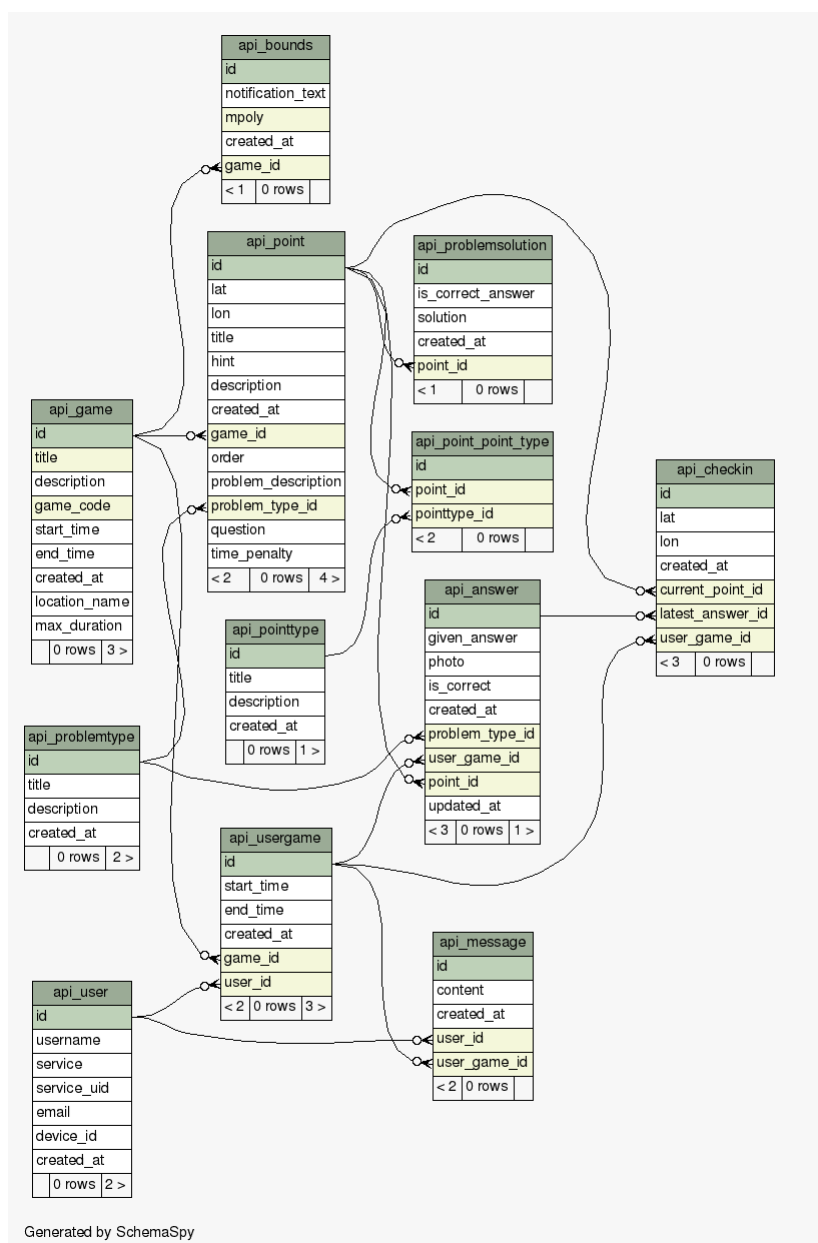
¹ <https://www.nginx.com/>

² <https://uwsgi-docs.readthedocs.io/en/latest/#>

³ <https://www.djangoproject.com/>

4.1 Andmebaas

Andmebaasina on kasutusel PostgreSQL. Andmemudel koosneb 12 tabelist. Andmebaasi disain on genereeritud Django ORM poolt kontseptuaalsete mudelite abil. Andmemudel pole lõplik, puudu on näiteks kasutajate õigussüsteem, mis lisandub tulevikus. Detailsemalt on võimalik andmemudelit uurida aadressil: <http://dijkstra.cs.ttu.ee/~Hans.Lahe/andmemudel/index.html>



Joonis 3: Andmemudel

4.2 Kliendi suhtlus serveriga

Androidi rakenduse ja serveri omavaheliseks suhtluseks tuli luua API (Lisa 2 – API kirjeldus). Rakendus suhtleb serveriga HTTP päringute abil ning lähtuti REST metoodika printsiipidest. Serveri vastuse kuju on JSON formaadis, kuna teise populaarseima alternatiivi XMLi lisavõimaluste järgi puudub vajadus. RESTi loomise hõlpsustamiseks on kasutatud Django REST raamistikku.

4.3 Veebirakendus

Mängude loomine ja haldamine käib hetkel läbi Django Administratsiooni paneeli (Lisa 3 – Administratsiooni paneeli üldvaade ja Lisa 4 – Administratsiooni paneeli piirava ala lisamise vaade). Veebirakenduse lähtekood on saadaval aadressil <https://gitlab.pld.ttu.ee/Hans.Lahe/thesis-backend.git> .

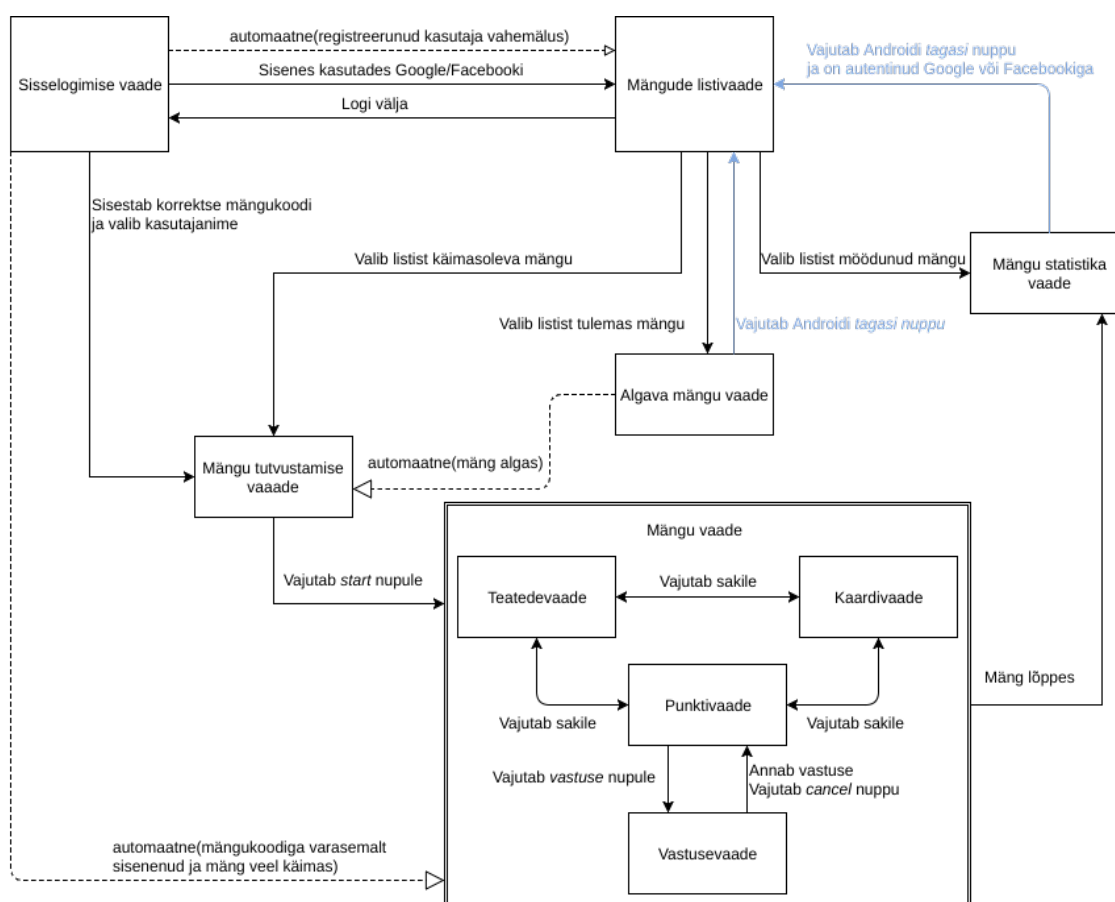
Projekti järgmise etapina on ette nähtud lisavõimalustega veebirakenduse loomine. Lisavõimaluste hulka kuulub muuhulgas mängijate asukoha kuvamine kaardil ning analüütika mängu ajal ja lõppedes.

5 Mobiilirakenduse kirjeldus

Käesolevas töös loodud mobiilirakenduse sihtversioon on Android 5.1. Rakendus on testitud 5.1 ja 7.1 versioonil, minimaalseim Androidi versioon, kus rakendus töötab on 4.4. 2017. aasta mai kuu seisuga katab see ligikaudu 90% Androidi seadmetest [12]. Lisa piiranguks on kaamera ja NFC võimekuse vajadus. Lähtekood on avalikult saadaval aadressil: <https://gitlab.pld.ttu.ee/Hans.Lahe/thesis.git>.

5.1 Kasutajaliides

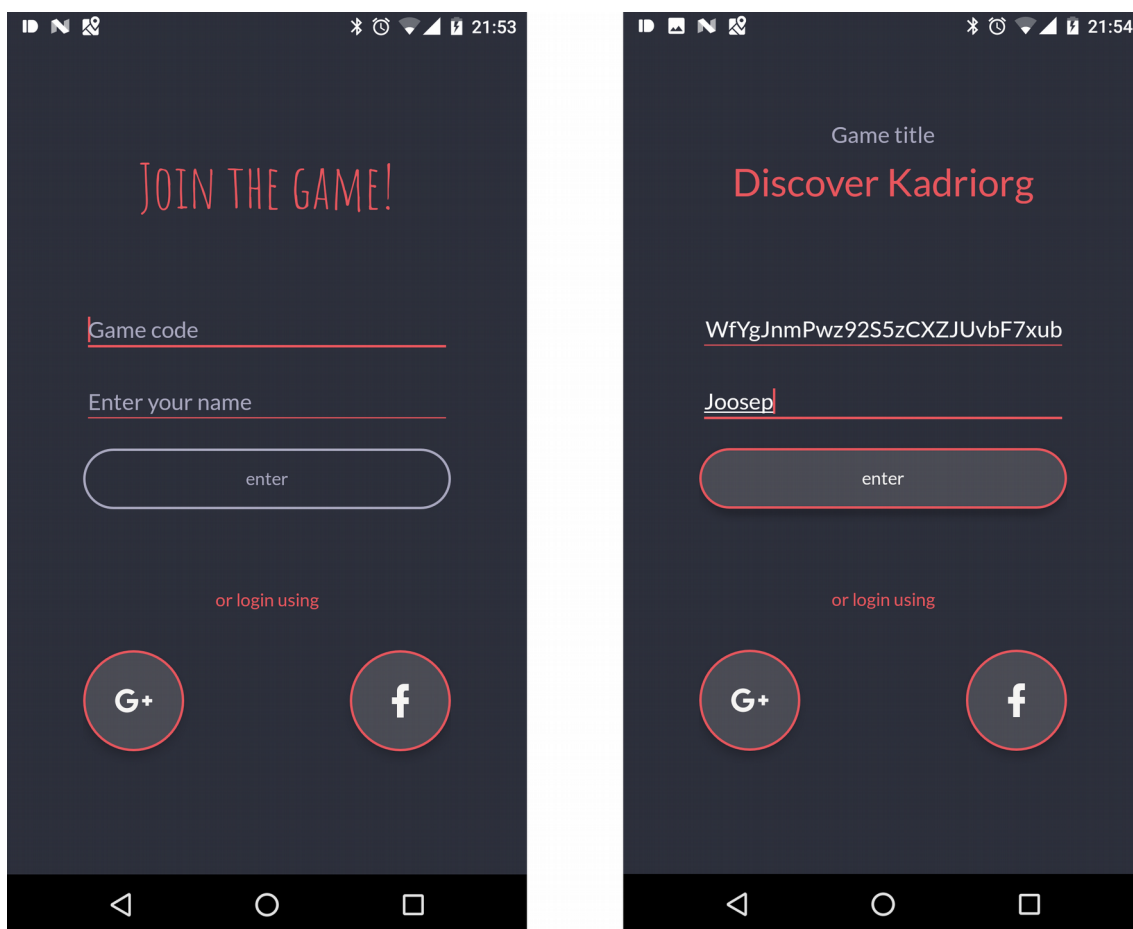
Kasutaja navigeerimisvõimalusi iseloomustav navigatsioonidiagramm on nähtav joonisel 4.



Joonis 4: Navigatsioonidiagramm

5.1.1 Sisselogimise vaade

Rakenduse avamisel kuvatakse kasutajale vaade (Joonis 5), kus on võimalik ennast autentida kasutades Google'i kontot või Facebooki kontot. Lisaks sisselogimisele on antud vaates võimalik sisestada ka mängukood ning kasutajanimi. Kuna rakendus on suunatud ettevõtetele ning ürituste korraldajatele, siis individuaalne osaleja ei pruugi soovida endale kasutajat teha, vaid eelistaks võimalikult kiiresti mängima asuda. Sisestades mängukoodi ja valides endale kasutajanime, avaneb kasutajal võimalus vajutada nupule *Enter*, mille peale avatakse Mängu tutvustamise vaade. Kui kasutaja oli varasemalt sisseloginud ning rakenduse vahemälu polnud kustutatud, siis avatakse automaatselt Mängude listivaade. Kui kasutaja oli varasemalt alustanud mängu kasutades mängukoodi ning kasutajanime ja mäng pole läbi, siis avaneb automaatselt Mängu vaade.



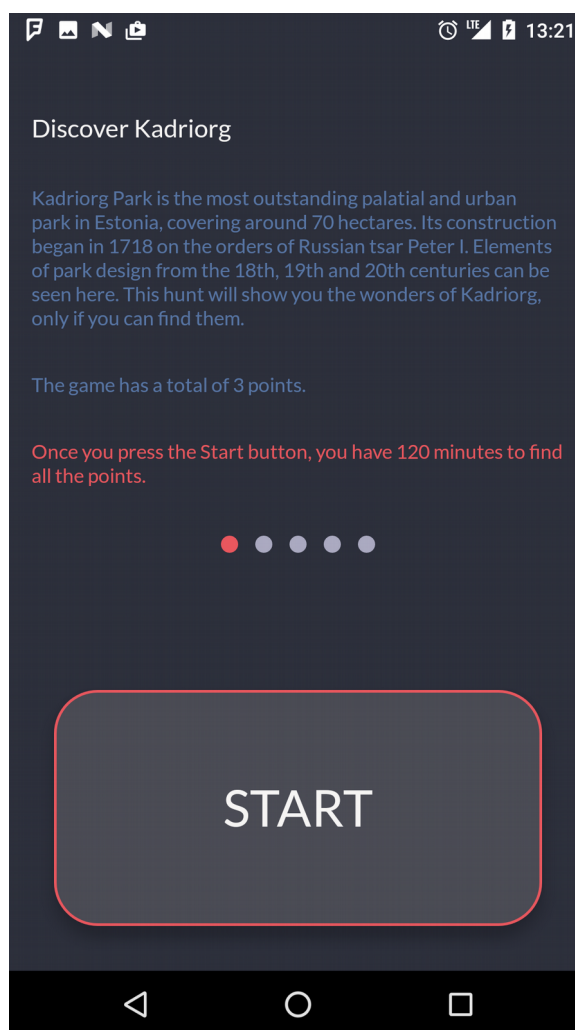
Joonis 5: Sisselogimise vaade

5.1.2 Mängude listivaade

Mängude listivaates on loend kolme alapeatükiga - toimunud, tulemas ja käimasolevad mängud. Antud vaates on ka nupp, millele vajutades avaneb dialoog, kuhu on võimalik sisestada mängukood, mis lisab eelpool mainitud listi vastava mängu juurde. Lisaks on nupp välja logimiseks.

5.1.3 Mängu tutvustamise vaade

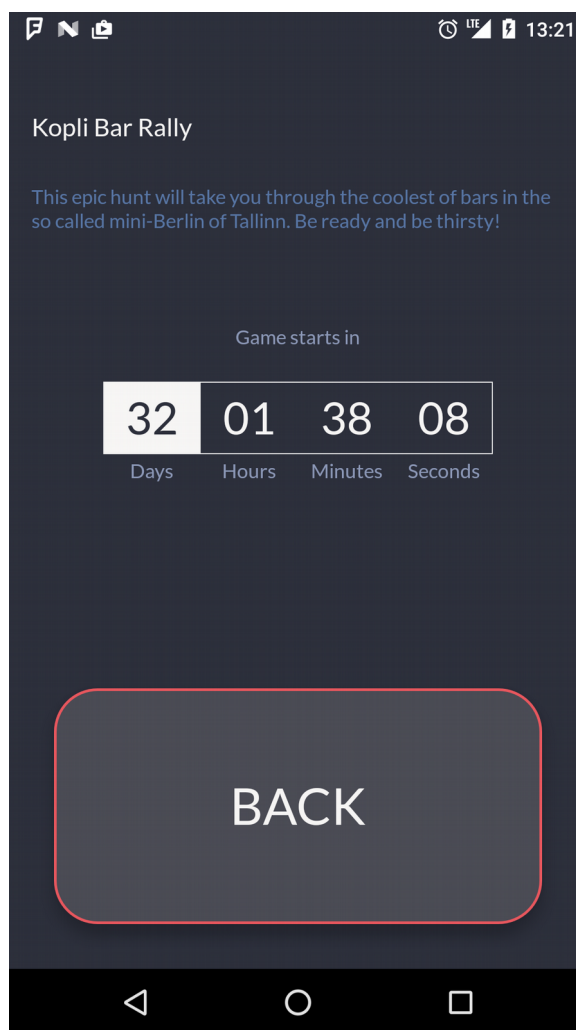
Vaates (Joonis 6) kuvatakse mängu nimi, kirjeldus, punktide arv. Kui mäng on ajaliselt piiratud, siis ka aeg, mitu minutit on mängimiseks. Selles vaates on võimalik ka sirvida õpetust. Vajutades *Start* nuppule avaneb Mängu vaade.



Joonis 6: Mängu tutvustamise vaade

5.1.4 Algava mängu vaade

Vaade (Joonis 7) kuvab mängu põhiinfo nagu pealkiri ja kirjeldus. Samuti näidatakse kasutajale taimerit, mis läheneb mängu alguse ajale. Nulli jõudes viiakse kasutaja automaatselt Mängu tutvustamise vaatesse.



Joonis 7: Algava mängu vaade

5.1.5 Mängu statistika vaade

Mängu statistika vaates saab mängija näha statistikat oma mängu kohta - teekonda kaardil, mitmenda koha saavutas, teekonna pikkust meetrites, keskmist kiirust jms.

5.1.6 Mängu vaade

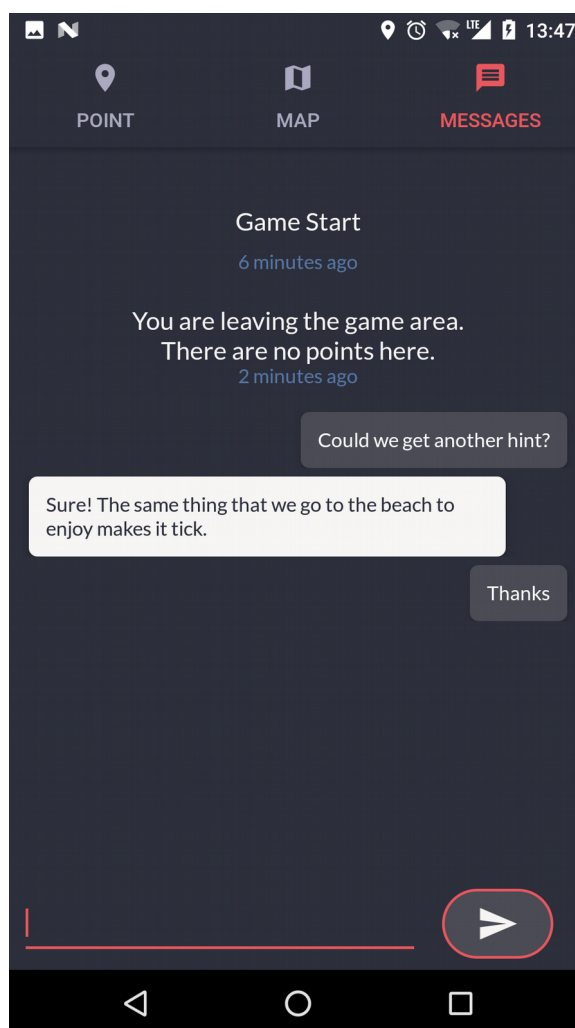
Mängu vaade koosneb neljast osast – punktivaade, kaardivaade, teadetevaade ja vastusevaade. Neist kolme esimese vahel on võimalik navigeerida sakkide (*tabide*) abiga. Järgnevalt on nad välja toodud detailsemalt

- Punktivaate (Joonis 8) eesmärk on juhendada mängija punkti läbima. Antud vaates on kasutajale näha punkti detailid: pealkiri, kirjeldus, mitmenda punktiga on tegu. Samuti kuvatakse punkti leidmiseks vajaminev info. Selleks võib olla vihje koos kaugusega punktist nagu väljatoodud näite puhul. Lisaks sellele võib kasutajat juhendada kompass. Vaate alumises osas on juhend, mida punktis teha tuleb. See võib olla lihtsalt punkti minek, QR-kood, NFC, pildistamine või küsimusele vastamine.



Joonis 8: Punktivaade

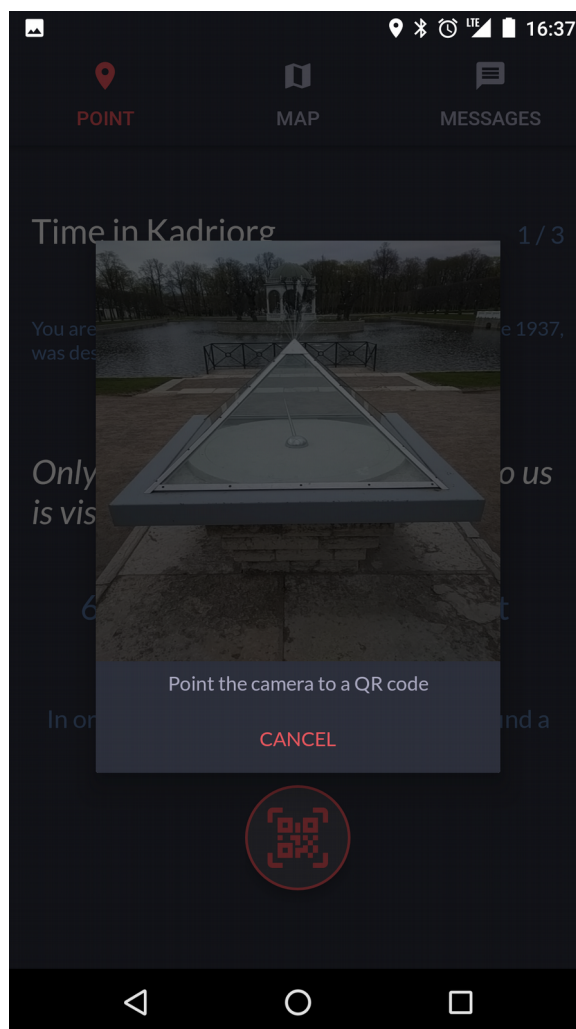
- Kaardivaade annab kasutajale mugava ülevaate tema hetke asukohast. Samuti võib siin olla märgitud ka otsitava punkti asukoht. See sõltub, kuidas mängujuht on punkti seadistanud.
- Teadetevaates (Joonis 9) on võimalik mängijal vestelda mängujuhiga. Siia vaatesse tuleb ka teade, kui mängija satub mängujuhi poolt defineeritud alasse.



Joonis 9: Teadetevaade

- Vastusevaate tüüpe on mitmeid. Tüüp sõltub, millist lahendust punkt nõuab. Selleks on küsimusele vastamine, NFC, pildi tegemine või QR-koodi skaneerimine (Joonis 10). Õige vastuse korral antud vaade asendub hüpikaknaga (pop-up), mis teavitab, et punkt sai läbitud ning kuvatakse järgmine

punkt. Vale vastuse korral vastusevaade sulgub ning mängija peab ootama mängujuhi poolt määratud arv sekundeid enne, kui saab uue vastuse esitada.



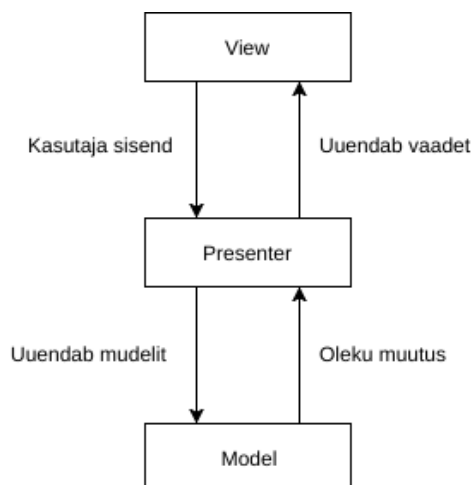
Joonis 10: Vastusevaade

5.2 Rakenduse arhitektuur

Rakenduse arhitektuuri paika panemisel võeti aluseks Google näidisprojekt *TODO-MVP-RXJAVA*¹. Autor võttis kasutusele RxJava teegi sooviga kasutada reaktiivse programmeerimise võtteid, et lihtsustada koodi haldamist. Peamiselt kasutatakse RxJavat *model* ja *presenter* kihtide vaheliseks suhtluseks.

¹ <https://github.com/googlesamples/android-architecture/tree/todo-mvp-rxjava/>

Androidi rakenduse ülesehitamisel kasutati MVP² (Joonis 11) arhitektuuri . Kuna Androidi raamistik ei suuna arendajaid kindlat arhitektuuri valima, siis suuremahulise projekti puhul nagu selleks on antud töö, võib kood minna väga raskesti hallatavaks, kui kindlat arhitektuuri ei kasutata. MVP kasuks räägivad modulaarsus ja komponentide selge eristatavus. MVP sarnaneb MVCle, suurim erinevus on, et MVP puhul *View* kiht ei suhtle *Model*ga otse. [13] [14]



Joonis 11: Model View Presenter diagramm

MVP jagab rakenduse kolmeks põhikomponendiks [13] :

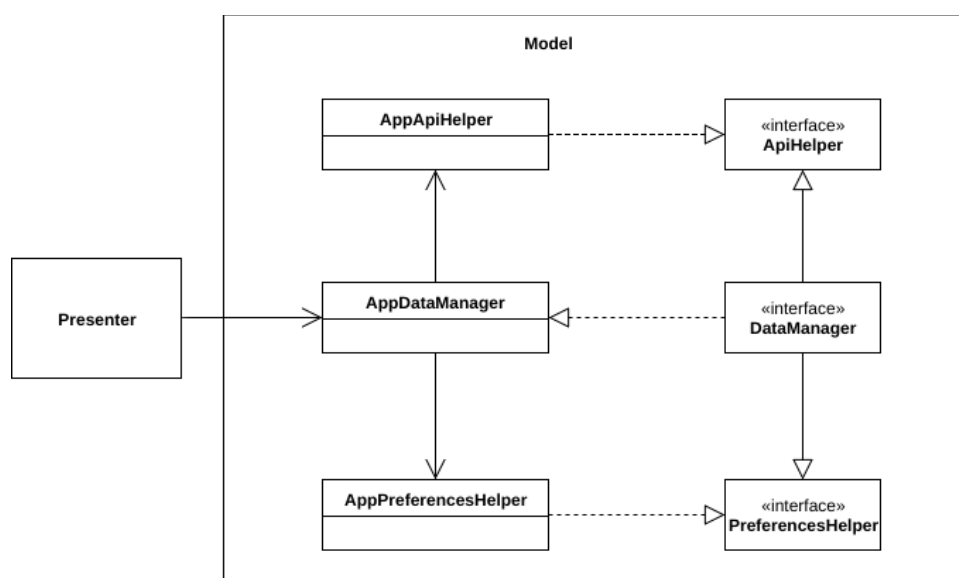
1. *Model*: tegeleb andmete haldamisega.
2. *Presenter*: on sild *modeli* ja *view* vahel. *Presenter*isse läheb kogu äri loogika. *Presenter* vastutab *Model*ist andmete pärimise ja *View* uuendamise eest ning vastupidi, kasutaja sisendi korral *View*'ga uuendab *Presenter* äri loogika alusel *Model*'it.
3. *View*: vastutab kindla infoga vaadete kuvamise eest ekraanile. *View* on implementeeritud *Activity*, *Fragmendi* või mõne muu Androidi raamistiku klassi poolt, mille vastutusalaks on kasutajaliides.

MVP seab ka paar reeglit eelpool mainitud komponentidele:

1. *View* ainuke kohustus on kasutajaliidese kuvamine ekraanile nii nagu, *Presenter* seda otsustab.

² <https://en.wikipedia.org/wiki/Model-view-presenter>

2. *View* edastab kasutaja sisendi oma *Presenter*ile.
3. *View* ei suhtle kunagi *Model*iga otse.
4. *Presenter* on kohustatud edastama *View*'st tulnud kasutaja sisendi *Model*ile ja *View*le omakorda vastavalt uusi juhendeid saatma.
5. *Model* vastutab andmete pärimise eest serverist, failisüsteemist ja andmebaasist.



Joonis 12: Mudeli osa detailne vaade

Modeli osa jaotub kolmeks (Joonis 12) - veebiserveriga suhtlev osa, vahemälu haldav osa ning kõike seda vahendav osa. *ApiHelper* ja *PreferencesHelper* on abiklassid, mis sisaldavad vastavalt API kutsumise ja vahemälu haldamise meetodeid. *ApiHelper*it realiseerib *AppApiHelper*, mis tegeleb veebiserveri suhtlusega. Antud töös on veebiliidesega suhtlemisel kasutatud *Rx2AndroidNetworking* teeki. *AppDataManager* klassi läbi käib rakenduses kõik andmetega seotud funktsionaalsus. Eelpool mainitud abiklassid *Apihelper* ja *PreferenceHelper* töötavad ainult *DataManager*eri heaks. See edastab kõik tegevused vastavale abiklassile. Peamine põhjus, miks *presenter* näiteks otse *AppApihelper*iga ei suhtle, on see, et olenevalt kasutusjuhust on vaja pöörduda näiteks vahemällu ja seejärel serveri poole.

5.3 Rakenduse struktuur

Antud projekti Androidi osa on üsna mahukas, sisaldades kokku üle 100 Java klassi, liidese ja annotatsiooni. Seega projekti tulevikku silmas pidades on oluline paika panna arusaadav ja projekti läbiv struktuur. Levinud on kaks fundamentaalselt erinevat lähenemist.[15]

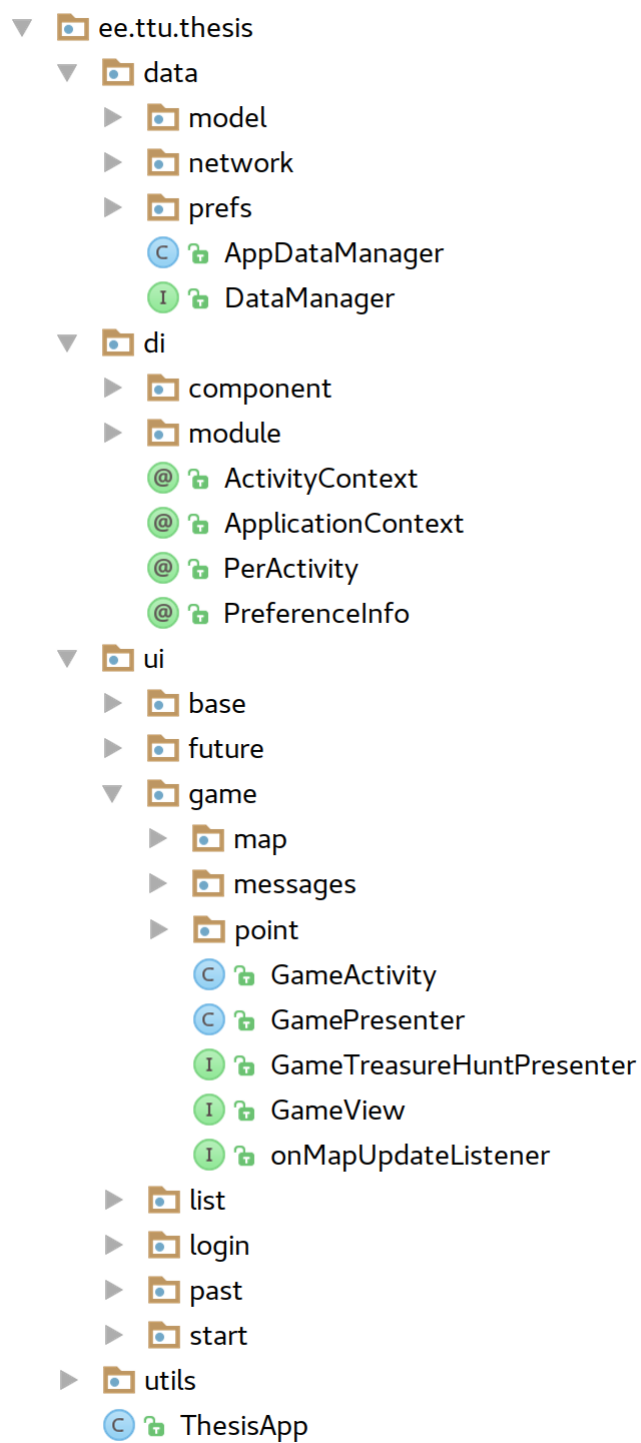
Esimene on jagada rakendus loogilisteks kihtideks ehk *Package By Layer(PBL)*. Pannes näiteks *ee.ttu.thesis.presenter* pakki kõik *presenterid* ning *ee.ttu.thesis.activities* pakki kõik *activity*'d. Selle lähenemise puhul on mugavam vahetada teeki, näiteks kui *activity*'d kasutavad GSON teeki, mis võimaldab JSONi vastavaks Java objektiks muuta, kuid nüüd arendaja sooviks GSONi asemel Jacksoni teegi kasutusele võtta, siis PBL puhul oleks selline vahetus märkimisväärselt mugavam, sest kõik *activity*'d on ühes kohas. [15]

Teine lahendus oleks vaadata konteksti ning panna ühte pakki need, mis keskenduvad samale funktsionaalsusele. See lähenemine on tuntud kui *Package By Feature(PBF)*. See tähendaks et *ee.ttu.thesis.ui.login* sisaldab sisselogimisega seotud klasse, nii *loginPresenterit*, *loginActivityt*, *loginFragmente* jms. Suuremate tükide puhul - nagu selleks on antud juhul mänguvaade - jaotuks *ee.ttu.thesis.ui.game* veel omakorda väiksemateks osadeks PBF põhimõtte järgi, näiteks *ee.ttu.thesis.ui.game.point*, mis sisaldab punktivaatega seotut. Selle lähenemise eeliseks on suurem modulaarsus, mugavam navigeerimine, kõrgem abstraktsuse tase ning projekti mahukuse kasvamisega on mugavam teda hallata. Nendel põhjustel otsustas autor ka selle lähenemise kasuks. [16] [17]

Käesoleva projekti Androidi struktuur (Joonis 13) on jaotunud neljaks suuremaks osaks.

- *data*, sisaldab klasse, mis on seotud andmete haldamisega rakenduses. Samuti on siin *modeli* klassid, kõik andmeobjektid, mis rakenduses kasutust leiavad.
- *di* hoiab kõikki sõltuvust süstivaid klasse. See aitab hoida kõik injektsiooni sätted ja vastutused ühes kohas, eraldatuna ülejäänust rakendusest.

- *ui* on kõige mahukam osa ning hoiab klasse, mis on seotud kasutajaliidese komponentidega rakenduses. Vastavalt PBF põhimõtetele on *ui* omakorda jaotunud vastavalt funktsionaalsule mitmeteks tasemeteks. See teeb arendustöö meeldivamaks ja haldamisel säästab aega otsingu pealt. Eraldi on *ui* pakis ka base alampakk, mis hoiab *BaseActivity*, *BasePresenter* jms klasse, mida teised *ui* klassid laiendavad või liidestavad.
- *utils* hoiab igasuguseid abiklasse, mis on seotud näiteks laadimisriba loomisega kuni ekraani mõõtmete hankimiseni välja.



Joonis 13: Rakenduse struktuur

6 Kokkuvõte

Eesmärgiks oli realiseerida maastikumänge toetav mobiilirakendus. Selle saavutamiseks analüüsiti esmalt olemasolevaid lahendusi ja nende funktsionaalsust. Seejärel sõnastati funktsionaalsed nõuded, mis said ettenähtud mahus täidetud.

Lõputöö raames seati püsti ka mobiilirakendust toetav server. Serverisse loodi Django raamistiku abiga esialgne veebirakendus, kus toimus mängude loomine ja haldamine. Projekti järgmise etapina on plaanis täiendava funktsionaalsusega veebirakenduse arendus.

Selle tööga sai valmis esmane Androidi mobiilirakenduse prototüüp, mis kombineerib orienteerumise ja mõistatuste lahendamise elemendid. Sai antud ülevaade Android rakenduse realiseerimisel kasutatud MVP arhitektuurist. Kirjeldati ka lähtekoodi ülesehitus ning põhjendati tehtud valikuid. Kuna mäng ei ole veel lõplik, sest veebirakendus on täiustamisel ja mobiilirakendus katsefaasis, siis pole rakendus veel avalikult kättesaadav. Küll aga on rakenduse lähtekood avalikult kättesaadav aadressilt: <https://Hans.Lahe@gitlab.pld.ttu.ee/Hans.Lahe/thesis.git>

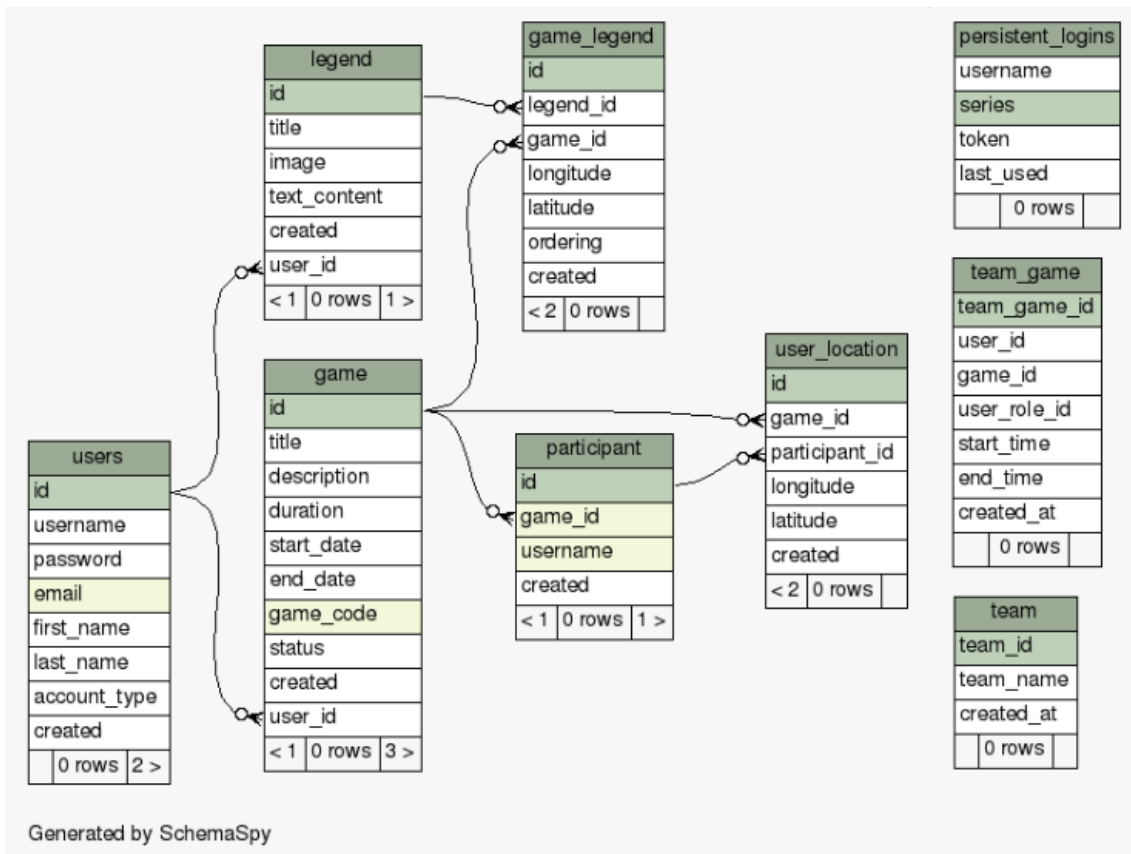
Lõputöö autorile andis tehtud töö arvestataval määral uusi tehnilisi teadmisi Androidi rakenduse ülesehitamisest kui ka Django raamistiku kasutamisest. Autor leiab, et valitud tehnoloogiad toimivad hästi ning kasutab neid ka tulevikus.

Kasutatud kirjandus

- [1] Geocaching. [WWW] <https://en.wikipedia.org/wiki/Geocaching> (20.05.2017)
- [2] Narrative Trip Platform: The Wiki. [WWW] <https://docs.google.com/document/d/1BVnv02y3fK2W8jZba2jgVlhDZUXecmxTcLQVZSN3nm0/edit?usp=sharing> (20.05.2017)
- [3] Simon Kemp. Digital in 2017: Global Overview. [WWW] <https://wearesocial.com/blog/2017/01/digital-in-2017-global-overview> (20.05.2017)
- [4] Go Team – Go hunt. [WWW] <http://www.goteam.guru/about/go-hunt> (20.05.2017)
- [5] Goosechase. [WWW] <https://www.goosechase.com/about/> (20.05.2017)
- [6] Klikaklu. [WWW] <http://www.klikaklu.com/faq.html> (20.05.2017)
- [7] Scavify. [WWW] <https://www.scavify.com/howitworks> (20.05.2017)
- [8] Actionbound. [WWW] <https://en.actionbound.com/> (20.05.2017)
- [9] Strayboots. [WWW] <https://www.strayboots.com/how-it-works> (20.05.2017)
- [10] Loquiz. [WWW] <http://loquiz.com/> (20.05.2017)
- [11] Loquizile eraldatud toetus. [WWW] http://www.eas.ee/toetatud-projektid/?s%5Bsort%5D=date_asc&s%5Bcompany_name%5D=Loquiz&s%5Byear%5D=-1&s%5Bcounty%5D=&s%5Bschema%5D=&s%5Bsearch%5D=Otsi&s%5Breset_value%5D=Taasta (20.05.2017)
- [12] Platform Versions. [WWW] <https://developer.android.com/about/dashboards/index.html> (20.05.2017)
- [13] Francesco Cervone. Model-View-Presenter: Android guidelines. [WWW] <https://medium.com/@cervonefrancesco/model-view-presenter-android-guidelines-94970b430ddf> (20.05.2017)
- [14] Architecture of Android Apps. [WWW] https://github.com/codepath/android_guides/wiki/Architecture-of-Android-Apps (20.05.2017)
- [15] Package by feature, not layer. [WWW] <http://www.javapractices.com/topic/TopicAction.do?Id=205> (20.05.2017)
- [16] Kailash Dabhi. [WWW] MVP With Better Naming of implementation classes & DRY principle. <https://medium.com/@kailash09dabhi/mvp-with-better-naming-of-implementation-classes-dry-principle-e8b6130bbd02> (20.05.2017)

- [17] Organizing your Source Files. [WWW] https://github.com/codepath/android_guides/wiki/Organizing-your-Source-Files (20.05.2017)
- [18] TODO-MVP-RXJAVA. [WWW] <https://github.com/googlesamples/android-architecture/tree/todo-mvp-rxjava/> (20.05.2017)
- [19] Design philosophies [WWW] <https://docs.djangoproject.com/en/1.11/misc/design-philosophies/> (21.05.2017)

Lisa 1 – Narrative Trip Platform andmemudel



Lisa 2 – API kirjeldus

Hangi kõik kasutaja mängud

Päringu näide: `curl -i -H "Accept: application/json" "<uri>/api/user/games?user=18"`

Päringu parameetrid:

- user - kasutaja unikaalne identifikaator

Tagastab listi kasutaja mängudest. Näide: [{"user_game_id": "1","game": {"id": 1,"title": "Kopli Bar Rally","description": "This epic hunt will take you through the coolest of bars in the so called mini-Berlin of Tallinn. Be ready and be thirsty!","game_code": "cIz5LaV2IKEG24sXK0Aw","location_name": "Kopli, Tallinn","max_duration": 60,"start_time": "2017-06-21T12:00:00Z", "end_time": null},"start_time": null,"end_time": null},...]

Mängu põhiinfo hankimine

Funktsiooni on mõeldud mängude leidmiseks. Näiteks mängukoodi sisetamisel kasutajale selle mängu põhiinfot kuvada.

Päringu näide: `curl -i -H "Accept: application/json" "<uri>/api/games?game_code=cIz5LaV2IKEG24sXK0Aw"`

Päringu parameetrid:

- game_code - kasutaja unikaalne identifikaator

Vastuse näide: [{"id": 3,"title": "Discover Kadriorg","description": "Kadriorg Park is the most outstanding palatial and urban park in Estonia, covering around 70 hectares. Its construction began in 1718 on the orders of Russian tsar Peter I. Elements of park design from the 18th, 19th and 20th centuries can be seen here. This hunt will show you

```
the wonders of Kadriorg, only if you can find them.,"game_code":  
"cIz5LaV2IKEG24sXK0Aw","location_name":  
"Kadriorg,Tallinn","max_duration":120,"start_time": "2017-05-  
17T06:00:00Z","end_time": "2017-06-22T12:00:00Z"}]
```

Registreerimata kasutaja lisamine

Funktsioon on mõeldud kasutajale, kes ei soovi registreeruda, vaid mängukoodi ning enda poolt valitud kasutajanimega mängima hakkata.

Päringu näide: `curl --data "username=Joosep&device_id=5rg0f762536274c1" <uri>/api/user/add`

Päringu parameetrid:

- username - kasutaja valitud nimi, kohustuslik
- device_id - kasutaja seadme unikaalne identifikaator, kohustuslik

Vastuse näide: {"user_id": 20}

Kasutajale mängu lisamine

Funktsioon on mõeldud uue mängu lisamiseks kasutajale. Lisatav kasutaja pannakse mängija rolli.

Päringu näide: `curl --dataFloatingActionButton "game_code=cIz5LaV2IKEG24sXK0Aw&user_id=222" <uri>/api/game/adduser`

Päringu parameetrid:

- game_code - mängu identifitseeriv unikaalne kood, kohustuslik
- user_id - kasutaja unikaalne identifikaator, kohustuslik

Vastuse näide: {"user_game_id": 1, "game": {"id": 1, "title": "Kopli Bar Rally", "description": "This epic hunt will take you through the coolest of bars in the so called

mini-Berlin of Tallinn. Be ready and be thirsty!", "game_code": "cIz5LaV2IKEG24sXK0Aw", "location_name": "Kopli, Tallinn", "max_duration": 60, "start_time": "2017-06-21T12:00:00Z", "end_time": null}}

Mängu alustamine

Funktsioon kutsutakse, kui mängija otsustab mängu alustada.

Päringu näide: `curl --data "game_code=cIz5LaV2IKEG24sXK0Aw&user_id=222" <uri>/api/game/start`

Päringu parameetrid:

- `game_code` - mängu identifitseeriv unikaalne kood, kohustuslik
- `user_id` - kasutaja unikaalne identifikaator, kohustuslik

Vastuse näide: HTTP seisundi kood, 201 kui oli edukas ja 4xx kui midagi ebaõnnestus.

Mängu punktide saamine

Funktsiooniga saadakse mängu kõik punktid ning nende punktide lahendused.

Päringu näide: `curl -i -H "Accept: application/json" "<uri>/api/game/points?game=3"`

Päringu parameetrid:

- `game` - mängu identifitseeriv unikaalne identifikaator

Vastuse näide: [{"id":1,"game":3,"lat":"59.437461","point_type":{"title":"DST"}, {"title":"HNT"}], "time_penalty":3000,"question":null,"problem_description":"In order to complete this point, you must find a QR code near the location.", "problem_type":{"title":"QRC"}, {"lon":"24.785737", "title":"Time in Kadriorg", "hint":"Only useful when the closest star to us is visible in the sky.", "description":"You are looking for something that was been there since 1937, was designed by A. Lomann and built by E. Uukivi.", "order":1, "solutions":

[{"id":1,"solution":"sundial","is_correct_answer":true}], {"id":2,"game":3,...]

Mängija asukohta uuendamine

Funktsiooni kutsutakse, kui mängija asukoht uueneb. Teatab, kas mängija on piirides, kui ei ole, siis tagastab vastava teate. Koordinaadid peavad olema kümnendkujul.

```
Päringu näide: curl --data "user_game_id=114&lat=54.485391&lon=24.753438&current_point_id=780" <uri>/api/game/checkin
```

Päringu parameetrid:

- user_game_id - mängu kasutaja kombot identifitseeriv unikaalne identifikaator, kohustuslik
- lat - mängija asukohta laiuskraad, kohustuslik
- lon - mängija asukohta pikkuskraad, kohustuslik
- current_point_id - hetkel otsitava punkti id, kohustuslik.
- latest_answer_id - viimase antud vastuse id.

Vastuse näide:{"bounds": [{"id": 21, "notification_text": "You are leaving the game area. There are no points here."}], "messages": [{"id": 79, "user": 5, "content": "Sure! The same thing that we go to the beach to enjoy makes it tick.", "created_at": "2017-05-20T10:46:33Z"}]}

Mängu statistika saamine

```
Päringu näide: curl -i -H "Accept: application/json" "<uri>/api/game/stats?game_id=3&user_id=23"
```

Päringu parameetrid:

- game_id - mängu identifitseeriv unikaalne identifikaator
- user_id – kasutajat identifitseeriv unikaalne identifikaator

Vastuse näide: [{"id": 1, "lat": "59.367141", "lon": "24.758450", "created_at": "2017-05-17T18:47:35.005675Z", "current_point": 57},...]

Uue vastuse lisamine

Päringu näide: curl --data "is_correct=1&point_id=1&user_game_id=2&problem_type_id=2&given_answer=sundial" <uri>/api/game/addanswer

Päringu parameetrid:

- is_correct – kas vastus on õige, kohustuslik
- point_id – punkti unikaalne identifikaator, kohustuslik
- user_game_id - mängu kasutaja kombot identifitseeriv unikaalne identifikaator, kohustuslik
- problem_type_id – probleemi tüübi unikaalne identifikaator, kohustuslik.
- given_answer – mängija poolt pakutud vastus, kohustuslik.

Vastuse näide: HTTP seisundi kood, 201 kui oli edukas ja 4xx kui midagi ebaõnnestus.

Sõnumi lisamine

Päringu näide: curl --data "content=thanks&user_game_id=12&user_id=2" <uri>/api/conversation/message

Päringu parameetrid:

- user_game_id – mängu kasutaja kombot identifitseeriv unikaalne identifikaator, kohustuslik.
- user_id - kasutaja unikaalne identifikaator, kohustuslik.
- content - sõnumi sisu, kohustuslik.

Vastuse näide: HTTP seisundi kood, 201 kui oli edukas ja 4xx kui midagi ebaõnnestus.

Lisa 3 – Administratsiooni paneeli üldvaade

Django administration WELCOME, **DEVELOPER**. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

[Home](#) » [Api](#)

Api administration

API	
Answers	+ Add 🔧 Change
Boundss	+ Add 🔧 Change
Check ins	+ Add 🔧 Change
Games	+ Add 🔧 Change
Messages	+ Add 🔧 Change
Point types	+ Add 🔧 Change
Points	+ Add 🔧 Change
Problem solutions	+ Add 🔧 Change
Problem types	+ Add 🔧 Change
User games	+ Add 🔧 Change
User roles	+ Add 🔧 Change
Users	+ Add 🔧 Change

Lisa 4 – Administratsiooni paneeli piirava ala lisamise vaade


Django administration WELCOME, DEVELOPER. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

[Home](#) · [Api](#) · [Boundss](#) · [Add bounds](#)

Add bounds

Game: Wonders of Saaremaa ▼ ✎ +

Notification text: You are entering private land. Please turn back

Mpoly: 

Delete all Features

Created at: **Date:** 2017-05-21 Today 📅
Time: 12:06:19 Now 🕒
Note: You are 3 hours ahead of server time.

Save and add anotherSave and continue editingSAVE