

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Ahti-Andreas Aasma 134910 IAPB

PETIFY: HÜBRIID-MOBIILIRAKENDUS LOOMAHOIU TEENUSE JAOKS

bakalaureusetöö

Juhendaja: Tanel Tammet
PhD

Tallinn 2018

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: Ahti-Andreas Aasma

21.05.2018

Annotatsioon

Käesoleva bakalaaurusetöö põhieesmärgiks oli luua hübriid-mobiilirakendus veebilehele Petify. Rakenduse koostamise aluseks olev veebilehekülg Petify.ee viib kokku loomaomanikud lemmiklooma hoidjatega. Lisaks hoiuteenusele on loomaomanikel võimalik leida oma lemmikule ka jalutaja. Lõputöö teiseks eesmärgiks oli võrdlus hübriid- ja platvormipõhise mobiilirakenduste vahel, ning võrdlus teiste sarnaste rakendustega.

Lõputöö käigus valmis Ionic raamistikul põhinev mobiilirakendus, mis võimaldab kasutajatel nutiseadmete vahendusel vaadata ja lisada kuulutusi ning ka omavahel suhelda. Lisaks rakenduse loomisele võrdles ja analüüsis töö autor töö teises pooles ka hübriid- ja platvormipõhiseid mobiilirakendusi.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 25 leheküljel, 4 peatükki, 13 joonist, 1 tabelit.

Abstract

Petify: hybrid mobile application for a pet sitting service

The main purpose of this thesis was to create a hybrid mobile application for website Petify. The website Petify.ee connects pet sitters with pet owners. In addition to pet sitting service, pet owners can also find a walker for their pet. The second purpose of this thesis was to compare hybrid and native mobile applications and also compare made application with similar applications already on the market.

In the course of making this thesis, a hybrid application based on Ionic framework was developed. Created mobile application allows users to post new listings, edit them and view as well as search other published listings. A chat feature was also implemented into the app. This gives customer the ability to contact other users easily.

In addition to developing a mobile application, the author of this thesis also compares and analyzes native and hybrid mobile applications.

The thesis is in Estonian and contains 25 pages of text, 4 chapters, 13 figures, 1 tables

Lühendite ja mõistete sõnastik

WP	<i>WordPress</i> on avatud lähtekoodiga ¹ tarkvara veebilehtede loomiseks ja haldamiseks
REST	<i>Representational State Transfer</i> on tarkvaraarhitektuuri stiil andmete saatmiseks ja vastu võtmiseks
API	<i>Application Programming Interface</i> ehk rakendus-/programmiliides
SDK	<i>Software Development Kit</i> on tarkvara arendamise tööriist
CLI	<i>Command-Line Interface</i> ehk käsurida
HTTP	<i>Hypertext Transfer Protocol</i> on protokoll teabe edastamiseks
JSON	<i>JavaScript Object Notation</i> on lihtsustatud andmevahetuse formaat
URL	<i>Uniform Resource Locator</i> on internetiaadress
Pistikprogramm	Pistikprogramm ehk <i>plugin</i> on tarkvaramoodul lisafunktsionaalsuse lisamiseks
Android	Google poolt loodud operatsioonisüsteem mobiilsetele seadmetele
iOS	Apple poolt loodud operatsioonisüsteem mobiilsetele seadmetele
GPS	<i>Global Positioning System</i> on asukoha määramiseks satelliitnavigatsiooni süsteem

¹ Avatud lähtekood (*open source*) – lähtekood on kõigile vabalt kättesaadav ja muudetav

Sisukord

1 Sissejuhatus	10
2 Taust	12
2.1 Sarnased valmisolevad rakendused	12
2.1.1 Rover	12
2.1.2 Fiverr	13
3 Rakenduse kirjeldus.....	16
3.1 Sisselogimise vaade.....	16
3.2 Registreerimise vaade.....	17
3.3 Parooli taastamise vaade.....	18
3.4 Valikute vaade	19
3.5 Päis, jalus ja menüü	21
3.5.1 Päis	21
3.5.2 Jalus	21
3.5.3 Menüü.....	21
3.6 Otsingu vaade	22
3.7 Kuulutuse vaade	23
3.8 Minu kuulutuste vaade.....	24
3.9 Kuulutuse lisamise vaade	25
3.10 Profiili vaade.....	26
3.11 Vestluste vaade	27
4 Kasutatud tehnoloogiad	28
4.1 Node.js.....	28
4.2 Apache Cordova	29
4.3 Ionic	29
4.3.1 Ionic Native	29
4.4 Angular	30
4.5 WordPress REST API	30
4.5.1 JWT Authentication for WP REST API.....	30
4.6 Firebase.....	30

5 Hübriid- ja platvormipõhise rakenduse võrdlus Petify näitel	32
5.1 Rakenduse laadimine	32
5.2 Piltide kuvamine	32
5.3 Vaadete vahel liikumine	33
5.4 Google Maps.....	33
5.5 Tulemus	33
6 Kokkuvõte	34
Kasutatud kirjandus	35
Lisa 1 – Vaadete vahel liikumise logi (hübriid vasakul)	37
Lisa 2 – Google Maps'i laadimine (hübriid vasakul)	38

Jooniste loetelu

Joonis 1. Rover.	13
Joonis 2. Fiverr.	14
Joonis 3. Sisselogimise vaade.	17
Joonis 4. Registreermise vaade.	18
Joonis 5. Parooli taastamise vaade.	19
Joonis 6. Valikute vaade.	20
Joonis 7. Päis, jalus ja menüü.	21
Joonis 8. Otsingu vaade.	22
Joonis 9. Kuultuse vaade.	23
Joonis 10. Minu kuulutuste vaade.	24
Joonis 11. Kuulutuse lisamise vaade.	25
Joonis 12. Profiili vaade.	26
Joonis 13. Vestluste vaade.	27

Tabelite loetelu

Tabel 1. Rakenduste võrdlus.	15
-----------------------------------	----

1 Sissejuhatus

Käesoleva lõputöö raames valmis hübriid-mobiilirakendus Petify. Põhiline eesmärk oli luua kasutajasõbralik rakendus iOS ja Android operatsioonisüsteemidele. Teema sai valitud sellepärast, et autor on mõnda aega tegelenud veebilehe Petify.ee arendamisega ning veebilehe pakutavaid teenuseid kasutab üha suurem hulk inimesi. Sellega seoses tekkis plaan tõsta kasutajamugavust ning luua veebilehele korralik rakendus. Google poolt loodud kodulehe analüüsimise tööriista Google Analyticsi¹ andmete põhjal jagunevad Petify.ee lehe kasutajad praktiliselt pooleks nutiseadmete ja arvuti kasutajate vahel. Samuti on üsna võrdselt Android ja iOS platvormi kasutajaid.

Autor valis hübriidrakenduse, kuna on sellega juba eelnevalt kokku puutunud ja selle abil saab kirjutada korraga nii Android kui ka iOS platvormile. Hübriidrakendused näevad välja ja käituvad nagu platvormipõhised rakendused nii iOS'il kui ka Android'il. Seda kõike kirjutades vaid ühte koodi ja kasutades keeli nagu TypeScript, CSS ja HTML. Platvormipõhist funktsionaalsust aitavad teostada Cordova pistikprogrammid.

Bakalaurusetöö põhieesmärgiks on saada valmis töötav rakendus, mille vahendusel saavad kliendid kuulutusi vaadata, kuulutusi lisada ning omavahel suhelda. Ühtlasi on ka eesmärgiks rakendus poodi üles laadida juuni alguseks. Lisaks sellele võrdleb autor enda tehtud rakendust juba olemasolevate sarnaste rakendustega. Teiseks eesmärgiks on võrrelda hübriid- ja platvormipõhist rakendust, tehes ka minimaalse platvormipõhise rakenduse funktsionaalsuse võrdluse.

Teises peatükis kirjeldab autor põhjalikumalt rakenduse tausta ning tutvustab juba turul olevad sarnaseid rakendusi. Kolmandas peatükis kirjeldab autor valmisolevat rakendust koos vaadete joonistega. Neljandas peatükis tutvustab autor rakenduseks vajaminevad tehnoloogiad. Kuuendas peatükis teeb autor minimaalse funktsionaalsusega platvormipõhise rakenduse ja võrdleb seda enda hübriidrakendusega. Täpsemalt võrdleb

¹ Google Analytics – Google poolt loodud kodulehe analüüsimise tööriist

autor rakenduse laadimist, vaadete vahel liikumist, Google Maps'i laadimist ja piltide kuvamist. Lisaks toob välja ka saadud tulemuse.

2 Taust

Petify.ee on 2015. aastal autori poolt valmis tehtud veebileht, mis pakub inimestele uutset ja ainulaadset keskkonda loomade hoiustamiseks. Kasutajatel on veebilehega liitudes võimalus loomi hoiule võtta või oma lemmik hoiule anda. Petify eesmärk on luua avatud ja sõbralik kogukond, mille abil loomaomanike elu mugavamaks muuta.

Portaali loomise idee tekkis otsesest vajadusest. Suheldes loomaomanikest sõprade ja tuttavatega selgus, et reisile minnes on raske lemmikule hoidjat leida. Arutelu käigus tuli välja, et turul valitseb auk ja nõudlus portaali järele on olemas. Peaaegu kolme tegevusaastaga on keskkonnas üle 100 kuulutuse ja loomahoidjaid võib leida pea igast suuremast Eesti linnast. Veebilehe kasutajaid on ka Eestist väljapool, nimelt on Petify.ee keskkonnaga on liitunud ka üks Soomes elav eestlane. Keskmiselt käib Google Analytics'i andmetel veebilehel ~900 unikaalset külastajat kuus ning keskmiselt lisatakse paar uut kuulutust nädalas.

2.1 Sarnased valmisolevad rakendused

Lisaraha teenimiseks on tehtud palju erinevaid rakendusi – nt. Taxify¹ ja Uber². Antud lõputöös uurib autor Petify'le sarnaseid kuulutuste lisamise ja otsmise rakendusi. Rakenduste funktsionaalsuste erinevusi võrreldakse tabelis 1.

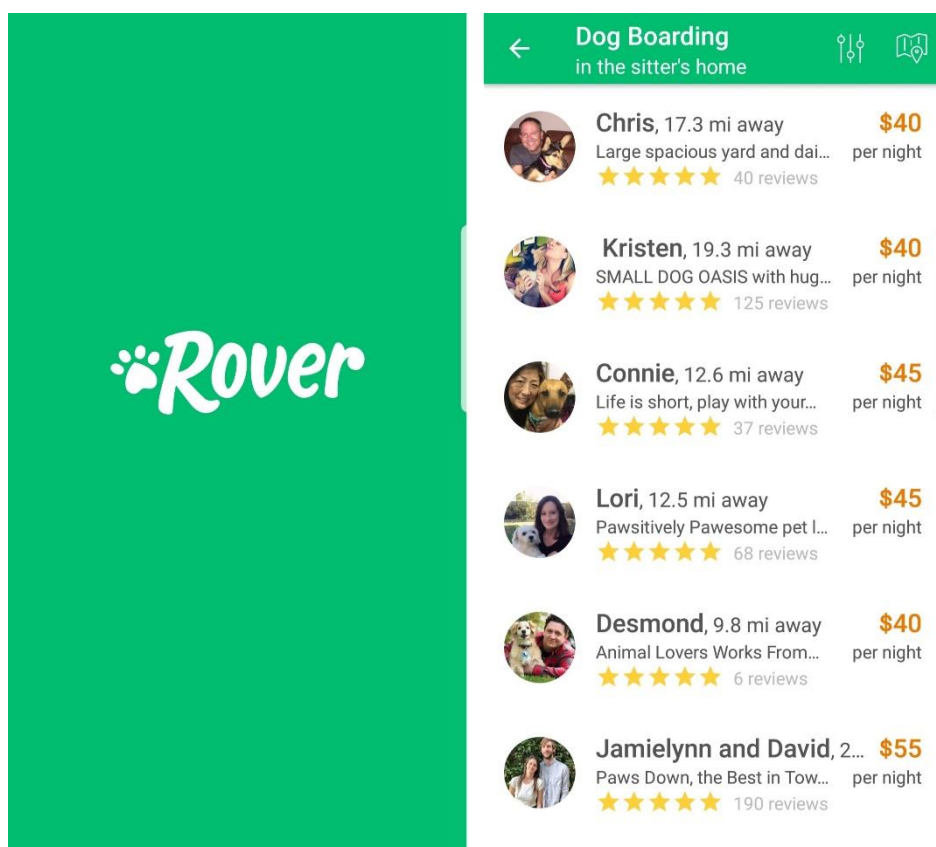
2.1.1 Rover

Rover (Joonis 1) on Ameerika Ühendriikide ettevõtte, mis vahendab koerte hoidmise ja jalutamise teenust. Sarnaselt Petify'le saavad kasutajad pakkuda ja otsida koertega jalutamise teenust. Vahendustasu võetakse iga eduka broneeringu pealt. Rakendus toimib ainult Ameerika Ühendriikide piires. Rover toimib nii veebilehena kui ka iOS ja Android rakendusena [1]. Järgnevas lõigus võrdleb autor Rover'i mobiilirakendust selle lõputöö raames valmis saanud mobiilirakendusega.

¹ <https://taxify.eu/et/>

² <https://www.uber.com/et-EE/>

Sisselogimine on analoogne Petify'ga – kas e-posti aadressi kaudu või Facebook'i vahendusel. Petify rakendusel on lisaks veel Google'i kaudu sisselogimine. Edasi suunatakse kasutaja kuulutuste otsimise vaatesse, mis küsib kasutajalt õigust saada telefoni asukoht, et seeläbi näidata kasutajale lähedal asuvaid pakkumisi. Hetkel see võimalus Petify rakenduses puudub. Seejärel on kasutajal võimalik otsida kuulutusi kasutades filtrit või otsides kaardilt. Viimast pole antud Petify versioonis kasutusel. Sarnaselt Petify'le saab Rover'i rakenduses hallata oma profiili, lisada kuulutusi, vaadata kuulutusi ning oma kuulutusi hallata.



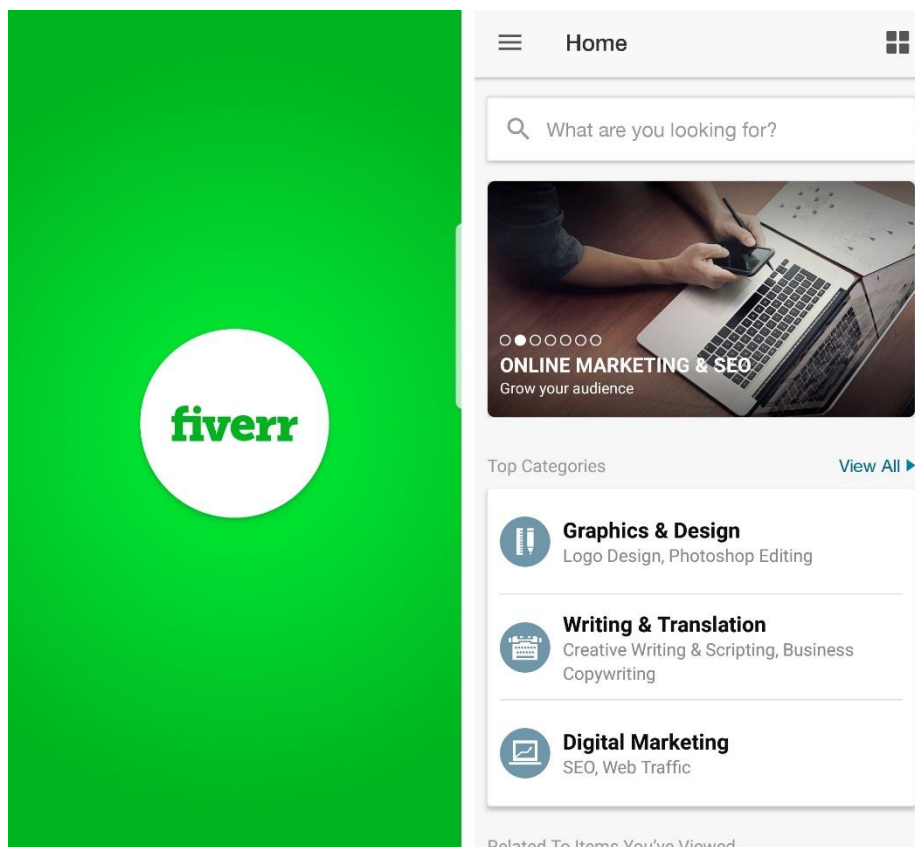
Joonis 1. Rover.

2.1.2 Fiverr

Fiverr (Joonis 2) on üks maailma suurimaid digitaalsete teenuste vahendajaid. Müüjad saavad luua kuulutusi ja sellele ise hinna määrata. Samalaadselt Rover'ile võtab Fiverr igalt kuulutuselt vahendustasu. Rakendusel on palju erinevaid kategooriaid - disainist kirjutamiseni ja ärist elustiilini. Pakutud teenused on näiteks logo kujundamine ja

konsultatsioon erinevates valdkondades [2]. Järgnevas lõigus võrdleb autor Fiverr'i mobiilirakendust selle lõputöö raames valmis saanud mobiilirakendusega.

Rakenduse avamisel avaneb sisselogimise vaade, mille erinevus Petify'ga on see, et kuulutuste vaatamiseks ei pea sisse logima, vaid sellest vaatest saab mööda minna. Pakkumiste otsing käib läbi kategooriate või kasutajate nimede. Mõlemad on ka Petify rakenduses olemas. Rakenduses puudub kasutaja haldamine ja kuulutuste lisamine.



Joonis 2. Fiverr.

Funktsionaalsus	Petify	Rover	Fiverr
Facebook'i sisselogimine/registreerimine	+	+	+
Google'i sisselogimine/registreerimine	+		+
E-posti sisselogimine/registreerimine	+	+	+
Parooli taastamine	+	+	+
Kuulutuste otsing	+	+	+
Kuulutuse otsing kasutaja asukoha järgi		+	
GPS kasutamine		+	
Kuulutuse lisamine/haldamine	+	+	
Profiili haldamine	+	+	
Piltide lisamine/kuvamine	+	+	+
Videote lisamine/kuvamine			+
Kaardi kuvamine	+	+	
Kaardi järgi otsimine		+	
Vestlus teiste kasutajatega	+		+
Maksmiskeskus		+	+

Tabel 1. Rakenduste võrdlus.

3 Rakenduse kirjeldus

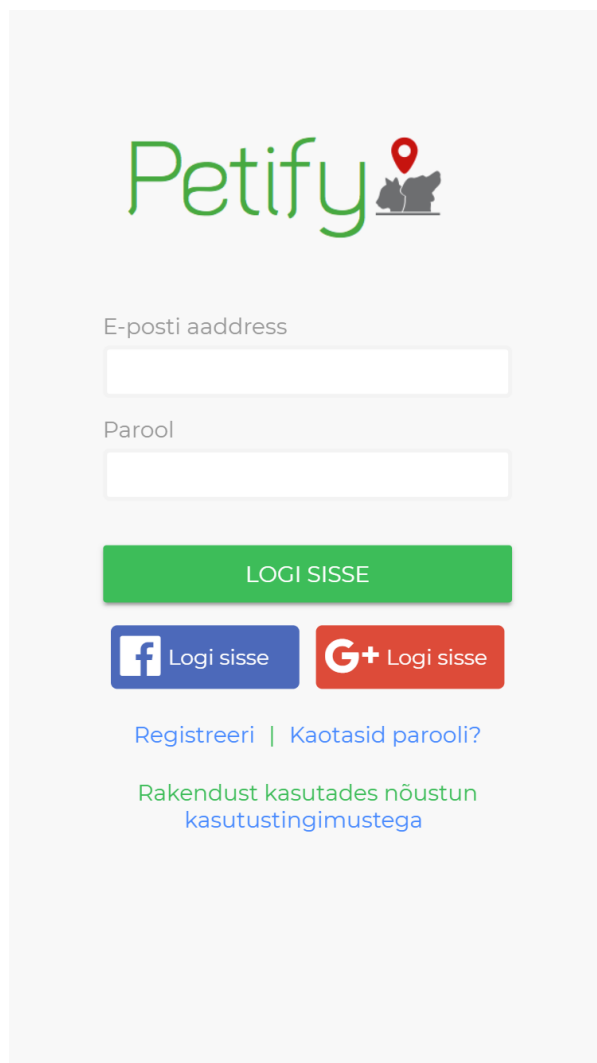
Käesolevas peatükis kirjeldatakse rakenduse funktsionaalsust koos joonistega. Petify mobiilirakendus on mõeldud inimestele, kes otsivad või pakuvad loomahoitu ja jalutamise teenust. Kuna veebileht on juba olemas ja see on tehtud WordPress'i põhjal, siis toimub kõik läbi selle. Rakenduse ja serveri omavaheliseks suhtluseks kasutatakse WP REST API't. Kõikide vormide väljad valideeritakse nii rakenduses kui ka serveri poolel.

Rakendus koosneb 11. vaatest ning igal vaatel on oma HTML, TS ja SCSS failid. Hübridrakenduse minimaalne Androidi versioon on 4.4 ja iOS versioon 10.

3.1 Sisselogimise vaade

Rakenduse esmasel avamisel kuvatakse vaade (Joonis 3), kus kasutajal on võimalik sisse logida. Selleks saab kasutada kas juba eelnevalt olemasolevat kontot või logida sisse kasutades Facebook'i või Google'i kontot. Kasutaja puudumisel on olemas võimalus registreerida vajutes vastavat nuppu, mis viib kasutaja teise vaatesse (Joonis 4). Sama funktsionaalsus kehtib ka parooli kaotamise korral, mis viib parooli taastamise vaatesse (Joonis 5). Edukal sisselogimisel suunatakse kasutaja edasi valikute vaatesse (Joonis 6). Kui kasutaja logib ennast sisse, sulgeb rakenduse ja avab selle uuesti, siis suunatakse ta koheselt otsingu vaatesse (Joonis 8).

Sisselogimine toimub läbi WP REST API ja „JWT Authentication for WP REST API“ pistikprogrammi. Edukal sisenemisel antakse kasutajale võti, mis pannakse igale järgnevale päringule kaasa.

The image shows a login page for 'Petify'. At the top center is the logo 'Petify' in green, with a red location pin icon and a grey silhouette of a dog and a cat to its right. Below the logo are two white input fields: the first is labeled 'E-posti aadress' and the second is labeled 'Parool'. Underneath these fields is a green button with the text 'LOGI SISSE'. Below the green button are two social media login buttons: a blue one with the Facebook 'f' logo and the text 'Logi sisse', and a red one with the Google+ 'G+' logo and the text 'Logi sisse'. Below these buttons is a link 'Registreeri | Kaotasid parooli?' in blue. At the bottom, there is a green link 'Rakendust kasutades nõustun kasutustingimustega'.

Joonis 3. Sisselogimise vaade.

3.2 Registreerimise vaade

Registreerimise vaates (Joonis 4) on kasutajal võimalik luua uus konto. Kui kasutaja otsustas sisse logida kas Facebook'i või Google'i kaudu, siis luuakse talle konto automaatselt. Kasutaja loomisel on kohustuslik täita kõik vormi väljad, nende puudumise korral näidatakse vastavat veateadet. Edukal registreerimisel saadetakse kasutajale sellekohane e-post ning suunatakse sisselogimise vaatesse (Joonis 3), kus ta saab äsja loodud kontot kasutada.

Eesnimi*

Perekonnanimi*

E-posti aadress*

Parool*

Parool uuesti*

REGISTREERI

[Kasutaja olemas? Logi sisse](#)

Rakendust kasutades nõustun [kasutustingimustega](#)

Joonis 4. Registreermise vaade.

3.3 Parooli taastamise vaade

Parooli taastamise vaates (Joonis 5) saadetakse kasutajale e-posti aadressi sisestamisel vastav meil, mille kaudu saab ta parooli taastada.

E-posti aadress

Lähtesta parool

Kasutaja olemas? [Logi sisse](#)

Joonis 5. Parooli taastamise vaade.

3.4 Valikute vaade

Esmakordsel sisselogimisel kuvatakse kasutajale valikute vaade (Joonis 6) ning 2 kaarti. Vajutades esimese kaardi peale suunatakse kasutaja minu kuulutused vaatesse (Joonis 10), teise kaardi korral otsingu vaatesse (Joonis 8).



Vali kategooria:

Olen hoidja/jalutaja

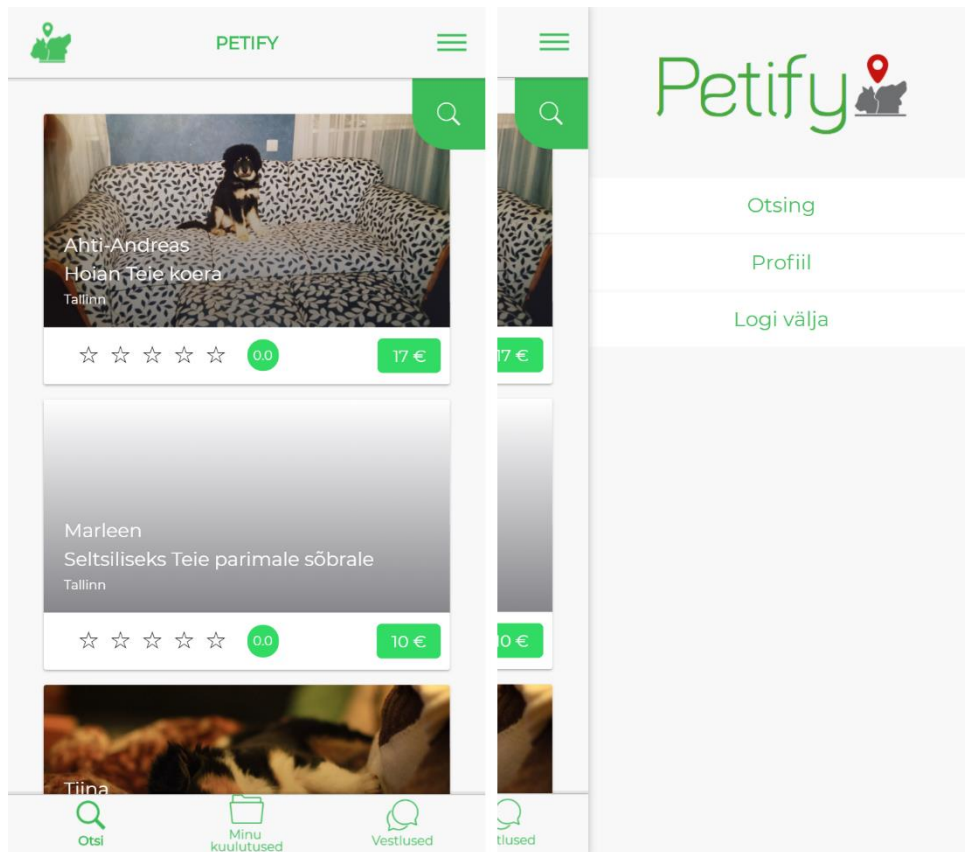
Pakun lemmikloomade hoidmise või jalutamise teenust

Otsin hoidjat/jalutajat

Otsin oma lemmikloomale hoidjat või jalutajat

Joonis 6. Valikute vaade.

3.5 Pääs, jalus ja menüü



Joonis 7. Pääs, jalus ja menüü.

3.5.1 Pääs

Pääses (Joonis 7) kuvatakse Petify logo, tiitel ning nupp menüü avamiseks. Logole vajutades suunatakse kasutaja otsingu vaatesse (Joonis 8).

3.5.2 Jalus

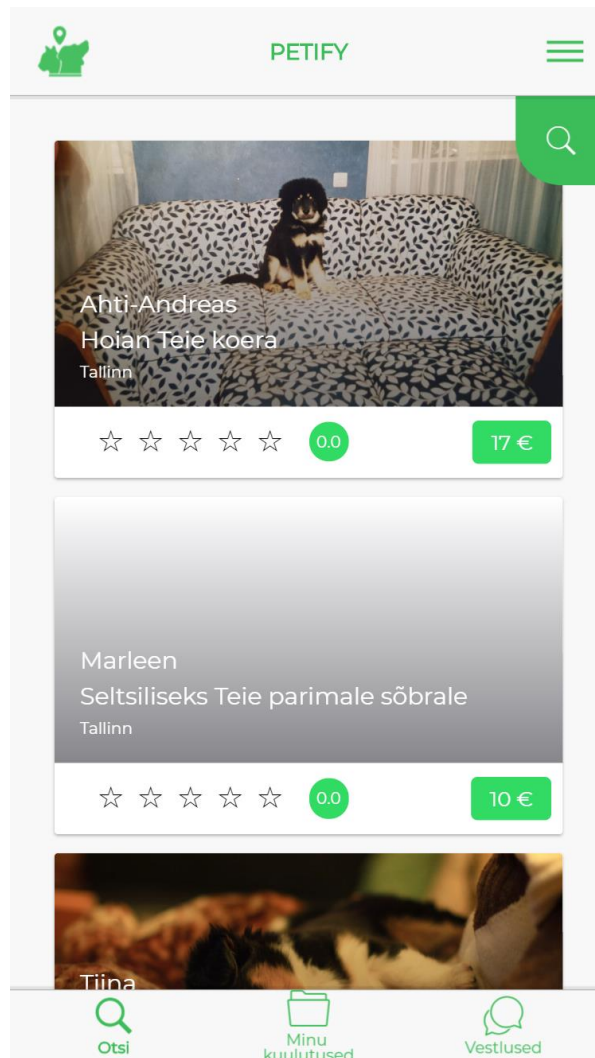
Jaluses (Joonis 7) on 3 nupp, mis suunavad vastavatesse vaadetesse.

3.5.3 Menüü

Menüü (Joonis 7) avanedes on kasutajal võimalik minna otsingu vaatesse (Joonis 8), enda profiili vaatesse (Joonis 12) ning välja logida. Välja logimisel suunatakse kasutaja sisselogmise vaatesse (Joonis 3).

3.6 Otsingu vaade

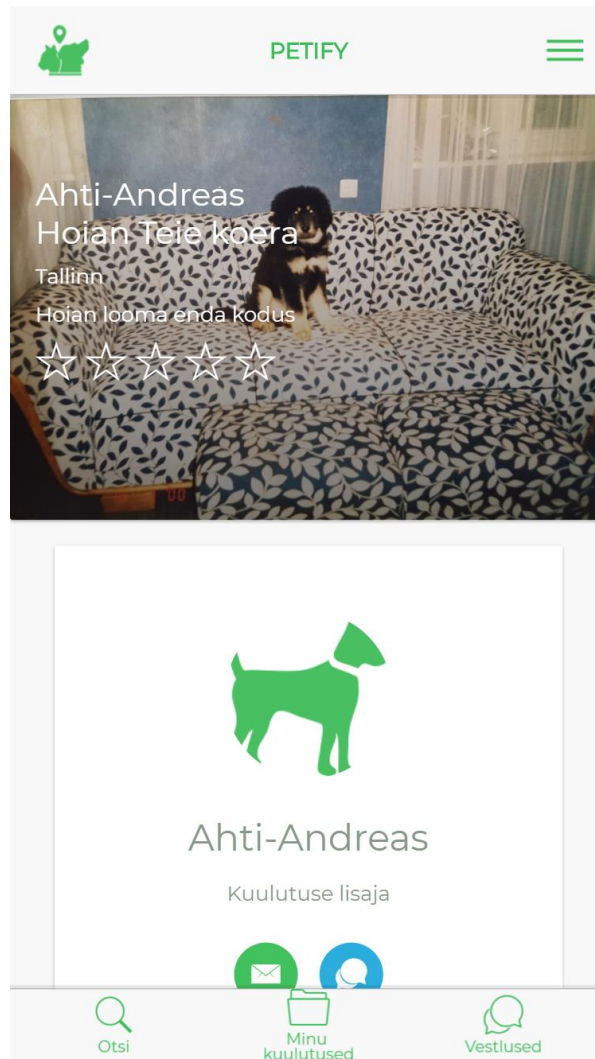
Otsingu vaatesse (Joonis 8) sisenedes kuvatakse kasutajale viimati lisatud kuulutused viie kaupa. Alla kerides tekib nupp, mida vajutades laetakse rohkem kuulutusi. Kuulutusi kuvatakse kaartidena kasutades Angular'i direktiivi ***ngFor**. Igal kaardil on küljes kuulutuse lisaja nimi, tiitel, asukoht, reiting ja hind, mille kaudu on kasutajal lihtne leida endale sobiv kuulutus. Lisaks sellele on vaate üleval paremal nurgas otsingu nupp, mis avab filtri. Filtris on 3 välja, mille kaudu saab kuulutusi täpsemalt otsida. Nendeks väljadeks on märksõna, asukoht ja kategooria.



Joonis 8. Otsingu vaade.

3.7 Kuulutuse vaade

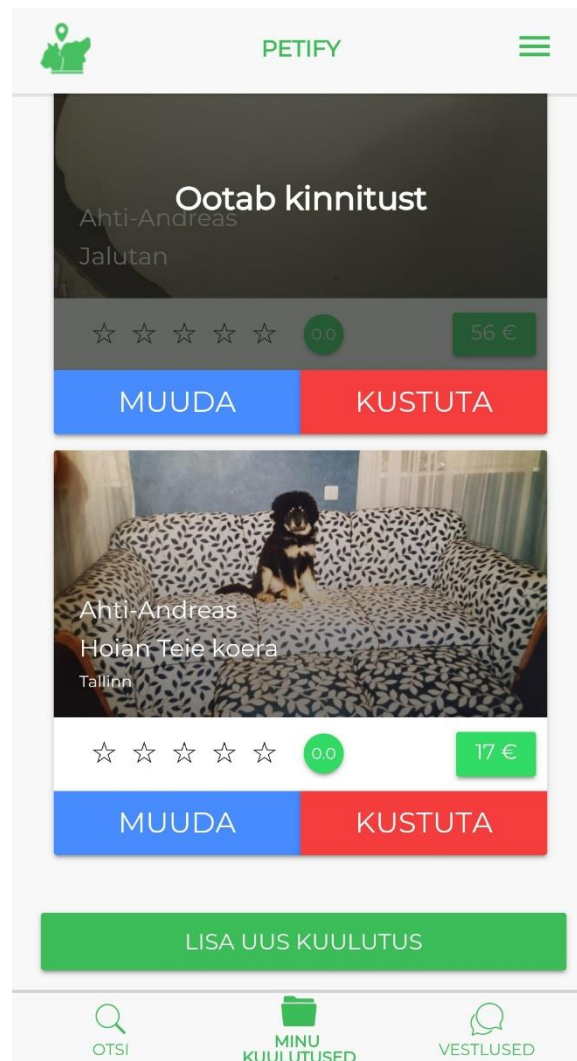
Kuulutuse vaates (Joonis 9) kuvatakse kuulutuse detailid ja kuulutuse autori info. Vajutades kuulutuse autori nime all olevaid nuppe on võimalik autorile saata kas e-posti kiri või vestlus rakenduse sees. E-posti nupule vajutades avaneb kasutajal vorm, mille täites saadetakse autorile kiri.



Joonis 9. Kuulutuse vaade.

3.8 Minu kuulutuste vaade

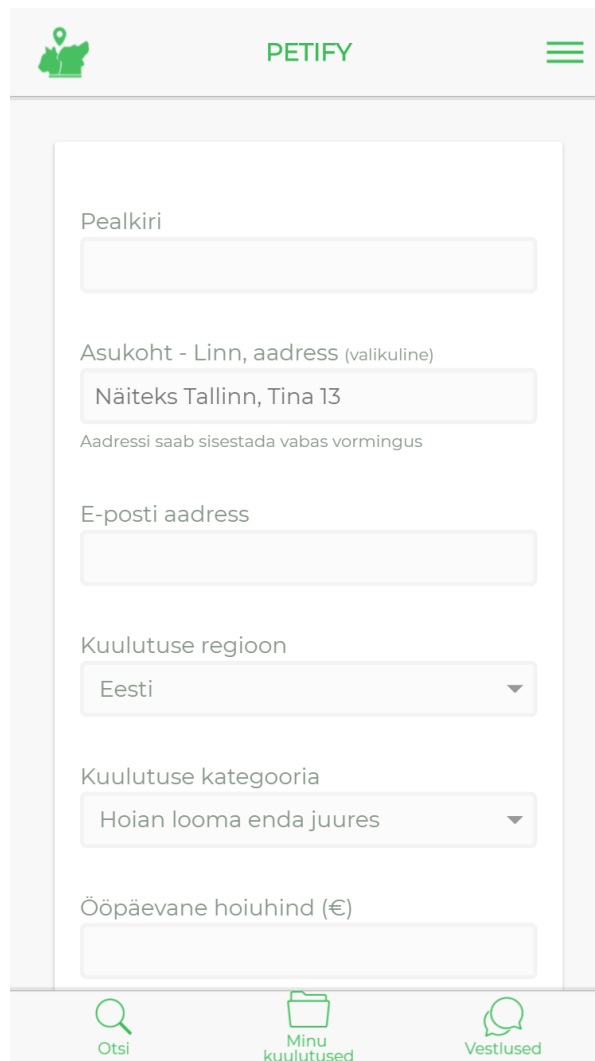
Minu kuulutuse vaates (Joonis 10) kuvatakse kasutajale enda kuulutused. Vajadusel saab olemasolevaid kuulutusi kas muuta või kustutada. Muutmisel suunatakse kasutaja kuulutuse lisamise vaatesse (Joonis 11), kus juba olemasolevad väljad on täidetud. Uue lisamisel on kõik väljad tühjad.



Joonis 10. Minu kuulutuste vaade.

3.9 Kuulutuse lisamise vaade

Kuulutuse lisamiseks tuleb täita kõik väljad. Peale vormi täitmist valideeritakse kõik väljad nii rakenduses kui ka serveri poolel. Kuulutuse edukal lisamisel suunatakse kasutaja tema kuulutuste vaatesse (Joonis 10). Kuulutus ei muutu kohe aktiivseks, vaid selle peab administraator aktiveerima.



The screenshot displays the 'PETIFY' mobile application interface for creating a new advertisement. The app's header includes a location pin icon, the brand name 'PETIFY', and a menu icon. The main content area contains several input fields and dropdown menus:

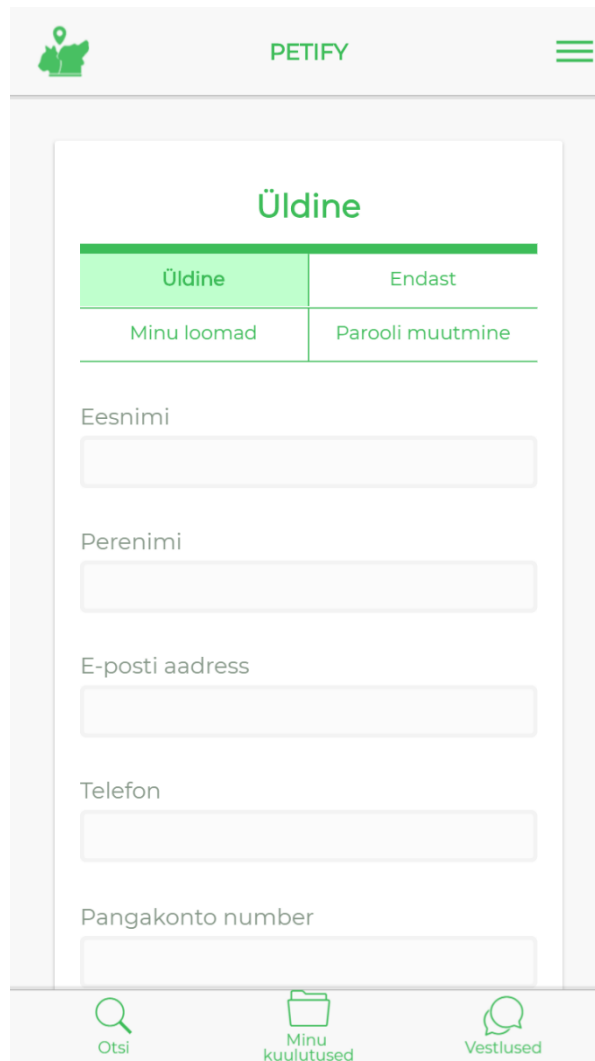
- Pealkiri**: A text input field for the advertisement title.
- Asukoht - Linn, aadress (valikuline)**: A text input field with the example text 'Näiteks Tallinn, Tina 13'. Below it, a note states 'Aadressi saab sisestada vabas vormingus'.
- E-posti aadress**: A text input field for the user's email address.
- Kuulutuse regioon**: A dropdown menu currently set to 'Eesti'.
- Kuulutuse kategooria**: A dropdown menu currently set to 'Hoian looma enda juures'.
- Ööpäevane hoiuhind (€)**: A text input field for the daily fee.

The bottom navigation bar features three icons: a magnifying glass labeled 'Otsi', a folder icon labeled 'Minu kuulutused', and a speech bubble icon labeled 'Vestlused'.

Joonis 11. Kuulutuse lisamise vaade.

3.10 Profiili vaade

Profiili vaates (Joonis 12) on kasutajal võimalik täita endast üldine info. Lisaks saab ka lisada enda loomi ning muuta enda parooli. Kõik info, mida kasutaja kirjutab „Endast“ ja „Minu loomad“ vaate all, läheb iga tema poolt lisatud kuulutusega kaasa.



Üldine	Endast
Minu loomad	Parooli muutmise

Eesnimi

Perenimi

E-posti aadress

Telefon

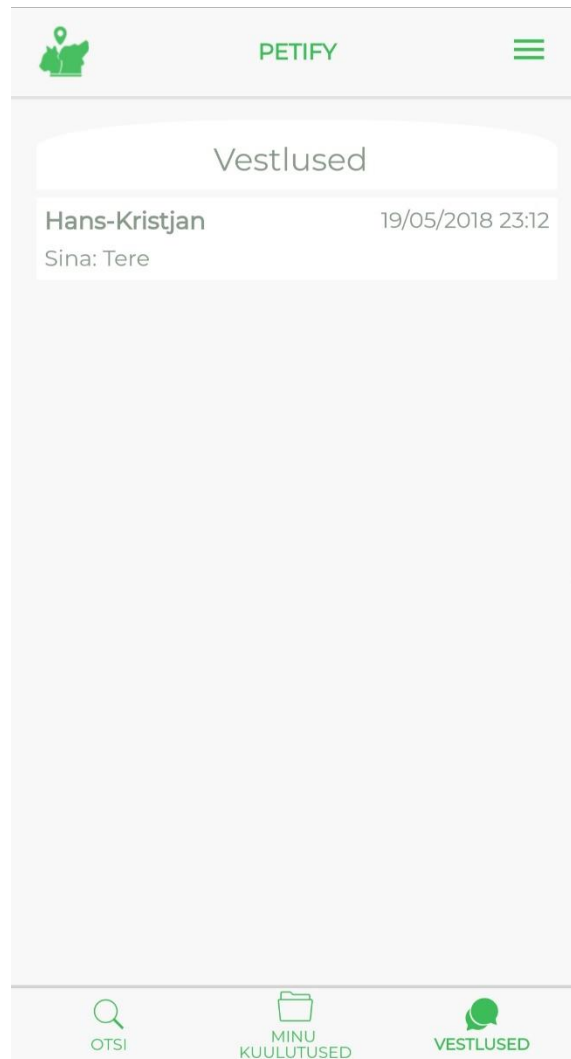
Pangakonto number

Otsi Minu kuulutused Vestlused

Joonis 12. Profiili vaade.

3.11 Vestluste vaade

Vestluse vaates (Joonis 13) kuvatakse kõik kasutaja vestlused teiste kasutajatega. Vajutades vestluse peale suunatakse kasutaja selle vestluse vaatesse, mis toimub reaalajas kasutades Firebase'i *Realtime database*'i.



Joonis 13. Vestluste vaade.

4 Kasutatud tehnoloogiad

Antud hübriidrakendus on tehtud Ionic raamistikul. Kuna tegemist on juba WordPress'i¹ peal oleva veebilehega, siis kasutas autor serveriga suhtluseks WP REST API't. See on WordPress'i sisse ehitatud REST põhimõttel töötav rakendusliides, mille abil saab väljaspool WordPress'i kasutada veebilehe funktsionaalsust nagu kuulutuse kuvamine, lisamine ja kustutamine. Rakenduses kuulutuste lisamiseks ja muutmiseks on vaja sisse logida, milleks kasutas autor pistikprogrammi JWT Authentication. Ionic raamistik on ehitatud Cordova peale Angular'i abiga ehk Ionic raamistik kasutab Angular'i, mis on TypeScript'i² põhine raamistik. Rakenduse arendamiseks on ka vajalik pakettide haldurit npm, mis tuleb node.js'ga kaasa.

Autor kasutas nii piltide laadmiseks kui ka Google kaardi kuvamiseks pistikprogramme, mille abil on telefoni funktsionaalsuse kasutamine lihtne ja ei nõua autori poolt palju koodi kirjutamist. Pistikprogrammid aitavad TypeScript koodi abil suhelda telefoni funktsionaalsusega. Reaalajas vestlusteks kasutas autor Firebase'i andmebaasi (inglise keeles *Realtime database*). Seda oli lihtne integreerida enda süsteemi ning seda arendatakse pidevalt edasi Google poolt.

4.1 Node.js

Node.js on avatud lähtekoodiga JavaScript'i käitussüsteem, mis käivitab JavaScript koodi serveri poolel. See kasutab mitteblokeerivat asünkroonset I/O mudelit, mis tähendab seda, et Node.js ei oota kunagi ühegi päringu taga, vaid võtab järgmise päringu ette. [3] [4]

Node.js installeerimisega tuleb kaasa ka npm, mis on autori rakenduses vajalik. Npm on pakettide haldur, mille abil on lihtne pakette luua ja alla tõmmata ning seejärel ka uuendada. Seda kõike saab teha käsurea kaudu. Käesoleva töö käigus kasutas autor npm'i selleks, et laadida alla nii Cordova kui ka Ionic. [5]

¹ WordPress – avatud lähtekoodiga tarkvara veebilehtede loomiseks ja haldamiseks [18]

² TypeScript – avatud lähtekoodiga programmeermiskeel, mis võimaldab lisada staatilist tüübikirjeldust ning mis kompileerub JavaScript'iks [19]

4.2 Apache Cordova

Apache Cordova (edaspidi Cordova) on avatud lähtekoodiga mobiilirakenduste arendamise raamistik. Selle abil saab luua hübriid-mobiilirakendusi kasutades HTML'i, CSS'i ja JavaScript'i. Cordova võimaldab kirjutada rakendusi mitmele platvormile korraga. [6]

Cordova ja riistvara funktsionaalsuse suhtluseks on vaja pistikprogramme, mille abil on võimalik platvormipõhist kood välja kutsuda JavaScript'iga. Lisaks ametlikele pistikprogrammidele on ka palju kolmandate osapoolte poolt välja antud pistikprogramme. [6]

4.3 Ionic

Ionic on avatud lähtekoodiga SDK hübriid-mobiilirakenduste loomiseks. Ionic rakendusi arendatakse Ionic'u käsurea tööriista (inglise keeles *Command-Line Interface*) abil ning läbi Cordova saab kirjutatud koodist lõpuks platvormipõhine rakendus. Ionic on tasuta ja juba olemasolevate Ionic mallidega on väga lihtne arendama hakata. Lihtsamaks teeb ka aktiivne kommuun ja põhjalik dokumentatsioon. Autor valis Ionic'u, kuna on sellega enne kokku puutunud ja teab, et selle abil saab luua väga platvormipõhise rakenduse sarnase hübriidrakenduse. [7] [8]

Esimene Ionic'u versioon kasutas AngularJS'i, hilisemad versioonid aga Angular'i. Arendamiseks kasutatakse CSS'i, HTML5'e, Sass'i¹ ja Angular'i, mis on TypeScript'i põhine. Ionic koosneb juba olemasolevatest UI komponentidest, mis lihtsustavad ja kiirendavad rakenduse loomist. [7] [8]

4.3.1 Ionic Native

Ionic Native on TypeScript'i *wrapper* Cordova pistikprogrammidele, mis teeb platvormipõhise funktsionaalsuse lisamise rakendusele väga lihtsaks. Autor kasutas Ionic Native pistikprogrammide näiteks kaamerat, failide üleslaadimist ja Google kaardi laadimist. [7]

¹ Sass (*Syntactically Awesome Style Sheets*) – laiendab CSS'i

4.4 Angular

Angular (teatakse ka kui „Angular 2“ või „Angular 5“) on TypeScript'i põhine avatud lähtekoodiga veebi raamistik. Angular'i kasutatakse rakenduse poolel ning keelteks on HTML ja TypeScript. Angular koosneb komponentidest, mis defineerivad vaateid (*views*) ja kasutavad teenuseid (*services*). Teenused annavad spetsiifilise funktsionaalsuse. Nii komponendid kui ka teenused on lihtsalt klassid. Vaadete vahel navigeerimiseks on olemas *Router service*. [9]

4.5 WordPress REST API

WordPress'ist on saamas täielik rakenduse raamistik. WordPress REST API annab kasutajale HTTP näitajad, mille abil on võimalik saada veebilehe andmed lihtsas JSON formaadis. Infot kättesaamine ja muutmine toimub lihtsa HTTP päringu kaudu, mis tähendab seda, et selle saab välja kutsuda kasutades mistahes programmeerimiskeelt. Autor valis just WP REST API, kuna WordPress'i peal oli juba veebileht olemas ning see võimaldas autoril lihtsa vaevaga kätte saada kõik juba veebilehel olevad andmed. [10] [11]

POST, PUT ja DELETE päringute tegemiseks on vajalik autentimine. Seda saab teha näiteks küpsistega (inglise keeles *cookie*) või kasutades pistikprogramme. Autor valis selleks WordPress'i pistikprogrammi „JWT Authentication for WP REST API“. [11]

4.5.1 JWT Authentication for WP REST API

Antud pistikprogramm kasutab autentimiseks turvalist *JSON Web Tokens* (edaspidi JWT) meetodit. JWT on JSON'i põhine avatud standard RFC 7519 juurdepääsu võtmete loomiseks. JWT on kompaktne, seega saab seda saata läbi URL'i, POST parameetrite või HTTP päise sees. [12] [13]

4.6 Firebase

Firebase on Google'i arendamise raamistik mobiili- ja veebirakendustele. Firebase pakub arendajatele palju erinevaid teenuseid. Nendeks on näiteks analüütika, autentimine, pilveteenused ja reaalaajas andmebaas. [14]

Autor kasutas lõputöös valminud rakenduses reaalajas andmebaasi, mis salvestab andmed Firebase'i pilve andmebaasi ja võimaldab andmeid reaalajas kasutajate vahel sünkroonida.

5 Hübriid- ja platvormipõhise rakenduse võrdlus Petify näitel

Autor on tegelema mitme erineva hübriidrakenduse arendamisega üle ühe aasta ja on märganud Ionic raamistiku pidevat arengut. Seda toetab Ionic'u uute versioonide pidevat väljalaskmist [15]. Lisaks arendatakse pidevalt edasi ka Angular'i ja Cordova't [16] [17]. Samuti kasvab pistikprogrammide arv pidevalt. Sellest tulenevalt võib väita, et hübriidrakenduste populaarsus on kasvamas.

Hübriid- ja platvormipõhise rakenduse võrdluseks tegi autor minimaalse funktsionaalsusega Android'i rakenduse Java's. Funktsionaalsus, mida testiti, oli rakenduse laadimine, pildi saamine mobiilist, vaadete vahel liikumine ja Google Maps'i kuvamine. Testimiseks kasutati mobiiltelefoni Samsung S8 (Android'i versioon 8.0.0).

5.1 Rakenduse laadimine

Esmalt testis autor rakenduste käivitamise aegu. Kümne testimise käigus selgus, et platvormipõhisel rakendusel võttis täielik laadimine aega keskmiselt 530ms. Hübriidrakendusel kulus selleks keskmiselt 5560ms. Siinkohal jääb Ionic'u hübriidrakendus selgelt alla. Ionic peab rakenduse käivitamise ajal enne enda komponendid ära laadima. Pikk laadimisaeg on lahendatud sellega, et kasutajale kuvatakse ootamise ajal arendaja poolt valitud taustapilti *Splash Screen*. Ionic rakenduse lõplik versioon laetakse käsurealt kasutades käskude `--prod --release`, mis optimeerib ja pakib koodi kokku. Tänu sellele vähenes keskmine laadimise aeg ~2 sekundit ehk rakendus võttis laadimiseks aega keskmiselt 3740ms.

5.2 Piltide kuvamine

Piltide saamiseks kasutajalt ja seejärel serverisse üleslaadimiseks oli vaja mitme pistikprogrammi abi ja siis need omavahel ühendada. Platvormipõhine rakendus nõudis selleks vähem koodi.

5.3 Vaadete vahel liikumine

Vaadete vahel liikumine toimub nii hübriid kui ka platvormipõhises rakenduses silmapilkselt ja visuaalselt on raske neid eristada. Hübriidrakenduses kasutas autor nupule vajutamisel ja lehe laadimisel koodi `new Date().getTime()` ja lahutas need omavahel. Kümne katsetuse korral saadi keskmiseks tulemuseks 6ms. Platvormipõhise rakenduse korral kasutas autor koodi `Calendar.getInstance().getTimeInMillis()`. Keskmise tulemus oli 58ms. Hübriidrakendus laadis vaate kiiremini, aga visuaalselt oli erinevust raske tuvastada. Logi on saadaval Lisas 1.

5.4 Google Maps

Google Maps laadimine on rakendustes erinev. Hübriid-mobiilirakendus vajab selleks eraldi pistikprogrammi tõmbamist. Koodi tuli aga kirjutada mõlema puhul vähe. Kaardi kuvamiseks kulunud aja sai autor nii, et esmalt mõõtis vaate laadimiseks kulunud aja ja seejärel lahutas sellest kaardi laadimiseks kulunud aja. Hübriidrakenduses kasutati sarnaselt eelmiste testidega koodi `new Date().getTime()` ja platvormipõhises `Calendar.getInstance().getTimeInMillis()`. Hübriidi korral toimus kaardi laadimine samaaegselt vaate laadimisega ehk keskmiselt kulus 2ms, Androidi puhul aga võttis see natuke kauem – peale vaate laadimist tuli veel oodata keskmiselt 48ms. Logi on saadaval Lisas 2.

Erinevust oli visuaalselt raske märgata. Google Maps'i kasutamine on sujuv nii hübriid- kui ka platvormipõhises rakenduses. Mõlemal on sama funktsionaalsus ja kasutamisel erinevusi ei tekkinud.

5.5 Tulemus

Testides ja kasutades autor suuri erinevusi hübriid- ja platvormipõhise rakenduste vahel ei leidnud. Kõige suurem puudus hübriidrakendusel oli käivitamiseks kulunud aeg. Kui rakenduses on ette nähtud kasutajale avamisel rakenduse logo kuvamine paariks sekundiks, siis ei tekita see probleemi. Vaadete vahel liikumises ja Google Maps'i kuvamises oli hübriidrakenduses kiirem.

6 Kokkuvõte

Bakalaurusetöö teema sai valitud vajadusest teha rakendus lehele Petify.ee. Antud lõputöö peamiseks eesmärgiks oli saada valmis töötav ja Android'i ning iOS poodi üleslaadimiseks valmis olev hübriid-mobiilirakendus, mis sai ka täidetud. Rakendus laetakse poodi üles nimega Petify juuni alguseks.

Töö käigus uuriti juba valmisolevaid rakendusi ja võrreldi valminud rakendusega. Kuna tegemist on reaalse projektiga, siis võttis autor eriti tõsiselt kasutajamugavust ja turvalisust.

Lisaks sarnaste rakenduste võrdlemisele kirjeldas autor kasutatud tehnoloogiaid ja põhjendas oma valikuid. Töö teises pooles kirjeldas töö tegija valminud rakendust ning võrdles omavahel hübriid- ja platvormipõhist rakendust Petify näitel. Tulemusena leidis autor, et mõlemal on omad eelised ja puudused. Kindlasti on telefoni funktsionaalsust lihtsam platvormipõhisel rakendusel kasutada, kuid paljud erinevad ja hästi dokumenteeritud pistikprogrammid aitavad hübriid-mobiilirakendusel võistelda platvormipõhiste rakendustega.

Valminud rakendus on alles esimene versioon ja seda saab kindlasti edasi arendada. Funktsionaalsusest saaks veel lisada GPS toe. Autor tegeleb selle projektiga ka edaspidiselt.

Kasutatud kirjandus

- [1] „Rover,“ [Võrgumaterjal]. Available: <https://www.rover.com/>. [Kasutatud 12 mai 2018].
- [2] „Fiverr,“ [Võrgumaterjal]. Available: <https://www.fiverr.com/>. [Kasutatud 12 mai 2018].
- [3] „Node.js,“ [Võrgumaterjal]. Available: <https://nodejs.org/en/>. [Kasutatud 18 aprill 2018].
- [4] „Wikipedia - Node.js,“ [Võrgumaterjal]. Available: <https://en.wikipedia.org/wiki/Node.js>. [Kasutatud 18 aprill 2018].
- [5] „Npmjs,“ [Võrgumaterjal]. Available: <https://docs.npmjs.com/>. [Kasutatud 18 aprill 2018].
- [6] „Apache Cordova,“ [Võrgumaterjal]. Available: <https://cordova.apache.org/docs/en/latest/>. [Kasutatud 19 aprill 2018].
- [7] „Ionic,“ [Võrgumaterjal]. Available: <https://ionicframework.com/docs/>. [Kasutatud 19 aprill 2018].
- [8] „Wikipedia - Ionic,“ [Võrgumaterjal]. Available: [https://en.wikipedia.org/wiki/Ionic_\(mobile_app_framework\)](https://en.wikipedia.org/wiki/Ionic_(mobile_app_framework)). [Kasutatud 19 aprill 2018].
- [9] „Angular,“ [Võrgumaterjal]. Available: <https://angular.io/docs>. [Kasutatud 19 aprill 2018].
- [10] „WP REST API,“ [Võrgumaterjal]. Available: <https://v2.wp-api.org/>. [Kasutatud 19 aprill 2018].
- [11] „WP REST API Handbook,“ [Võrgumaterjal]. Available: <https://developer.wordpress.org/rest-api/>. [Kasutatud 19 aprill 2018].
- [12] „JWT Authentication for WP REST API,“ [Võrgumaterjal]. Available: <https://wordpress.org/plugins/jwt-authentication-for-wp-rest-api/>. [Kasutatud 19 aprill 2018].
- [13] „JWT,“ [Võrgumaterjal]. Available: <https://jwt.io/>. [Kasutatud 19 aprill 2018].
- [14] „Firebase,“ [Võrgumaterjal]. Available: <https://firebase.google.com/docs/>. [Kasutatud 12 mai 2018].
- [15] „Gemnasium - Ionic,“ [Võrgumaterjal]. Available: <https://gemnasium.com/npms/ionic/versions>. [Kasutatud 13 mai 2018].
- [16] „Gemnasium - Cordova,“ [Võrgumaterjal]. Available: <https://gemnasium.com/npms/cordova/versions>. [Kasutatud 13 mai 2018].
- [17] „NgDevelop,“ [Võrgumaterjal]. Available: <https://www.ngdevelop.tech/angular/history/>. [Kasutatud 13 mai 2018].
- [18] „WordPress,“ [Võrgumaterjal]. Available: <https://wordpress.org/>. [Kasutatud 17 aprill 2018].

[19] „Wikipedia - Typescript,“ [Võrgumaterjal]. Available:
<https://en.wikipedia.org/wiki/TypeScript>. [Kasutatud 18 aprill 2018].

Lisa 1 – Vaadete vahel liikumise logi (hübriid vasakul)

Clicked button: 1526259593975	Clicked button: 1526261967768
Page loaded: 1526259593983	Page loaded: 1526261967839
Time taken in ms: 8	Time taken in ms: 71
Clicked button: 1526259595323	Clicked button: 1526261970151
Page loaded: 1526259595330	Page loaded: 1526261970220
Time taken in ms: 7	Time taken in ms: 69
Clicked button: 1526259596605	Clicked button: 1526261972284
Page loaded: 1526259596609	Page loaded: 1526261972340
Time taken in ms: 4	Time taken in ms: 56
Clicked button: 1526259597889	Clicked button: 1526261974397
Page loaded: 1526259597890	Page loaded: 1526261974450
Time taken in ms: 1	Time taken in ms: 53
Clicked button: 1526259598964	Clicked button: 1526261976297
Page loaded: 1526259598970	Page loaded: 1526261976352
Time taken in ms: 6	Time taken in ms: 55
Clicked button: 1526259600392	Clicked button: 1526261979273
Page loaded: 1526259600403	Page loaded: 1526261979330
Time taken in ms: 11	Time taken in ms: 57
Clicked button: 1526259601866	Clicked button: 1526261981907
Page loaded: 1526259601868	Page loaded: 1526261981967
Time taken in ms: 2	Time taken in ms: 60
Clicked button: 1526259605375	Clicked button: 1526261983776
Page loaded: 1526259605382	Page loaded: 1526261983829
Time taken in ms: 7	Time taken in ms: 53
Clicked button: 1526259606638	Clicked button: 1526261985935
Page loaded: 1526259606645	Page loaded: 1526261985985
Time taken in ms: 7	Time taken in ms: 50
Clicked button: 1526259607842	Clicked button: 1526261987993
Page loaded: 1526259607849	Page loaded: 1526261988048
Time taken in ms: 7	Time taken in ms: 55

Lisa 2 – Google Maps'i laadimine (hübriid vasakul)

Button clicked: 1526265071386	Clicked button: 1526264412014
View loaded: 1526265071405	View loaded: 1526264412650
Map loaded: 1526265071405	Map loaded: 1526264412710
Time taken in ms: 0	Time taken in ms: 60
Button clicked: 1526265073347	Clicked button: 1526264415803
View loaded: 1526265073353	View loaded: 1526264415935
Map loaded: 1526265073356	Map loaded: 1526264415990
Time taken in ms: 3	Time taken in ms: 55
Button clicked: 1526265074628	Clicked button: 1526264418381
View loaded: 1526265074634	View loaded: 1526264418511
Map loaded: 1526265074635	Map loaded: 1526264418559
Time taken in ms: 1	Time taken in ms: 48
Button clicked: 1526265076148	Clicked button: 1526264420332
View loaded: 1526265076153	View loaded: 1526264420454
Map loaded: 1526265076155	Map loaded: 1526264420511
Time taken in ms: 2	Time taken in ms: 57
Button clicked: 1526265078026	Clicked button: 1526264422190
View loaded: 1526265078033	View loaded: 1526264422308
Map loaded: 1526265078035	Map loaded: 1526264422356
Time taken in ms: 2	Time taken in ms: 48
Button clicked: 1526265079403	Clicked button: 1526264424190
View loaded: 1526265079406	View loaded: 1526264424298
Map loaded: 1526265079407	Map loaded: 1526264424343
Time taken in ms: 1	Time taken in ms: 45
Button clicked: 1526265080728	Clicked button: 1526264426334
View loaded: 1526265080735	View loaded: 1526264426439
Map loaded: 1526265080737	Map loaded: 1526264426478
Time taken in ms: 2	Time taken in ms: 39
Button clicked: 1526265082446	Clicked button: 1526264429161
View loaded: 1526265082451	View loaded: 1526264429266
Map loaded: 1526265082454	Map loaded: 1526264429306
Time taken in ms: 3	Time taken in ms: 40
Button clicked: 1526265084047	Clicked button: 1526264431042
View loaded: 1526265084056	View loaded: 1526264431154
Map loaded: 1526265084061	Map loaded: 1526264431203
Time taken in ms: 5	Time taken in ms: 49
Button clicked: 1526265092593	Clicked button: 1526264438712
View loaded: 1526265092595	View loaded: 1526264438832
Map loaded: 1526265092596	Map loaded: 1526264438870
Time taken in ms: 1	Time taken in ms: 38