

TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND
Arvutiteaduse instituut
Infotehnoloogia eriala

Gerrit Kraav
Mobiilse haiglainfosüsteemi broneeringu
rakenduse arendamine

Bakalaureusetöö (6 EAP)

Juhendaja: lektor Vambola Leping

Autor: " " mai 2012

Juhendaja: " " mai 2012

Lubada kaitsmisele

Professor: " " mai 2012

1 Sisukord

2	Sissejuhatus	3
3	Süsteemi üldine kirjeldus	4
3.1	Olemasolevad lahendused	4
3.2	Funktsionaalsed nõuded	4
3.2.1	Vastuvõtuoja otsimine arsti järgi	4
3.2.2	Vastuvõtuoja otsimine eriala/teenuse järgi	4
3.2.3	Väljavalitud arsti/eriala vastuvõtuaeegade detailvaade.....	5
3.2.4	Sobiva vastuvõtuoja väljavalimine ja broneerimine.....	5
3.2.5	Arstile helistamine.....	5
3.2.6	Vastuvõtu hoone kuvamine kaardil	5
3.2.7	Seadete määramine.....	6
3.2.8	Enda broneeringute vaatamine	6
3.3	Mittefunktsionaalsed nõuded	6
3.3.1	Kasutatavus	6
3.3.2	Käideldavus.....	7
3.3.3	Jõudlus.....	7
3.3.4	Kasutajatugi.....	7
4	Arhitektuur	8
4.1	Android platvorm	8
4.2	Arenduskeskkond ja -vahendid	9
4.3	Disain	9
4.3.1	Pealeht.....	13
4.3.2	Otsimise vaade	13
4.3.3	Otsingu tulemused.....	14
4.3.4	Arsti vastuvõtuaeegade nimekiri.....	15
4.3.5	Vastuvõtuoja detailvaade ja broneerimine	16
4.3.6	Isikuandmete määramine.....	17
4.3.7	Minu broneeringud.....	18
4.3.8	Asukoha kaardi kuvamine	19
4.4	Rakenduse paigaldamine.....	20
	Kokkuvõte.....	21
	Mobile Reservation System Development	22
	Kirjandus.....	23
	Lisa.....	24

2 Sissejuhatus

Kuna tänapäeval on nutitelefonide arv järjest kasvamas ning paljud teenused kolivad mobiilsetesse seadmetesse, siis on eesmärk uurida ja arendada praktiline rakendus haiglale, mis võimaldaks otsida arstide vastuvõtuaegasid, ning vajadusel teha ambulatoorseid broneeringuid. Rakenduse platvormiks saab olema Android (1), mille arendus toimub *java* programmeerimiskeelt kasutades. Töö eesmärk on luua klient-server rakendus, kus kliendipoolne rakendus saadab serverile tavalisi *HTTP* (2) päringuid, misjärel server vastab *JSON* (3) formaati kasutades. Kuna paljudel juhtudel on andmeside piiratud resurss, siis eeltoodud lahendus on andmemahult efektiivne võrreldes näiteks veebiteenustega. Lisaks on plaanis pakkuda võimalust näidata haigla asukohta kaardil. Serverirakenduse esialgseks majutamiseks ning prototüüpimiseks on plaanis kasutada toodet Google App Engine (4), mis võimaldab tasuta jooksutada *java* põhiseid veebirakendusi. Lõpptulemusena peaks valmima rakendus, mis on tasuta kättesaadav kõigile, intuitiivse kasutajaliidesega, turvaline ning töötab kõigis Android operatsioonisüsteemiga varustatud seadmetes, olenemata ekraani suurusest ja eraldusvõimest. Lisaks valmib põhjalik dokumentatsioon antud rakenduse arhitektuurist ning kasutatavatest kolmandate osapoolte tekidest. Näiteks on mõistlik arenduse kiirendamiseks kasutada *JSON parserit* - Google Gson (5). Programmi esialgne arendus ja testimine toimub emulaatorit kasutades ning hilisemas etapis reaalsete seadmete peal.

3 Süsteemi üldine kirjeldus

3.1 Olemasolevad lahendused

Eesti kontekstis eraldi rakendust haiglale ei õnnestunud leida. Otsingu baasiks olid suuremad Eestis tegutsevad raviasutused - Sihtasutus Tartu Ülikooli Kliinikum, AS Ida-Tallinna Keskhaigla ja Sihtasutus Põhja-Eesti Regionaalhaigla. Samas olid olemas veebipõhised vastuvõtuoja otsimise komponendid, mida väiksema ekraaniga seadmetes on ebamugav kasutada. Maailmas seevastu olid vastavad rakendused nii Androidile kui ka iPhone'le.

3.2 Funktsionaalsed nõuded

3.2.1 Vastuvõtuoja otsimine arsti järgi

Eeltingimused	Kasutaja on otsimise kuval
Tulemus	Kasutaja näeb sisestatud nimele vastavate arstide nimekirja
Põhivoog	<ul style="list-style-type: none">• Kasutaja sisestab otsitava nime• Vajutatakse arsti järgi otsimise nuppu• Kuvatakse sisestatud nimele vastavate arstide nimekiri
Alternatiivsed vood	<ul style="list-style-type: none">• Kasutaja jätab otsingu välja tühjaks - kuvatakse informatiivne teade, et täpsustatakse otsingutingimusi• Puudub internetiühendus - kuvatakse informatiivne teade, et hetkel ei saa otsingut teostada, kuna puudub ühendus

3.2.2 Vastuvõtuoja otsimine eriala/teenuse järgi

Eeltingimused	Kasutaja on otsimise kuval
Tulemus	Kasutaja näeb sisestatud tekstile vastavate teenuste/erialade nimekirja
Põhivoog	<ul style="list-style-type: none">• Kasutaja sisestab otsitava eriala/teenuse• Vajutatakse eriala järgi otsimise nuppu• Kuvatakse sisestatud tekstile vastavate erialade/teenuste nimekiri
Alternatiivsed vood	<ul style="list-style-type: none">• Kasutaja jätab otsingu välja tühjaks - kuvatakse informatiivne teade, et täpsustatakse otsingutingimusi• Puudub internetiühendus - kuvatakse informatiivne teade, et

	hetkel ei saa otsingut teostada, kuna puudub ühendus
--	--

3.2.3 Väljavalitud arsti/eriala vastuvõtuaegade detailvaade

Eeltingimused	Kasutaja on otsingu tulemuste nimekirja kuval
Tulemus	Kasutaja näeb valitud arsti/eriala/teenuse detailvaadet
Põhivoog	<ul style="list-style-type: none"> • Kasutaja vajutab nimekirjas sobiva arsti peale • Kuvatakse antud arsti vastuvõtuaaja andmed
Alternatiivsed vood	<ul style="list-style-type: none"> • Puudub internetiühendus - kuvatakse informatiivne teade, et hetkel ei saa otsingut teostada, kuna puudub ühendus

3.2.4 Sobiva vastuvõtuaaja väljavalimine ja broneerimine

Eeltingimused	Kasutaja on arsti detailvaate kuval
Tulemus	Kasutajale kuvatakse detailne ülevaade vastuvõtuaajast ja -kohast ning võimalus antud aeg broneerida
Põhivoog	<ul style="list-style-type: none"> • Kasutaja vajutab nimekirjas sobiva päeva ja kellaaja peale • Kuvatakse antud päevale vastav lähim vaba vastuvõtuaeg koos arsti, asukoha ja kabineti andmetega
Alternatiivsed vood	<ul style="list-style-type: none"> • Puudub internetiühendus - kuvatakse informatiivne teade, et hetkel ei saa otsingut teostada, kuna puudub ühendus

3.2.5 Arstile helistamine

Eeltingimused	Kasutaja on arsti detailvaate kuval
Tulemus	Kasutaja helistab väljavalitud arstile
Põhivoog	<ul style="list-style-type: none"> • Kasutaja vajutab arsti detailvaate kuval helistamise nuppu • Rakendus helistab soovitud numbril
Alternatiivsed vood	<ul style="list-style-type: none"> • Arstil puudub telefoninumber - kuvatakse informatiivne teade

3.2.6 Vastuvõtu hoone kuvamine kaardil

Eeltingimused	Kasutaja on broneeringu andmete kuval
---------------	---------------------------------------

Tulemus	Kasutajale kuvatakse antud vastuvõtu hoone asukoht kaardil
Põhivoog	<ul style="list-style-type: none"> • Kasutaja vajutab "näita kaardil" nuppu • Rakendus kuvab kaardil vastava hoone asukoha
Alternatiivsed vood	<ul style="list-style-type: none"> • Kasutaja telefonis puudub kaardi tarkvara - kuvatakse informatiivne teade

3.2.7 Seadete määramine

Eeltingimused	Kasutaja on esilehel
Tulemus	Kasutaja isiku- ja kontaktandmed salvestatakse
Põhivoog	<ul style="list-style-type: none"> • Kasutaja vajutab seaded nuppu • Kuvatakse nime, isikukoodi, elukoha ja kontaktnumbri väljad • Kasutaja vajutab nuppu salvesta • Vajalikud andmed salvestatakse telefoni püsimällu
Alternatiivsed vood	<ul style="list-style-type: none"> • Puuduvad

3.2.8 Enda broneeringute vaatamine

Eeltingimused	Kasutaja on esilehel
Tulemus	Kasutajale kuvatakse nimekiri endaga seotud broneeringutest
Põhivoog	<ul style="list-style-type: none"> • Kasutaja vajutab broneeringud nuppu • Kuvatakse nimekiri tehtud broneeringutest
Alternatiivsed vood	<ul style="list-style-type: none"> • Tekib viga telefoni mälust faili lugemisega - kuvatakse informatiivne teade

3.3 Mittefunktsionaalsed nõuded

3.3.1 Kasutatavus

- Rakendus peab olema kergesti loetav erinevate ekraanisuuruste korral
- Rakendus peab olema lihtne ja loogiline
- Rakendus peab olema lihtsasti tõlgitav ja automaatselt valima õige keele vastavalt kasutaja seadme süsteemsetele parameetritele

- Kõik aega nõudvad protsessid, näiteks andmete laadimine serverist, peavad toimuma tagataustal ja mitte blokeerima kasutajaliidest, juhuks kui kasutaja otsustab otsingu katkestada
- Rakenduses esinevate ootamatute protsesside, näiteks sissetuleva kõne puhul, peab programm säilitama kasutaja hetke kuva ja andmed ning minema tagataustale

3.3.2 Käideldavus

- Rakendus peab andma informatiivseid veateateid võimalike vigade korral
- Rakendus ei tohi sisaldada kriitilisi vigasid

3.3.3 Jõudlus

- Kuna tegemist on reaajasüsteemiga ja kõik hetke andmed saadakse serverist siis määrab suure osa jõudlusest kasutaja interneti kiirus. Kuna saadetavad päringud on teksti kujul siis nende andmete töötlemine mobiilses seadmes peab olema kiire ning ei tohi võtta kauem kui 4 sekundit

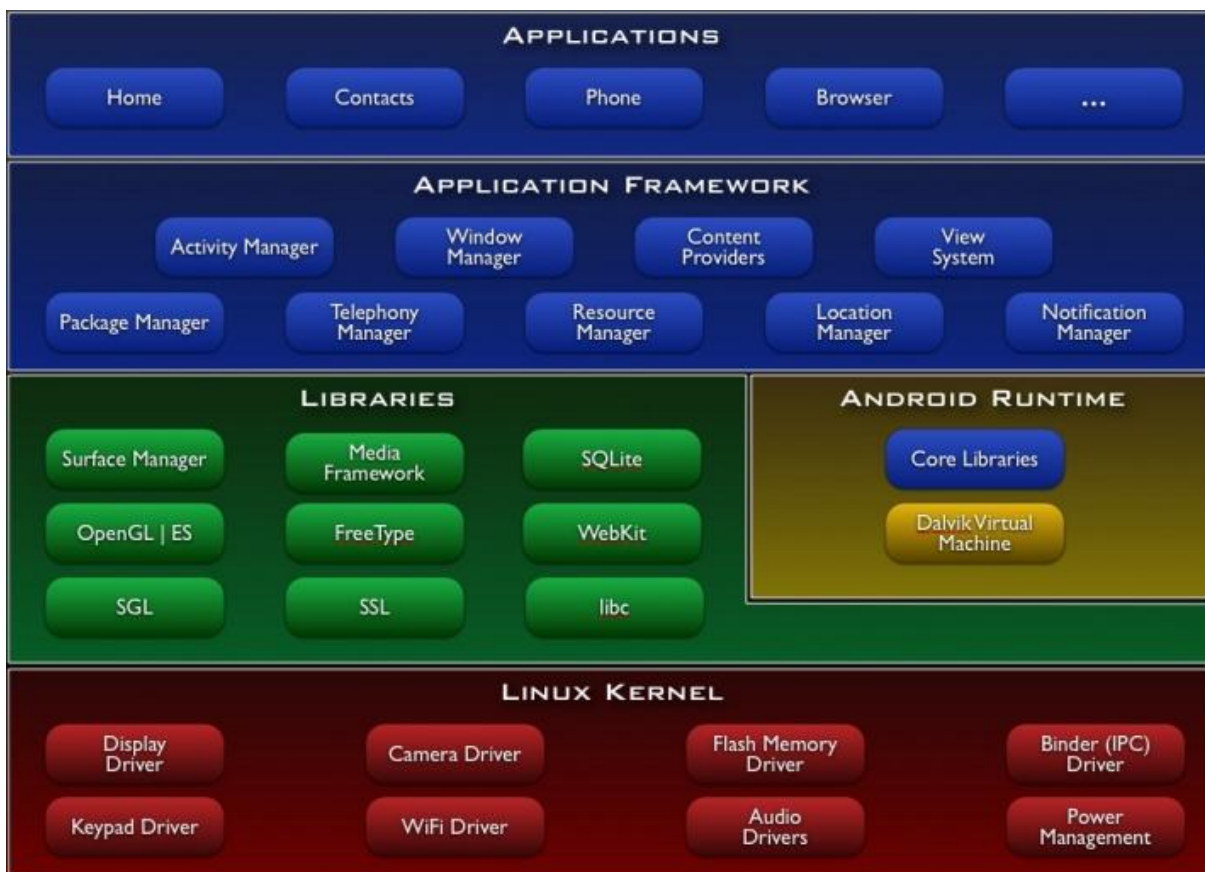
3.3.4 Kasutajatugi

- Rakendus peab olema dokumenteeritud ning manuaal lõppkasutajale kättesaadav
- Rakendus peab olema kättesaadav Google Play keskkonnast

4 Arhitektuur

4.1 Android platvorm

Android on tarkvarakogum erinevatele seadmetele, mis koosneb operatsioonisüsteemist, vahekihist, mille abil saab raudvara teenuseid kasutada, ning erinevatest rakendustest, nagu näiteks veebibrauser, telefoniraamat jne.



Joonis 1. Diagramm Androidi komponentidest <http://developer.android.com/guide/basics/what-is-android.html>

Joonisel 1 toodud diagrammil on näha Androidi jagunemine erinevate ülesannetega kihtideks

- Rakendused - koosneb erinevatest kasutaja jaoks olulistest programmidest, näiteks SMS-i saatmise rakendus, e-maili klient, telefoniraamat, mängud jne.
- Rakenduste vahekiht - võimaldab kasutada erinevaid raudvara teenuseid ja võimalusi, näiteks kasutada *GPS-i* et saada teada asukohta, kasutada *WiFi-t* võrguteenuste jaoks, uurida seadme tehnilisi andmeid ja failisüsteemis olevaid objekte. Sisaldab endas ka erinevaid taaskasutatavaid vaateid ja komponente erinevate programmide jaoks, näiteks nimekirjad, nupud, valikukastid, tabelid jne.
- Teegid - sisaldab endas C/C++ teeke, mida kasutavad erinevad programmid. Antud võimalusi saab kasutada eelnimetatud rakenduste vahekihi abil. Need raudvaraga

suhtlemise ja töötlemise teegid sisaldavad näiteks meedia teeke erinevate audio- ja pildioperatsioonide jaoks. Relatsioonilise andmebaasi teeki jne.

- Android *runtime* - pakub samu võimalusi, mis java põhiteegid, mõningate eranditega, mis ei ole mobiilsetele seadmetele vajalikud.
- Linux Kernel – Android baseerub Linuxi versioonil 2.6 ning kasutab ära selle võimalusi mälu ja protsesside ja muu haldamiseks.

4.2 Arenduskeskkond ja -vahendid

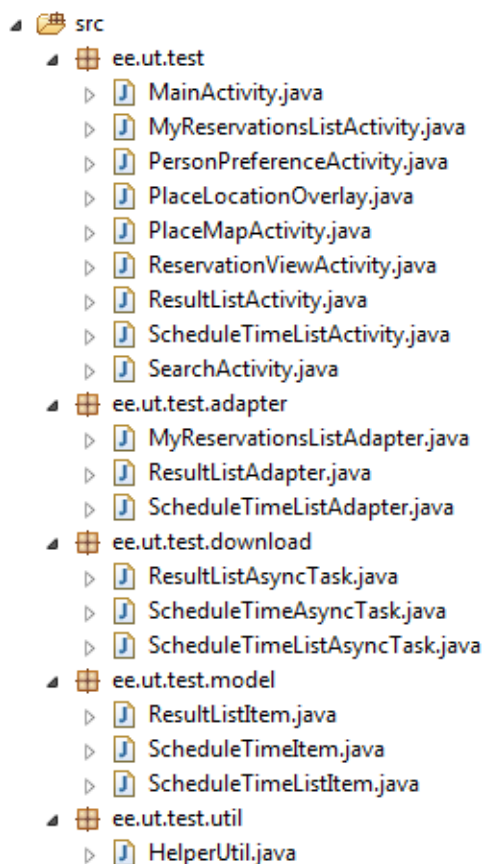
Androidi arendus toimub *java* programmeerimiskeelt kasutades ning graafilise kasutajaliidesena kasutan Eclipse (6) keskkonda, millele on loodud mitmeid abistavaid tööriistu arenduse kiirendamiseks. Lisaks sellele on vajalik *JDK* (7) olemasolu ja installitud peab olema ka Android SDK (8), mis võimaldab alla laadida ja hallata erinevaid Androidi versioone ning seejärel arendada mingit kindlat versiooni arvesse võttes. See on vajalik selleks, et verifitseerida ja olla kindel, et antud rakendus jookseb ka mõne vanema seadme peal. Antud projekti käigus kasutatakse 2.1 platvormi, kuna hetkel on piisavalt telefone, mis seda kasutavad ning lisaks ei ole oluline implementeerida kõige uuemaid tehnoloogiaid, mis on saadaval näiteks 4.0 platvormil.

Arendus hakkab toimuma vastavat emulaatorit kasutades, mis võimaldab kasutada kõiki funktsioone, mis telefon – võimalus simuleerida sissetulevat kõnet, lülitada seade välja, muuta *GPS* koordinaate ja saata vastav teade rakendusele jne. Seega toimub rakenduse lõplik testimine füüsilise seadme peal alles viimases faasis, kus uut funktsionaalsust enam ei lisandu. Probleemide tekkimisel on lisaks võimalus seade otse ühendada arenduskeskkonnaga ja siluda seeläbi eettulevaid vigu.

4.3 Disain

Tegemist on klient – server rakendusega, kus mobiilpoolne rakendus saadab *HTTP* päringuid mingile aadressile, mis vastab *JSON* formaati kasutades. Antud projekti raames on serveri majutamiseks kasutatud Google Appengine'it arenduse kiirendamiseks. Hiljem on antud lahendust lihtne üle viia mingile reaalsele andmebaasile ning suunata vajalikud päringud ümber korrektsetele aadressidele. Näiteks Ida-Tallinna Keskhaigla lehel otsides neuroloogi vastuvõtuaegasid, on antud teenus saadaval vastavalt *URL-ilt* http://www.itk.ee/vastuvotuajad/_service/Neuroloog, sarnast *REST* (9) stiilis lahendust hakkab kasutama ka antud rakenduse implementatsioon

Kogu projekti lähtekood on jagatud eraldi pakettidesse, et anda parem ülevaade ja parandada koodi hilisemat modifitseerimist. Joonisel 2 on toodud väljavõte projekti lähtekoodist

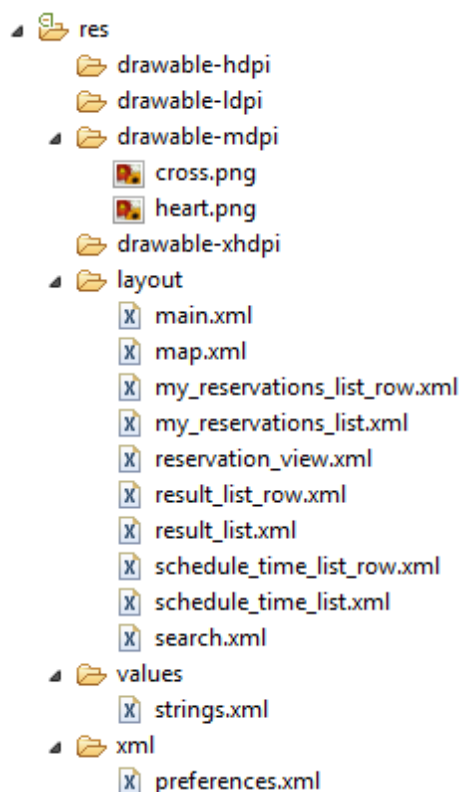


Joonis 2 Lähtekoodi struktuur

- Põhipakett on *ee.ut.test*, mille alla kuuluvad kõik rakenduse tegevused, ehk mingid ekraanivaated, kuhu saab liikuda ja teatud tegevusi sooritada.
- Paketti *ee.ut.test.adapter* on koondatud kõik failid, mis otsustavad ja määravad kuidas etteantud ridasid kuvada mingis loetelus.
- Kolmandas alajaotuses *ee.ut.test.download* asuvad failid, mis kannavad hoolt selle eest, kuidas mingilt aadressilt vastavad tulemused kätte saada. Kõik antud paketti kuuluvad klassid laiendavad Androidi abstraktset klassi *android.os.AsyncTask*, mis võimaldab allalaadimist teostada eraldi protsessis, ilma et põhiprotsess, kus kasutaja asub, oleks blokeeritud (11)
- *ee.ut.test.model* alajaotuses on kõik failid, mida kasutada mingi tervikliku andmekogumi hoidmiseks ning näiteks kuvada seda kuskil loetelus. Antud failid genereeritakse automaatselt vastavalt saadud päringu tulemusele Gson teeki kasutades
- Viimases pakettis *ee.ut.test.util* asub abiklass, mis võimaldab mitmes kohas ära kasutada samu vajadusi, näiteks broneeringu salvestamiseks telefoni püsimällu või siis seal lugemiseks. Samuti on seal meetod, mis võimaldab kindlaks teha kas hetkel on

interneti ühendus olemas või mitte ja läbi selle anda kasutajale infot järgnevate protsesside kohta.

Eelmises loetelus toodud failid olid kõik *java* laiendiga, kuid järgmises olulises kaustas asuvad antud projekti kõik kujundusega seotud failid, väljavõtte lahtisest struktuurist asub joonisel 3.



Joonis 3 Ressursside kaust

- *drawable* prefiksiga kaustades asuvad kõik rakenduses kasutatavad pildifailid, vastavalt sellele, millisele seadmele antud failid on mõeldud. Näiteks antud projekti raames on loodud ainult üks eksemplar kumbagi ikooni, ning pandud need *drawable-mdpi* kausta, mis tähendab et need ikoonid on mõeldud seadmele, mille *Screen pixel density (dpi)* (10) on umbes 160dpi-d. Samas ei tähenda see, et suuremate või väiksemate ekraanide ja eraldusvõimetega seadmed ei suudaks seda kuvada.
- *layout* kaustas asuvad kõik projekti vaated, mis määravad kujunduse mingi konkreetse tegevuse ajal. *_row* sufiksiga failid määravad kuidas mingis loetelus antud rida kuvada, näiteks arstide loetelus peaks olema arsti nimi ja eriala ühel real ja vastuvõtu ajad teisel.
- *values* kaustas asub *string.xml* fail, mis kirjeldab kõik antud rakenduses kasutatavad tõlked kujul `<string name="võti">väärtus</string>`. Antud projekti raames on tõlkimisel piiratud ainult eesti keelega, kui vajadusel saab kergesti lisada muid keeli.

- *xml* kaustas asub *preferences.xml* fail, mida kasutab seadete määramise kuva

Üks oluline komponent lisaks on *AndroidManifest.xml* fail, kus tuleb üles märkida kõik rakenduses kasutatavad komponendid ja tegevused. Lisaks antud rakenduse hetke versioon, mis on vajalik selleks, kui programm üles laadida Google Play (12) keskkonda, siis uue versiooni tekkimisel oleks võimalik koheselt kasutajat teatada uuendustest.

Samuti saab seal kirjeldada minimaalse platvormi versiooni, millel antud rakendus jookseb. See aitab vältida olukorda, kus kasutaja laeb alla programmi, mis kasutab uuema platvormi võimalusi, mistõttu ei tööta rakendus õigesti.

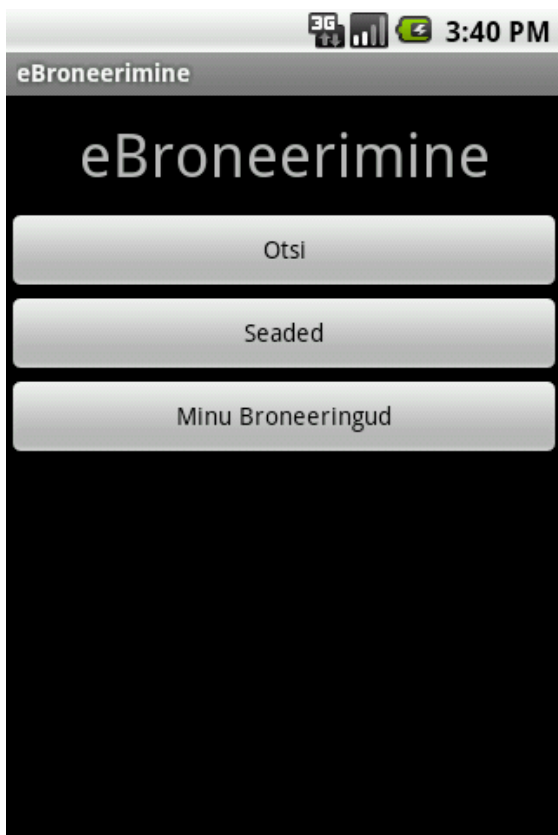
Lisaks on vaja kirjeldada antud failis kõik õigused, mida antud rakendus kasutab. See on vajalik selleks, et kasutajal oleks ülevaade, mida programm teha tohib ja olla kindel, et ei kasutataks ebavajalikke andmeid. Antud rakenduses on vajalik kolme õiguse olemasolu

```
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Esimene õigus *CALL_PHONE* on vajalik arstile helistamise funktsionaalsuse tagamiseks. Teine *INTERNET* õigus lubab rakendusel kasutada võrku andmete laadimiseks ning kolmas *ACCESS_NETWORK_STATE* õigus võimaldab kindlaks teha kas hetkel on võimalik kasutada internetti andmete vaatamiseks

4.3.1 Pealeht

Antud vaade joonisel 4 kasutab suuri nuppe kasutusmugavuse tagamiseks ning pakub kiiret võimalust enam kasutatavate funktsioonide kasutamiseks



Joonis 4 Pealeht

4.3.2 Otsimise vaade

Joonisel 5 toodud kuva võimaldab kasutajal sisestada sõna ning otsida arsti või teenuse ja eriala järgi. Hetkel tagastatakse serverist staatilised testandmed, kuid päring saadetakse kujul <http://android-ee.appspot.com/search?searchString=...&searchType=...>

kus *searchString* on kasutaja sisestatud sõna ning *searchType* on kas *doctor* või *profession* vastavalt kasutaja tehtud valikule



Joonis 5 otsimise vaade

4.3.3 Otsingu tulemused

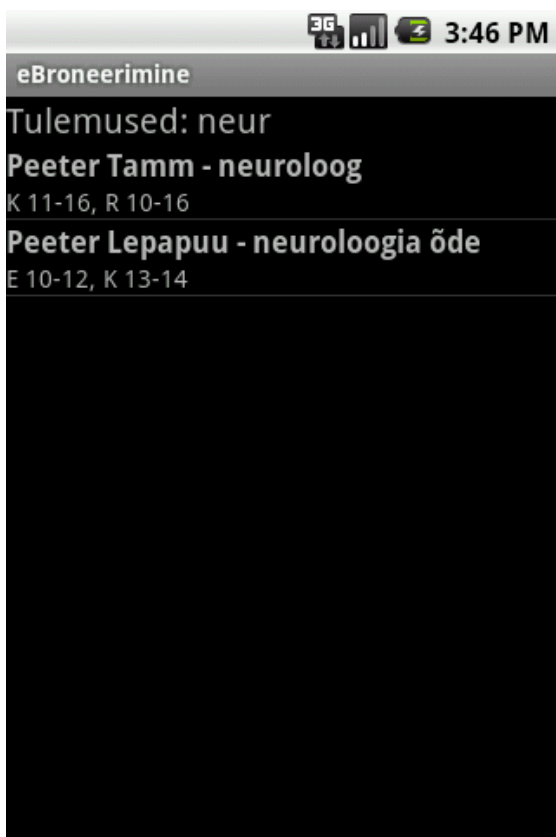
Joonisel 6 toodud loetelus kuvatakse vastavalt kasutaja sõnale vastavad read, kus rea esimene tekst näitab arsti nime ja eriala ning teine vastvõtu päevi. Hetkel eeldatakse, et server on suuteline tagastama formaaditud kujul vastuvõtuaegasid, kuna mobiilne seade on tavaliselt väiksema andmetöötlemisjõudlusega, siis on targem see lasta serveril koostada.

Antud nimekiri pannakse kokku *JSON* nimekirjast, kujul

```
[{
  "staffId": 1,
  "staffName": "Peeter Tamm",
  "telephone": "3556234",
  "profession": "neuroloog",
  "scheduleDays": "K 11-16, R 10-16"
}, {
  "staffId": 2,
  "staffName": "Peeter Lepapuu",
  "telephone": "12312332",
  "profession": "neuroloogia õde",
  "scheduleDays": "E 10-12, K 13-14"
}, {
  ...
}]
```

Valitud rea valimisel tehakse päring aadressile

<http://android-ee.appspot.com/scheduletimes?staffId=...>, kus *staffId* on vastava rea arsti identifikaator



Joonis 6 Otsingu tulemused

4.3.4 Arsti vastuvõtuaegade nimekiri

Joonisel 7 toodud vaates on võimalik kasutajal valida soovitud aeg vastuvõtuks välja, lisaks on soovi korral olemas võimalus arstile helistada ja küsida täpsustavaid küsimusi. Hetkel eeldatakse et võimalikud vastuvõtuajad tulevad ajaliselt kasvavas järjekorra, see tähendab et sorteerimine tuleb sooritada serveris

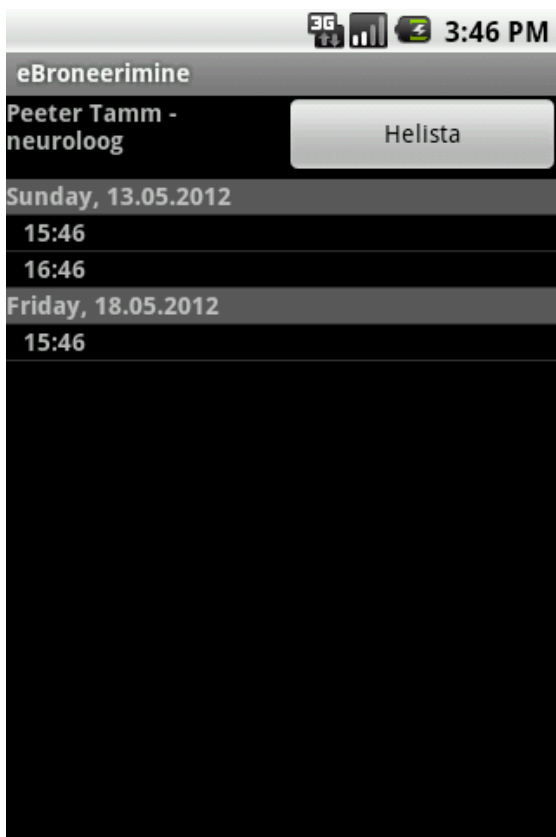
Nimekiri koostatakse *JSON* nimekirjast, kujul

```
[{
  "id": 10,
  "date": "May 13, 2012 4:35:04 PM"
}, {
  "id": 11,
  "date": "May 13, 2012 5:35:04 PM"
}, {
  "id": 12,
  "date": "May 18, 2012 4:35:04 PM"
}]
```

Kus *id* on vastava vastuvõtu aja identifikaator, ja *date* vastav vastuvõtuaja kuupäev ja kellaeg

Valitud rea väljavalimisel tehakse päring aadressile

<http://android-ee.appspot.com/scheduletime?scheduleTimeId=...>, kus `scheduleTimeId` on väljavalitud rea identifikaator



Joonis 7 vastuvõtuaegade nimekiri

4.3.5 Vastuvõtuaja detailvaade ja broneerimine

Joonisel 8 toodud kuval on võimalik kokkuvõtvalt vaadata broneeringu andmeid ning lisaks kuvada antud vastuvõtuaja asukoha maja kaardil ning soovi korral broneering ära vormistada. Kui broneeringu tegemisel on kasutajal isikuandmed määramata, kuvatakse dialoog, mis võimaldab kasutajal minna isikuandmeid määrama. Tagasi tulles nüüd broneeringu kuvale on võimalik antud reserveering teha. Pärast antud operatsiooni tegemist salvestatakse broneeringu andmed ka telefoni mällu teksti kujul, et hiljem oleks mugav enda broneeringuid vaadata ja ei peaks kasutama internetti ning pärima vastavaid andmeid. Hetkel realselt broneeringut kuskile ei salvestata, kuna see ei ole antud projekti skoobis ning serveri pool on ainult testvastuste genereerimiseks.

Vaade pannakse kokku *JSON* kirjest

```
{
  "id": 10,
  "date": "May 13, 2012 4:50:00 PM",
  "staffname": "Peeter Tamm",
  "staffTelephone": "21313231",
```



```
"place": "Puusepa 8",  
"cabinet": "A123",  
"placeLat": 58.369351,  
"placeLon": 26.70055  
}
```



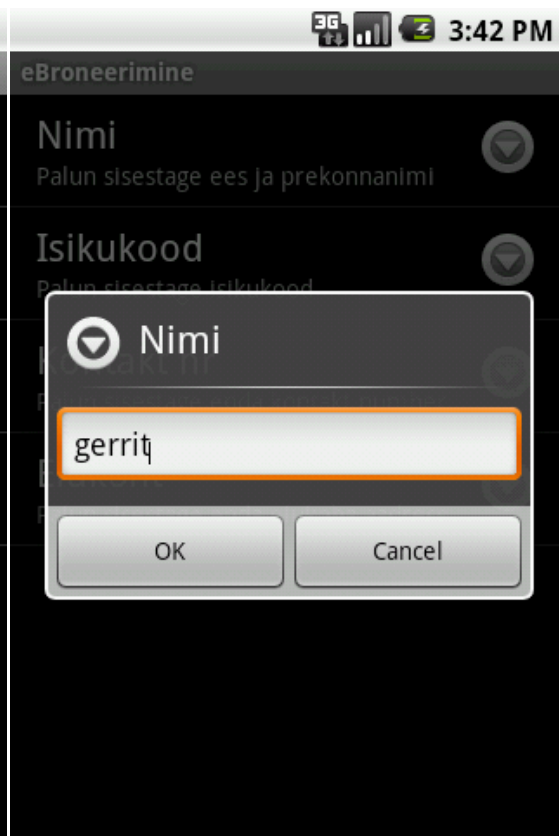
Joonis 8 vastuvõtujale broneerimine

4.3.6 Isikuandmete määramine

Joonisel 9 ja 10 on näha kuva, kus kasutaja saab sisestada enda andmeid, mis on vajalikud broneeringu tegemiseks. Andmed salvestuvad kohe peale vastavas dialoogis OK nupu vajutamist.



Joonis 9 Isikuandmete määramine



Joonis 10 Nime sisestamine

4.3.7 Minu broneeringud

Joonisel 11 on kasutajal võimalus vaadata enda tehtud broneeringuid, mis on telefoni mällu salvestatud ning vastava rea väljavalides (joonis 12) saab täpsustada tegevust, mida antud broneeringu korral teha. Võimalik on helistada arstile, kustutada antud rida või vaadata asukoha kaarti.



Joonis 11 Minu broneeringud



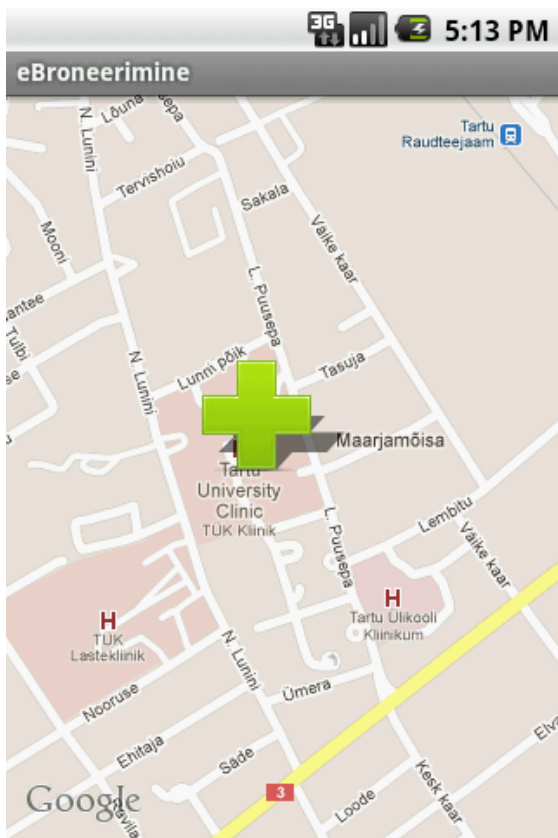
Joonis 12 Minu broneeringud tegevus

4.3.8 Asukoha kaardi kuvamine

Joonisel 13 on võimalik vaadata antud haigla asukoha kaarti

Selleks, et oleks võimalik Google kaarti kasutada peab telefonis olema vastav teek olemas ning manifesti failis tuleb määrata, et kasutataks vastavat võimalust.

```
<uses-library android:name="com.google.android.maps" />
```



Joonis 13 Asukoha kaart

4.4 Rakenduse paigaldamine

Kogu rakendus on kokkupakitud failiks eBroneerimine.apk. Kuna hetkel ei ole see Android Play keskkonda üles laetud, siis tuleb telefoni paigaldamiseks kasutada Android SDK võimalusi. Selleks tuleb telefon ühendada arvutiga USB-d kasutades ning käivitada käsk

adb install eBroneerimine.apk

Adb (Android Debug Bridge) on käsurea utiliit, mis võimaldab suhelda kas emulaatori või reaalse seadmega ning selle leiab <sdk>/platform-tools/ alt, kus <sdk> on kasutaja Androidi arendusvahendite paigaldamise kaust.

Kokkuvõte

Käesolevas bakalaureusetöös arendati mobiilne broneerimissüsteem haiglale Android platvormi jaoks. Hetkel ei tööta see rakendus küll päris andmebaasi peal, kuid vajadusel ning mõne haigla või polikliiniku huvi korral on võimalik rakendus seadistada töötama uue andmebaasi peal. Käesoleval hetkel (mai 2012) saab rakendus andmed test serverilt, mis asub aadressil <http://android-ee.appspot.com>. Lisaks hõlmas antud töö ka selle serveri püstipanekut, et simuleerida reaalselt andmevahetust üle võrgu, mitte testandmetega rakenduses endas.

Kuna mobiilsed seadmed on järjest rohkem kasutamist leidvad erinevates valdkondades nagu pangandus ja m-parkimine, siis võiks olla tulevikutrend ka haiglasse ambulatoorse broneeringu tegemise võimaldamine läbi mobiilse seadme. Antud töös ei jäi küll mõni funktsionaalsus implementeerimata, näiteks mobiili-id kasutamine, mis isikuandmete kvaliteedi tagamiseks peaks olema olemas, kuid põhiline protsess, mis spetsifikatsioonis sai üles seatud, sai tehtud. Peamine teadmiste allikas antud töös oli Androidi arendajatele mõeldud dokumentatsioon (13) ja raamat Professional Android Application Development (14) ning mõne probleemi ilmnedes ka erinevad foorumid.

Mobile Reservation System Development

Bachelor thesis (6 EAP)

Gerrit Kraav

Summary

The purpose of this bachelor thesis was to develop a mobile reservation system for Android platform. At the moment this program does not work on a real database server. But in case of interest by some hospital it could be ported to use some real data server. At the moment (May 2012) each query is sent to <http://android-ee.appspot.com> . Getting up this server was also one necessity in order to simulate real time data transfer over the internet and not use static data included in mobile application

As the mobile industry is in a rapid development and m-banking and m-parking are very familiar for everybody then the future for making an ambulatory reservation in the hospital could also be made more simple. This work misses some functionality, for example using mobile-id for better personal data but the main process that was required, has been implemented.

Kirjandus

- 1.* Allikas: Android, the world's most popular mobile platform:
<http://www.android.com/about/>
- 2.* Allikas: Hypertext Transfer Protocol:
http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- 3.* Allikas: Introducing JSON: <http://www.json.org/>
- 4.* Allikas: Power Your Business with Google Cloud:
<https://developers.google.com/appengine/>
- 5.* Allikas: A Java library to convert JSON to Java objects and vice-versa:
<https://code.google.com/p/google-gson/>
- 6.* Allikas: Eclipse Classic: <http://www.eclipse.org/downloads/moreinfo/classic.php>
- 7.* Allikas: Java Development Kit: http://en.wikipedia.org/wiki/Java_Development_Kit
- 8.* Allikas: Download the Android SDK: <http://developer.android.com/sdk/index.html>
- 9.* Allikas: Principled Design of the Modern Web Architecture:
<http://www.ics.uci.edu/~taylor/documents/2002-REST-TOIT.pdf>
- 10.* Allikas: Pixel density: http://en.wikipedia.org/wiki/Pixel_density
- 11.* Allikas: Android Threads, Handlers and AsyncTask - Tutorial:
<http://www.vogella.com/articles/AndroidPerformance/article.html>
- 12.* Allikas: Introducing Google Play: <https://play.google.com/store>
- 13.* Allikas: Android: <http://developer.android.com/index.html>
- 14.* Allikas: Professional Android Application Development:
<http://www.wrox.com/WileyCDA/WroxTitle/Professional-Android-Application-Development.productCd-0470344717.html>

Lisa

Antud rakenduse lähtekood on saadaval tööga kaasas oleval CD-l