

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Janer Gorohhov

**Platvormist sõltumatu mobiilirakendus
lõunatajatele**

Bakalaureusetöö (6 EAP)

Juhendaja: Pelle Jakovits

Tartu 2016

Platvormist sõltumatu mobiilirakendus lõunatajatele

Lühikokkuvõte:

Käesoleva bakalaureusetöö eesmärgiks on luua platvormist sõltumatu mobiilirakendus lõunatajatele. Töös kirjeldatakse ära uute tutvuste leidmise probleem ning selgitatakse lahendusena arendatud mobiilirakendust. Bakalaureusetöös võrreldakse olemasolevaid lahendusi ning kirjeldatakse nende puudusi. Lahendus arendatakse valmis prototüübina, kus nõuded on seatud võimalikult realistlikult vastavalt lõppkasutaja ootustele. Antud töös kirjeldatakse rakenduse käitumismudelid ning esitatakse näited piltidena.

Võtmesõnad:

Mobiilirakendus, Apache Cordova, AngularJS, Ionic, Sotsiaalne võrgustik, ASP.NET, C#, Platvormist sõltumatu, Prototüüp

CERCS: P175

Platform Independent Mobile Application for Lunchers

Abstract:

The purpose of this bachelor's thesis is to create a platform independent mobile application for lunchers. Present thesis describes the issues with finding new acquaintances and provides the mobile application as a solution. Current thesis analyzes existing solutions and points out their pros and cons. The solution is developed as a prototype in which the requirements are presented as realistic as possible from the view of the end user. The thesis includes use cases and snapshots of the application to simplify the understanding.

Keywords:

Mobile application, Apache Cordova, AngularJS, Ionic, Social network, ASP.NET, C#, Platform independent, Prototype

CERCS: P175

Sisukord

1.	Sissejuhatus	5
2.	Olemasolevad lahendused	7
2.1	Olemasolevad lahendused Eestis.....	7
2.2	Olemasolevad lahendused mujal maailmas.....	7
2.2.1	NeverEatAlone.....	7
2.2.2	LunchBuddy.....	8
2.2.3	LunchMatcher	8
3.	Funktsionaalsuse nõuded	9
3.1	Sisselogimine.....	9
3.2	Välja logimine	10
3.3	Lõunasoovi lisamine.....	10
3.4	Lõunakutse saatmine	11
3.5	Lõunakutse vastuvõtmine	12
3.6	Kasutaja profiili muutmine.....	12
4.	Rakenduse arhitektuur.....	14
4.1	Kasutatud tehnoloogiad	14
4.1.1	Apache Cordova	14
4.1.1.1	Arhitektuur	14
4.1.2	Ionic	15
4.1.3	JavaScript, AngularJS	15
4.1.4	ASP.NET MVC, C#.....	16
4.1.4.1	Arhitektuur	16
4.1.5	Microsoft SQL	18
4.1.6	Microsoft Azure	18
4.1.7	CSS3	19
4.1.8	HTML5	19
4.2	Kontrollerid	19
4.2.1	Veebiteenuse kontrollerid	19
4.2.2	Mobiilirakenduse kontrollerid	21
4.3	Andmemudel	22
5.	Rakenduse kirjeldus	24
5.1	Päis, jalus ja menüü	24
5.2	Maandumisleht	26
5.3	Profiili kuva.....	27

5.4	Lõuna postitamine	28
5.5	Aktiivsed kutsed ning lõunate ajalugu	29
5.6	Isikustamine.....	31
5.7	Minu profiil	33
5.8	Rakenduse kirjeldus.....	33
5.9	Väljalogimine	34
6.	Kokkuvõte	35
7.	Kasutatud materjalid	37
Lisad	39
I.	Programmi lähtekood	39
II.	Litsents	40

1. Sissejuhatus

Tänapäeval on sotsiaalse suhtlusvõrgustiku laiendamine piiratud eelkõige ajaliselt, aga ka oskuslikult. Samas teame, et tänapäeva maailmas edu üheks võtmeks on inimsuhted ning nende oskuslik kasutamine õigel ajahetkel. USAs läbiviidud uuringus tõestati, et neli inimest kümnest leiavad hea töökohta just läbi sotsiaalse võrgustiku [1]. Kui lisada uuringule juurde veel läbi sotsiaalvõrgustiku saadud tööpakkumised, siis näeme, et sotsiaalse võrgustiku ja suhete osatähtsus inimeste elus on suur.

Antud bakalaureusetöö põhieesmärk on arendada valmis mobiilse rakenduse prototüüp, mis võimaldab inimestel valida oma eelistuste põhjal endale lõunakaaslane ning laiendada seeläbi oma sotsiaalset võrgustikku. Kuna lõunasöök on inimeste igapäevane osa, siis on võimalik seda aega kulutada otstarbekamalt, ühendades lõunatamise aja tutvudes mõne uue inimesega. Kui leppida lõunasöök kokku mugavalt ja kiirelt, ei kaota inimene oma olulist aega, kuid saab arendada ennast sotsialiseerudes inimesega väljastpoolt oma tavapärasest tutvusringkonda. Lisaks enesearendamisele saab pikaajalise kasutamise tagajärjel kasutaja suurendada oma tutvusringkonda selliste oskuste ja huvidega tutvustega, kelle järele inimene sotsiaalset või praktilist vajadust tunneb. Kõige olulisem selle juures on see, et inimene ei kaota oma päevarutiinist olulist aega.

Sekundaarseks eesmärgiks on arendada prototüüp valmis platvormist sõltumatu lahendusena, mida on võimalik laiendada tulevikus lisaks prototüübi alusplatvormile (Android) ka ilma *native* rakendust tegemata teistele mobiilplatvormidele (IOS, Windows).

Rakendus kasutab kasutajate tuvastamiseks Facebooki veebiteenust, läbi mille süsteem talletab kasutaja profiiliinfo, mida mobiilirakendus kasutab kasutaja isikustamiseks. Rakendus võimaldab kasutajatel:

- kuvada inimeste lõunaavaldusi
- postitada kuvamiseks kasutajatele lõunasooviavalduse
- kuvada lõunate ajalugu
- kuvada, aksepteerida ning tühistada kasutajale saadetud lõunakutseid
- kuvada kasutajate profiiliinfot

Töö esimeses peatükis analüüsitakse olemasolevaid lahendusi. Teises peatükis antakse ülevaade funktsionaalsetest nõuetest. Kolmandas peatükis esitatakse mobiilirakenduse arhitektuur, kirjeldus ja funktsionaalsus koos ekraanikuvadega..

Antud bakalaureusetöö loomisel kasutatakse järgnevaid tehnoloogiaid (Tehnoloogiad kirjeldatakse peatükis 4.1 Kasutatud tehnoloogiad):

- JavaScript (AngularJS)
- HTML5 (Ionic)
- CSS3
- Microsoft SQL
- ASP.NET MVC, C#
- Apache Cordova

2. Olemasolevad lahendused

Antud peatükis kirjeldatakse olemasolevaid rakendusi. Rakenduste kasutamine on jaotatud kaheks – rakendused, mida kasutatakse praegusel hetkel Eestis ning rakendused, mis on geoblokeeringu tõttu kättesaadavad ainult mujal maailmas.

2.1 Olemasolevad lahendused Eestis

Praegusel hetkel toimub tehnoloogiliselt inimeste lõunate kokkuleppimine eelkõige läbi Skype'i, Facebooki, Slacki ja Facebook Messengeri. Kõigil neist puudub lahendus uute inimestega lõunale minekuks just eelkõige tänu inimeste diskreetsusele. Olemasolevate lahendustena võib arvestada ka Tinderit ja OkCupidit, mis võimaldavad kasutajatel tutvuda uute inimestega, kuid lahendustel puudub professionaalne pool. Tegu on rakendustega, mis lubavad eranditega tutvuda ainult inimestega, kes on vastassoost, eesmärgiga luua inimsuhteid, mitte professionaalseid tutvusi. Olemasolevatest lahendustest kõige sarnasema mudeliga käesolevale mobiilirakendusele on LinkedIn. LinkedIn on ärilise suunitlusega sotsiaalvõrgustik, mis on loodud eelkõige tööpakkumisportaalidele konkurendiks. LinkedInis on võimalik kuvada oma *curriculum vitae*-d profiilina, otsida endale tööd, tutvuda uute inimestega ja palju muud. Kui võrrelda arendatud mobiilirakenduse mudelit ja LinkedIni mudelit, võib näha palju sarnasusi. Antud töö eeliseks on nišš, mis võimaldab inimestel keskenduda oma professionaalse võrgustiku laiendamisele ja teha seda ühendades kaks tegevust – söömine ja sotsialiseerumine.

2.2 Olemasolevad lahendused mujal maailmas

2.2.1 NeverEatAlone

Suurimaks konkurendiks ärimudeli ja kliendibaasi poolest on praegusel hetkel NeverEatAlone[2]. NeverEatAlone on mobiili- ja *desktop* rakendus, kus kasutajatel on võimalik firmasiseselt kokku leppida ühiseid lõunaid. NeverEatAlone klientideks on suurkorporatsioonid, kus väga paljud inimesed omavahel ei suhtle ning see rakendus võimaldab firmadel parandada oma töökeskkonda[3]. NeverEatAlone mobiilirakendusel puudub võimalus leppida kokku lõunaid inimestega, kes ei ole kliendiks oleva firma töötaja. Rakenduse põhieesmärgiks on firma kultuuri parandamine. Käesolevas töös arendatud rakenduse eeliseks NeverEatAlone ees on võimalus tutvuda inimestega, kes ei tööta samas keskkonnas.

2.2.2 LunchBuddy

LunchBuddy[4] puhul on tegu sarnase ärimudeliga rakendusega, kuid on suunatud ainult USA turule. Rakendus on mõeldud selleks, et luua suurematele gruppidele ühiseid lõunaid, mille käigus saab diskuteerida valitud teemade üle. Rakendus ei võimalda kahel inimesel lihtsasti valida söögikohta ja minna sinna lõunale. Rakendus on allalaetav ainult Apple Appstores ehk teistele platvormidele antud rakendus ei ole kättesaadav (01.05.2016. a seisuga). Käesolevas töös arendatud rakenduse eeliseks LunchBuddy ees on platvormist sõltumatus ehk prototüüpi on võimalik kergesti arendada ka teistele platvormidele.

2.2.3 LunchMatcher

LunchMatcher[5] on Dubais kasutatav mobiilirakendus, mida on allalaaditud 500-1000 korda. Rakendus võimaldab LinkedIniga sisse logides määrata raadiust, kust kasutajaid otsitakse ning süsteem kuvab antud piirkonnas kasutajaid, kes on märkinud, et sooviks sobival kellaajal lõunat süüa. Antud rakendus on ärioloogika poolest kõige tugevam konkurent, kuna saavutab nišiga arvatavasti tugeva turuosa Eestis. Seetõttu on tarvis, et OneMeal jõuaks potentsiaalsete klientideni enne kui LunchMatcher laiendab oma geograafilisi võimalusi.

3. Funktsionaalsuse nõuded

Antud rakenduse funktsionaalsed nõuded on kirjeldatud arvestades rakenduse rakenduslikku eesmärki ja eesmärgi saavutamist. Funktsionaalsus on üle kantud analüüsides lõunate kutsumise algoritmi päris elus. Põhivood on kirjeldatud kasutades *KISS* printsiipi ning analüüs on teostatud põhieesmärgiga, et rakendus oleks kergesti õpitav ning kasutajale võimalikult kergesti omandatav. Alternatiivsete voogudena on kirjeldatud kõik võimalused, mille käigus kasutaja saab põhivoost kõrvale kalduda, analüüsides rakenduse võimalusi ning rakendades *white-box* ja *black-box* testimismeetmeid.

3.1 Sisselogimine

Kirjeldus: Kasutaja logib sisse

Kasutaja: Kasutaja, kes soovib rakendusse sisse logida

Eeldused: Kasutajal on registreeritud Facebooki konto

Põhivoog:

- Kasutaja vajutab “Log in” nupule
- Kasutaja suunatakse Facebooki sisselogimise lehele
- Kasutaja sisestab e-maili aadressi või telefoninumbri ja parooli
- Kasutaja vajutab “Log in” nupule
- Esmakordsel kasutamisel suunatakse kasutaja õiguste kinnitamise lehele
- Kasutaja kinnitab, et soovib avaldada rakendusele oma andmed ja vajutab nupule “Continue”
- Kasutaja suunatakse maandumislehele

Tulemus: Kasutaja on sisse logitud.

Alternatiivsed vood:

- Kasutaja vajutab tagasi nupule enne, kui lõpetab isikustamise.
 - Kasutajale kuvatakse teade “There was a problem signing in!”
- Kasutaja poolt sisestatud e-maili aadress või telefoninumber ei ole seotud ühegi kontoga
 - Kasutajale kuvatakse teade “The email address or phone number that you’ve entered doesn’t match any account. Sign up for an account”
- Kasutaja poolt sisestatud parool ei ole õige

- Kasutajale kuvatakse teade “The password that you’ve entered is incorrect. Forgotten password?”
- Kasutajal on kasutajanimi sisestamata
 - Kasutajale kuvatakse teade “The email address or phone number that you’ve entered doesn’t match any account. Sign up for an account”.
- Kasutajal on parool sisestamata
 - Kasutajale kuvatakse teade “The password that you’ve entered is incorrect. Forgotten password?”
- Kasutaja ei kinnita rakenduse õigusi ja suundub tagasi
 - Kasutajale kuvatakse teade “There was a problem signing in!”

3.2 Välja logimine

Kirjeldus: Kasutaja logib välja

Eeldused: Kasutaja on sisselogitud

Põhivoog: Kasutaja vajutab menüüs nupule “Log out”

Alternatiivsed vood: Puuduvad.

Tulemus: Kasutaja suunatakse sisselogimise vaatesse ning seansiinfo kustutatakse brauserist.

3.3 Lõunasoovi lisamine

Kirjeldus: Kasutaja esitab lõunasöögiavalduse, mis kuvatakse teistele kasutajatele.

Eeldused: Kasutaja on sisselogitud.

Põhivoog:

- Kasutaja asub maandumislehel
- Kasutaja vajutab jaluses olevale kohvitassi nupule
- Kasutaja vajutab kuupäevale ning talle kuvatakse modaalaken kuupäeva valikuga
- Kasutaja valib sobiva kuupäeva ja kellaaja
- Kasutaja vajutab nupule “OK”
- Kasutaja valib söögikoha, kus soovib lõunat süüa
- Kasutaja vajutab nupule “Send”

- Kasutaja suunatakse maandumislehele

Alternatiivsed vood:

- Kasutaja sisestab kuupäeva ja kellaaja, mis on minevikus
 - Kasutajale kuvatakse “*Send*” nupuvajutuse peale veateade “Your OneMeal date is in the past. Please choose a new one”
- Kasutaja jätab lõunasöögi koha valimata.
 - Kasutajale kuvatakse “*Send*” nupuvajutuse peale veateade “Please choose a location for your OneMeal”

3.4 Lõunakutse saatmine

Kirjeldus: Kasutaja leiab sobiva lõunasöögiavalduse ning saadab avalduse teinud kasutajale kutse.

Eeldused:

- Kutset saatev kasutaja on sisselogitud
- Süsteemis on autonoomseid, tulevikus toimuvaid lõunasöögikuulutusi.

Põhivoog:

- Kasutaja valib välja maandumislehel olevatest autonoomsete lõunasöögiavaldustest endale kõige sobivama
- Kasutaja vajutab väljavalitud lõunasöögiavaldusel.
- Kasutajale kuvatakse lõunasöögiavalduse info ja kasutaja profiil
- Kasutaja tutvub lõunasöögiavalduse teinud kasutaja profiiliga ja veendub, et soovib kasutajaga lõunale minna
- Kasutaja vajutab kinnitamiseks nupule „*Send*“ ning suunatakse maandumislehele

Alternatiivsed vood:

- Kasutajale ei sobi väljavalitud lõunasöögiavaldusega seotud kasutaja
 - Kasutaja vajutab oma mobiiltelefoni tagasi nupule ja suunatakse maandumislehele ilma kutset saatmata
 - Kasutaja vajutab jaluses olevale kodu nupule ja suunatakse maandumislehele ilma kutset saatmata

3.5 Lõunakutse vastuvõtmine

Kirjeldus: Kasutaja võtab vastu talle saadetud lõunasöögikutse.

Eeldused:

- Kutset aksepteeriv kasutaja on sisselogitud
- Kasutajale on saadetud lõunasöögikutse(id)

Põhivoog:

- Kasutaja asub maandumislehel
- Kasutaja vajutab jaluses olevale inimeste nupule ning suunatakse ajaloo vaatesse
- Kasutaja valib „*Pending*“ lõunate alt välja kutse, mida soovib vastu võtta
- Kasutaja vajutab väljavalitud lõunasöögikutsele
- Kasutaja tutvub lõunasöögikutse esitanud kasutaja profiili ja lõunasöögi infoga
- Kasutaja vajutab kinnitamiseks nupule „*Send*“ ning suunatakse maandumislehele

Alternatiivsed vood:

- Kasutajale ei sobi väljavalitud lõunasöögikutse esitanud kasutaja ja ei soovi seda enam näha
 - Kasutaja hoiab näppu kutse peal ning tõmbab näppu ekraanil vasakule
 - Kasutajale kuvatakse kinnitusteade: “Are you sure you wish to reject the OneMeal?” kus kasutaja kinnitab, et soovib kutse kustutada
- Kasutaja soovib lõunasöögi kutse vastu võtta ilma profiili nägemata
 - Kasutaja hoiab näppu kutse peale ning tõmbab näppu ekraanil paremale
 - Kasutajale kuvatakse kinnitusteade: “Are you sure you wish to accept the OneMeal?” kus kasutaja kinnitab, et soovib kutse vastu võtta

3.6 Kasutaja profiili muutmise

Kirjeldus: Kasutaja muudab oma profiili andmeid.

Eeldused:

- Kasutaja on sisselogitud

Põhivoog:

- Kasutaja avab menüüst „*Profile*“

- Kasutaja teeb vajalikud muudatused väljadel „*About*“, „*Profession*“ ja „*Keywords*“
- Kasutaja kinnitab muudatused ja vajutab päises olevale nupule „*Submit*“

4. Rakenduse arhitektuur

Rakenduse saab arhitektuuriliselt jaotada kaheks – mobiilirakendus ning veebiteenus. Antud peatükk kirjeldab käesoleva mobiilirakenduse ja veebiteenuse kasutatud tehnoloogiaid, rakenduse kontrollereid ja klasse ning andmemudelit.

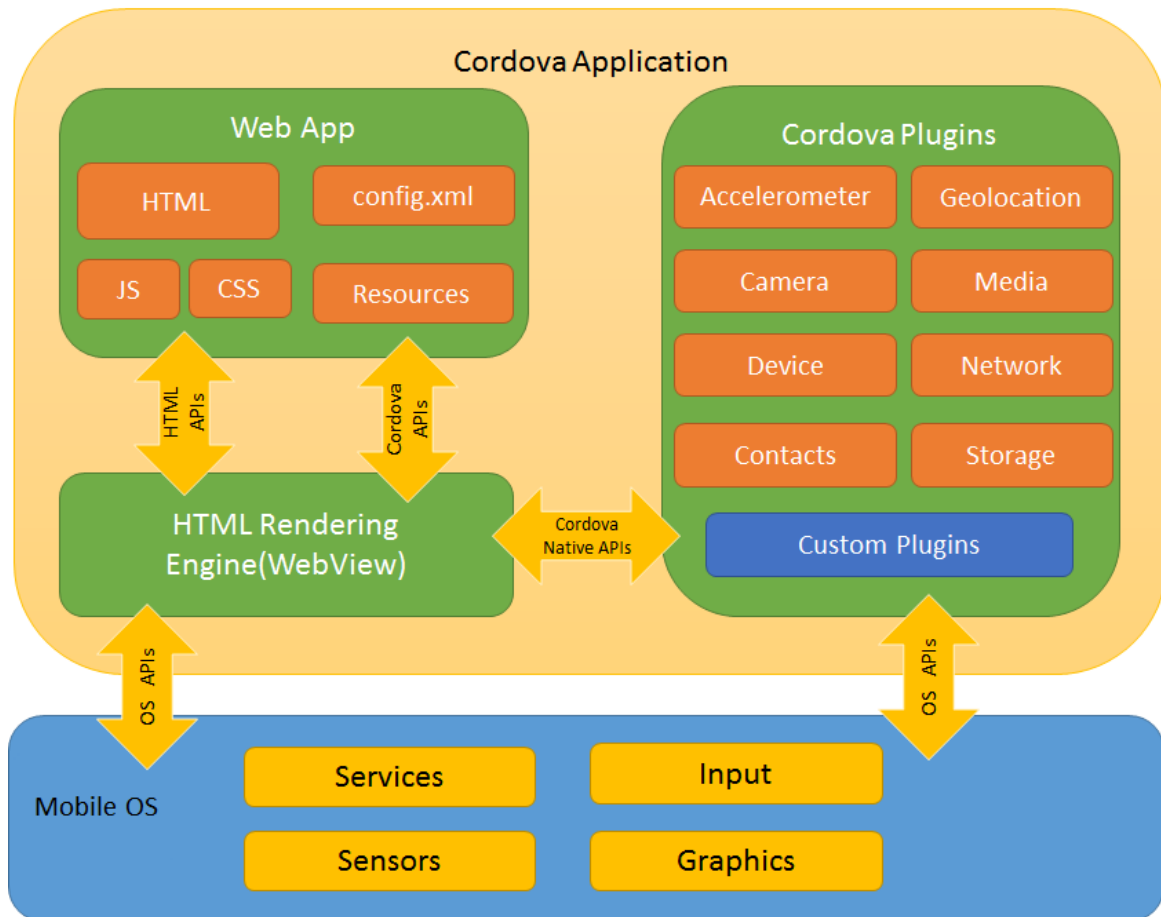
4.1 Kasutatud tehnoloogiad

4.1.1 Apache Cordova

Apache Cordova on vabavaraline mobiili arendus-raamistik, mis toetab kõiki suuremaid platvorme (Android, iPhone, BlackBerry ja Windows Phone). Apache Cordova võimaldab arendada platvormist sõltumatusid mobiilirakendusi kasutades kolme veebitehnoloogiat: JavaScript, HTML5 ja CSS3. Apache Cordova raamistik võimaldab rakendusel kasutada mobiiltelefonide olemasolevat võimekust (riistvara, mis on ühendatud tarkvaraga). Apache Cordova põhiliseks eeliseks teiste raamistike ees on just platvormist sõltumatus. Raamistik võimaldab seda tänu brauseritehnoloogiate kasutusele. Nimelt kasutavad kõik mobiilplatvormid brauserites lehekülgede kuvamiseks just nimelt neid kolme keelt: JavaScript, HTML5 ja CSS3. Seetõttu peitub minimaalse ärioloogikaga mobiilirakenduse arendamise juures võimalus teha seda kõikidele platvormidele korraga [6].

4.1.1.1 Arhitektuur

Apache Cordova võib struktuurselt jaotada kolmeks komponendiks (vt joonis 1). Esimene komponent on veebirakendus, milles elab rakenduse kood. Rakendust hoitakse kui veebilehekülge, mille iga lehekülg on defineeritud *.html* faililaiendiga ning, kus iga veebilehega on seotud JavaScript, CSS, pildid, logod ja kõik muu mida rakenduse tööks kasutatakse. Rakendus elab ise veebi-vaates, mis on kohandatud vastavalt mobiilplatvormile. Veebi vaade on brauser, mida platvorm kasutab *World Wide Web*'i lehtede kuvamiseks, ilma brauseri menüü ja tööriistaribata[7]. Tavaliselt veebibrauserist erineb Apache Cordova oma kolmanda komponendi poolest. Nimelt pakub Apache Cordova liidestust brauseri ja pistikprogrammide vahel. Pistikprogrammid võimaldavad kasutada sama riistvaralist ja tarkvaralist võimekust kui arendada igale platvormile omarakendus.



Joonis 1. Apache Cordova arhitektuur [6]

4.1.2 Ionic

Ionic on vabavaraline tarkvaraarenduse komplekt, mis võimaldab ehitada platvormist sõltumatu mobiilirakendusi arvestades kasutaja harjumusi konkreetse mobiiliplatvormiga. Ionic kasutab kasutajaliidese kuvamiseks kõiki kolme veebitehnoloogiat: HTML, CSS ja JavaScript. Ionic ei ole mõeldud asendusena Apache Cordovale, vaid on ehitatud Cordovale juurde, et pakkuda arendajatele lisavõimalusi. Lisavõimalusteks on eeldefineeritud välimuse ja omadustega staatiliste veebikomponentide kasutamine. Nimelt kasutatakse Ionicut põhiliselt kasutajaliidese kuvamiseks ja seda eelkõige koostöös AngularJS-ga [8].

4.1.3 JavaScript, AngularJS

JavaScript on maailma üks populaarsemaid programmeerimiskeeli ja *World Wide Webi* üks kolmest kõige rohkem kasutatavast keelest. JavaScript on dünaamiline programmeerimiskeel, mida saab arendada objektorienteeritult. Antud bakalaureusetöökontekstis on JavaScripti kasutatud veebilehe skriptina, mis võimaldab dünaamiliselt väärtustada kasutajale kuvatavaid andmetüpe ja nende väärtusi.

AngularJS on Google poolt arendatud avatud lähtekoodiga JavaScripti raamistik, mida kasutatakse, et arendada ühe-leheküljelisi rakendusi, mis sarnanevad töölaua tarkvaraga. AngularJS pakub raamistikku, et arendada eessüsteem, mis kasutab mudel-vaade-kontroller raamistikku. AngularJS on valmistatud eeldusega, et seda kasutatakse deklaratiivse programmeerimise paradigmana, kus ei defineerita ära koodi juhtimisvoogu. Seeläbi võimaldab see arendajal luua kasutajaliidseid ning ühendada tarkvara komponentidega. AngularJS eeliseks teiste lahenduste juures on koormuse vähendamine serveri poolelt. Tänapäeva arendusmeetodid viivad arvutuskoormuse maksimaalselt eesliidesele, et koormata võimalikult vähe serverit ja rakendada võimalikult palju loogikat iga kliendi liidese[9].

4.1.4 ASP.NET MVC, C#

Active Server Pages tehnoloogia edasiarendus ASP.NET on üks olulisemaid komponente Microsofti poolt arendatud rakenduste arendamise platvormis .NET. ASP.NET raamistikku kasutatakse, et arendada dünaamilisi veebilehekülgi, veebirakendusi ja veebiteenusi [11]. Antud mobiilirakenduse veebiteenus on arendatud kasutades ASP.NET MVC'd, mis on ASP.NET edasiarendus. MVC ehk *model-view-controller* tehnoloogia võimaldab struktureerida rakenduse kolme rolli: mudel, vaade ja kontroller.

Mudeli kujutab endast rakenduse hetkestaatust ning hoiab endas ärilise kihi andmeid. Kontroller võtab vastu interaktsioone ja värskendusi ning vastavalt sellele uuendab mudelit. Seejärel edastab kontroller vaatele uue mudeli, mis kuvatakse kasutajale. Antud veebiteenus kasutab oma loogikas ainult mudelit ja kontrollerit. Mudeliks on andmevormid, mis täidetakse kas mobiilirakenduse poolt saadetava infoga ja seejärel saadetakse uuendusena andmebaasi või päritakse andmebaasist ja saadetakse mobiilirakendusele. Käesolevas veebiteenuses on kontroller see osa, mis vastutab päringute vastuvõtmise, kontrollerile vastava äriloojika rakendamise ning päringule vastamise eest[10].

4.1.4.1 Arhitektuur

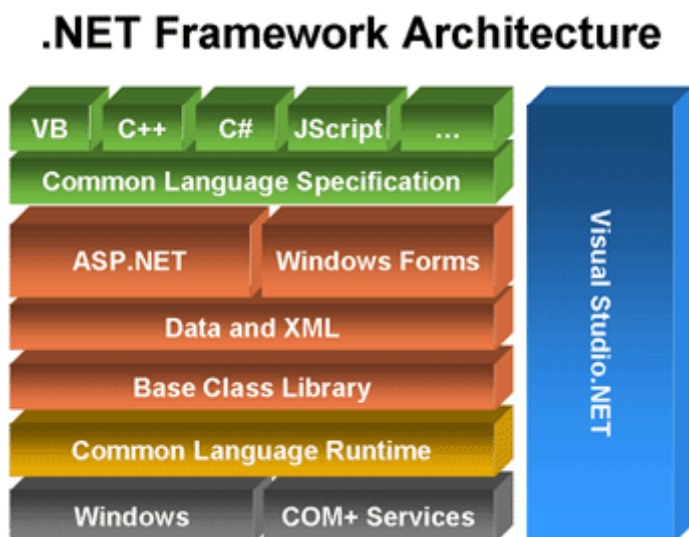
CLR ehk *common language runtime* moodustab .NET raamistiku aluse ning vastutab käivitumisaegse koodi täitmise eest. CLR on see osa arhitektuurist, mis käitab arendatud komponentide ja rakenduste koodi ning tegeleb automaatse mäluhaldamise ja -koristamisega. CLR võimaldab arendajal kasutada soovitud funktsionaalsust pakkudes selleks raamistiku parameetreid, meetodeid ja sündmusi, mida on võimalik vastavalt

soovitud funktsionaalsuse laiendada. Lisaks võimaldab CLR hallata moodulite erinevaid versioone, mis lihtsustab sama komponendi erinevate versioonide haldamist arendus- või päriskeskkonnas.

Oranžina on tähistatud joonisel 2 .NET raamistiku klasside teegid, mis pakuvad väga laia valiku erinevaid klasse, mis võimaldab arendada ärile vastavat loogikat. ADO.NET, mis kuulub *Data and XML* kihti võimaldab pärida, liigutada ja töödelda andmeid relatsioonilistest andmebaasidest. Käesoleva mobiilirakenduse veebiteenus kasutab ADO.NET tehnoloogiat, et töödelda andmebaasis olevaid kirjeid[11].

Roheline kiht kujutab endast rakenduse arendamise tehnoloogiaid. CLS ehk *Common Language Specification* kihti kuulub ka ASP.NET, mida antud rakendus kasutab veebiteenuse loomiseks. CLS on kogum defineeritud featuuridest, mida paljud rakendused kasutavad ning see võimaldab programmeerimiskeelel kergesti teostada rakenduse eeldefineeritud funktsioone[12].

Kõige pealmise kihi moodustavad programmeerimiskeeled. Antud rakenduse veebiteenus on arendatud kasutades C# programmeerimiskeelt. C# on objektorienteeritud keel, mis on välja arenenud C programmeerimiskeelest ning võimaldab lisaks tavapärasele funktsionaalsusele rakendada ka funktsionaalse programmeerimise meetodeid ning lihtsustatud klasside pärimist.



Joonis 2. .NET raamistiku arhitektuur [13]

4.1.5 Microsoft SQL

Microsoft SQL Server on relatsioonilise andmebaasi haldussüsteem, mille on arendanud Microsoft[14]. Tegu on tasulise tarkvaraga, mida kasutatakse tarkvara andmete hoiustamiseks ja pärimiseks. Antud mobiilirakenduses on andmebaas virtuaalmasinal, mis asub Azure poolt pakutavas pilves. Azure üks paljudest teenustest on Microsofti pilveversioon andmebaasist, mida pakutakse tarkvara teenusena. Rakenduse serveripoolne süsteem kasutab Microsoft andmebaasiga suhtlemiseks LINQ to SQL'i. Antud rakenduse serveri-poolne kood kasutab LINQ to SQL lahendust, kus LINQ päringust genereeritakse SQL päring, mis pärib andmebaasist päringule vastavad andmed. Antud lahenduses on kasutatud matemaatilist analüüsist tuntud lambda avaldisi (vt. joonis 3), kus defineeritakse andmete kogust pärimisel funktsioon, mis tagastab ette antud väärtustele vastavad tulemid[15].

```
List<Meals> activeMeals = db.Meals.Where(x => x.MealDate >= DateTime.Now && x.PartnerID == null).ToList();
```

Joonis 3. Lambda avaldise näidis C# koodis

4.1.6 Microsoft Azure

Microsoft Azure on Microsofti poolt veebruaris 2010 avaldatud andmetöötlusplatvorm ja taristu, mis asub pilves. Varasema nimega Windows Azure toetab paljusid programmeerimiskeeli, tööriistu ja raamistike, kuhu kuuluvad nii Microsofti spetsiifilised kui ka kolmanda osapoole poolt loodud tarkvara ja süsteemid. Microsoft on jaotanud Azure teenused 11 tüübiks:

- Arvutus - teenus, mis pakub virtuaalseid masinad ja virtualiseerimist
- Veeb ja mobiil – veebi- ja mobiilirakenduste arendamine ja hoiustamine, veebiteenuste manageerimine ning statistika
- Andmehoidlad – andmebaas teenusena, mis võimaldab kasutada nii SQL kui ka NoSQL-i
- Analüüs – hajussüsteemide ja suurandmete analüüs
- Võrgustus – virtuaalsed võrgud ja nimeserveri hostimine
- Meedia ja sisulevivõrk – teenus, mis võimaldab luua voogedastust, meedia kodeerimist ning taasesitust
- Hübriidne integratsioon – võimaldab servereid varundada ja taastada
- Isikustamine ja õiguste haldus

- Esemevõrk – teenused, mis abistavad talletada, hoiustada, monitoorida ja analüüsida esemevõrgu andmeid
- Arendus – teenused, mis võimaldavad arendajatel jagada koodi, testrakendusi ja hoida silma peal arendustsüklitel
- Haldamine ja õigused – tooted, mis võimaldavad administraatoritel hallata Azure paigaldamist, planeerida ja jooksutada ajastatud töid ning automatiseerida protsesse. Samuti võimaldab hallata veebiplatvormi õigustepääsmeid

Antud mobiilirakenduse veebiteenus jookseb Microsoft Azure ja veebiteenuse andmeid hoitakse Microsoft Azure SQL serveris. Käesolev mobiilirakendus kasutab Azure veebi ja mobiili lahendust. Antud tehnoloogia sai valitud käesoleva töösse kuna võimaldab ASP.NET arendusplatvormi hõlpsasti integreerida Microsoft Azure'ga [20].

4.1.7 CSS3

Cascading Style Sheets (CSS) ehk kaskaadlaadistik on arvuti keel, mida kasutatakse HTML dokumentide kuvamisreeglite defineerimiseks brauserites. Antud keelt kasutatakse käesoleva töö käigus Ionicu poolt, et kuvada rakenduse kasutajale liideseid ning vaateid spetsiaalselt defineeritud välimuse alusel. CSS3 võimaldab kasutaja rakenduse kasutamise kogemust mugandada muutes komponentide paigutust, värve, paistvust, mis võimaldab saavutada kontrolli kasutaja kogemuse üle [16].

4.1.8 HTML5

HTML5 ehk *HyperText Markup Language* on veebilehtede märgendkeele viies versioon. HTML failide kuvamiseks kasutatakse veebilehitsejaid ehk brausereid. Kuna kõik mobiilplatvormid kasutavad veebilehtede kuvamiseks brausereid, siis antud ühisosa platvormide vahel võimaldab arendada platvormist sõltumatut lahendust. Antud viimane versioon HTML'ist on loodud eelkõige parendusena multimeedia seadmetele, kaotamata keele lihtsust ja loetavust. HTMLi kasutatakse dokumendi struktuuri defineerimiseks, kus on ära mainitud ka kasutatavad CSS'i kujundused ning JavaScript, mis tekitab veebilehes dünaamilisust[17].

4.2 Kontrollerid

4.2.1 Veebiteenuse kontrollerid

Veebiteenus koosneb viiest erinevast kontrollerist, mida rakendus kasutab suhtlemiseks mobiilirakendusega, et töödelda, pärida ja hoiustada andmeid.

- **ActiveProfilesController**

Antud kontrolleri eesmärk on töödelda kahte *Get* päringut ning ühte *Post* päringut.

Esimene *Get* päring saab sisendiks mobiilirakendusest koordinaadid ning kauguse ja selle põhjal tagastab kõik lõunasöögiatlused, mis ei asu parameetrina antud kaugusest eemal.

Teine *Get* päring saab sisendiks rakendusest lõunasöögiavalduse identifikaatori ja kasutaja profiili identifikaatori ning tagastab lõunasöögi ja kasutaja profiili info.

Post päringut kasutatakse lõunasöögiavalduse lisamiseks süsteemi, kus parameetriks saadetakse kuupäev, lõuna toimumiskoht ning kasutaja, tulemusena lisatakse andmebaasi uus lõunasöögiavaldus.

- **ArchiveMealsController**

Antud kontrolleri eesmärk on töödelda *Get* päringut, mis saab sisendiks kasutaja identifikaatori ning tagastab andmebaasist kõikide lõunate kirjed, kus sisendina etteantud kasutaja on osalenud ning edastatakse need kirjed mobiilirakendusele.

- **FacebookProfileController**

Antud kontrolleri eesmärk on töödelda *Post* päringut, mida kasutatakse kasutaja profiili lisamiseks või muutmiseks. Sisendiks saab päring kasutaja identifikaatori, pääsmiku, ees- ja perekonnanime, töökoha, sünnipäeva, kirjelduse ja märksõnad. Süsteem kontrollib, kas kasutaja kirje on juba andmebaasis olemas. Kui ei ole, siis lisab selle, ning kui on, muudab kasutaja kirjeldust, märksõnu ja töökoha infot.

- **OneMealController**

Antud kontrolleri eesmärk on töödelda *Post* päringut, mida kasutatakse lõunasöögikutsete haldamiseks. Päringu sisendiks antakse lõunasöögiavalduse identifikaator, tõeväärtus lõuna kutse olemasolu kohta ning teise osapoole identifikaator. Kui lõuna kutse on olemas, siis on tegu kutse kinnitamisega ning autonoomne lõunasöögiavaldus seotakse ära teise osapoolega. Kui kutset ei ole, lisatakse lõunasöögi kutsete tabelisse uus kirje.

- **PendingMealsController**

Antud kontrolleri eesmärk on töödelda *Get* päringut, mis saab sisendiks kasutaja identifikaatori ning tagastab andmebaasist kõikide lõunate kirjed, kus sisendina etteantud kasutaja on lõunasöögile kutsutud ning edastatakse need kirjed mobiilirakendusele.

4.2.2 Mobiilirakenduse kontrollid

Mobiilirakendus koosneb kuuest kontrollist, kus iga vaade on omaette kontroll. Rakendus kasutab kontrollereid, et defineerida iga vaate skoobi muutujad, funktsioonid ning kui vaja teostatakse päring rakenduse veebiteenusele, et pärida andmed, mille alusel täidetakse muutujad ja asustatakse skoobi väärtused. Igas kontrollis kontrollitakse, kas kasutajal on brauseri sessioonis aktiivne pääsmis ja pannakse paika ligipääsuõigused.

- **LoginController**

Kontroller, kus pääsmiku puudumisel teostatakse rakendusse sisse logimine, kasutades selleks Facebooki autoriseerimist. Kui autoriseerimine on edukas, kirjutatakse kasutaja pääsmik brauseri sessiooni ning edastatakse kasutaja info veebiteenuse FacebookProfileControllerle.

Kui kasutajal on aktiivne sessioonivõti, suunatakse kasutaja edasi maandumislehele.

- **LogoutController**

Kontroller, kus kustutatakse kasutaja sessioonialalugu ning suunatakse sisselogimise vaatesse.

- **MainScreenController**

Kontroller, kus saadetakse veebiteenusele päring kasutaja mobiili koordinaatidega ning päritakse ja omistatakse kõik aktiivsed lõunasöögiavaldused.

- **DateTimePickerControl**

Kontroller, kus defineeritakse ära meetod uue avalduse saatmiseks veebiteenusele.

- **ProfileController**

Kontroller, kus vastavalt eeldefineeritud loenditüübile (lõunasöögi kutse saatmine või vastuvõtmine) defineeritakse ära funktsioon, mille alusel saadetakse päring veebiteenusele. Lisaks tehakse päring veebiteenusele, et pärida info profiili andmete kohta, kus kasutaja hetkel viibib.

- **MyProfileController**

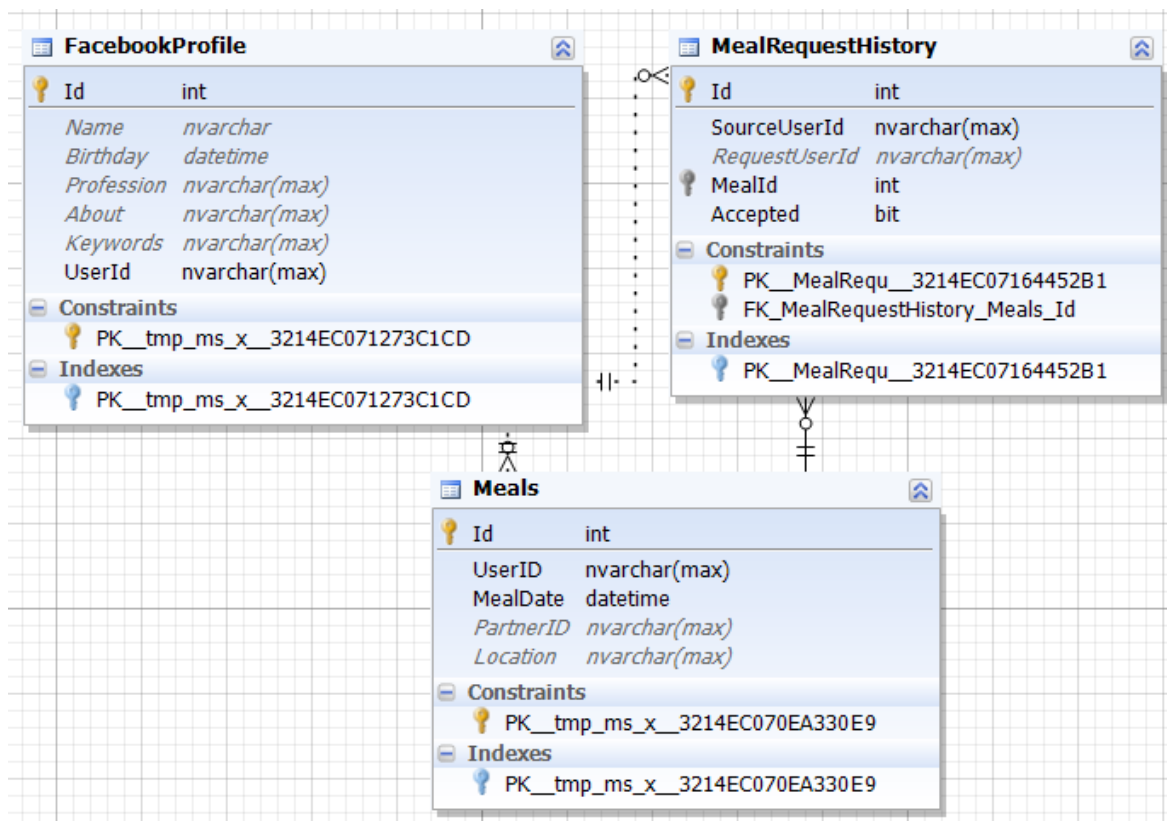
Kontroller, kus defineeritakse ära funktsioon, mis saadab muudetud profiiliinfo andmed veebiteenusele. Eelnevalt tehakse päring veebiteenusele, et pärida kasutaja profiiliandmed, mida kuvatakse kasutajale vaates.

- **MyHistoryController**

Kontroller, mis teeb kaks päringut veebiteenusele ning pärib lõunasöögikused ja arhiveeritud lõunasöögid, mis kuvatakse kasutajale antud vaates.

4.3 Andmemudel

Käesolev andmemudel kirjeldab mobiilirakenduse andmete hoiustamise struktuuri relatsioonises Microsoft Azure andmebaasis. Andmebaas on üles ehitatud kolme tabelina (vt. joonis 4): profiilid, lõunad ning lõunakutsed.



Joonis 4. Mobiilirakenduse andmemudel

Relatsioonilise andmebaasi tabelid on järgmised.

- 1) **FacebookProfile** – tabel, kus hoitakse registreerunud kasutaja kontaktandmeid.

Tabelis on järgnevad andmeväljad:

- *Id* – unikaalne number, mis identifitseerib kasutajat

- *Name* – kasutaja ees- ja perekonnanimi
 - *Birthday* – kasutaja sünnipäeva kuupäev ja aasta, kujul *dd-MM-yyyy hh:mm:ss*
 - *Profession* – kasutaja töökoht
 - *About* – kasutaja reklaamlause
 - *Keywords* – kasutaja märksõnad
 - *UserId* – kasutaja unikaalne identifikaator Facebook keskkonnas
- 2) ***Meals*** – tabel, kus hoitakse autonoomseid ja kinnitatud lõunasöögiavaldusi.

Tabelis on järgnevad andmeväljad:

- *Id* – unikaalne number, mis identifitseerib lõunasöögikuulutust
 - *UserID* – kuulutuse lisanud kasutaja *UserId*
 - *MealDate* – lõuna toimumise kuupäev ja kellaaeg kujul *dd-MM-yyyy hh:mm:ss*
 - *PartnerID* – kinnitatud lõunasöögikuulutuse puhul kasutaja *UserId*, kes on kinnitatud lõunasöögikaaslaseks
 - *Location* – söögikoha nimi, kus toimub lõuna
- 3) ***MealRequestHistory*** – tabel, kus hoitakse lõunasöögikutseid.

Tabelis on järgnevad andmeväljad:

- *Id* – unikaalne number, mis identifitseerib lõunasöögikutset
- *SourceUserId* – autonoomse lõunasöögiavalduse postitanud kasutaja *UserId*
- *RequestUserId* – lõunasöögikutse saatnud kasutaja *UserId*
- *Accepted* – tõeväärtus, kas lõuna on vastuvõetud (Jah/Ei)

5. Rakenduse kirjeldus

5.1 Päis, jalus ja menüü

Päis, jalus ja menüü on kasutajaliidese osad, mida kuvatakse iga vaate juures.

Päis asub igal lehel kõige üleval ning seal kuvatakse:

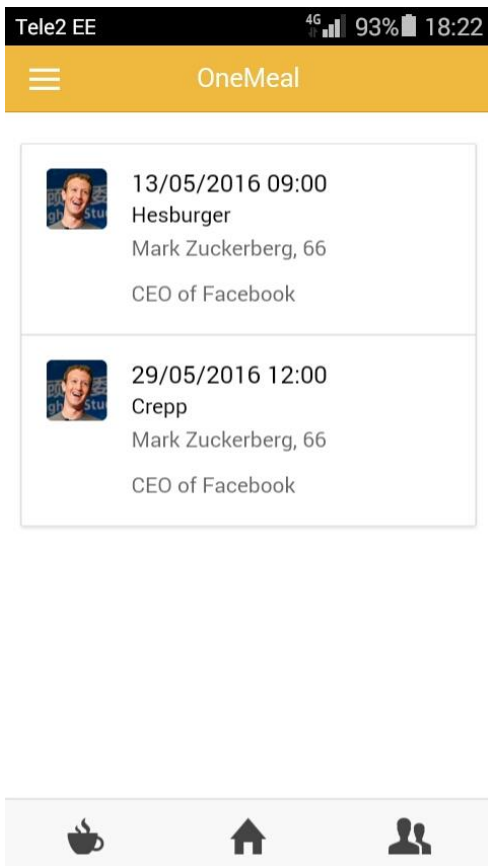
- menüü nappu, mis kuvab rakenduse vasakule osasse järgnevad menüü valikud
- rakenduse logo, millele vajutades suunatakse kasutaja maandumislehele
- nõustumisnappu, mida ei kuvata kõikidel vaadetel

Päis on defineeritud index.html lehel, mida süsteem kasutab kõikide vaadete puhul. Konkreetne lehekülje vaade, kus kasutaja paikneb, lisatakse rakenduse poolt index.html-ile juurde (vt. joonis 5 allpool).

Jalus asub igal lehel kõige all ning seal kuvatakse järgnevaid nuppe:

- Kohvitass – võimalus kasutajal avaldada soovi lõunale minekuks
- Kodu – suunab kasutaja maandumislehele
- Inimesed – suunab kasutaja vaatesse, kus kuvatakse aktiivsed lõunakutsed ning lõunate ajalugu

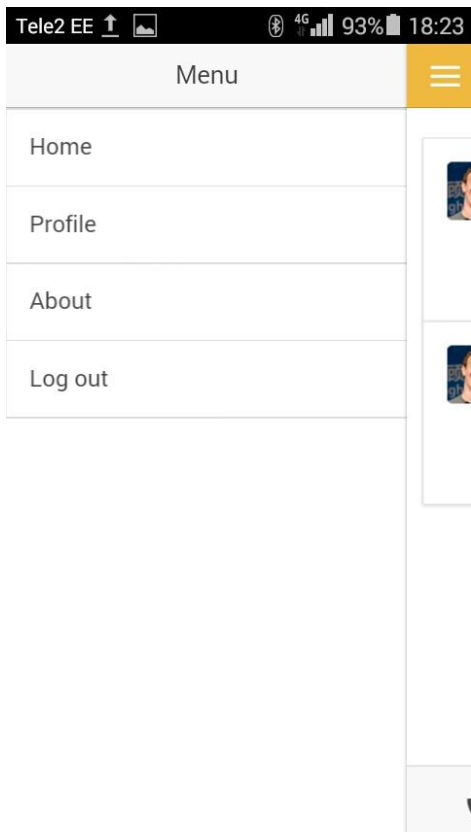
Jalus on defineeritud igal lehel eraldi ning käib iga vaatega eraldi kaasas (vt. joonis 5 allpool).



Joonis 5. rakenduse päis ja jalus

Menüü asub igal lehel vasakul ning on vaikimisi peidetud. Liigutades rakenduse vasakust äärest sõrme paremale või vajutades menüü nupule, kuvatakse menüü 90% mahus üle ekraani. Menüü on defineeritud index.html lehel (vt. joonis 6 allpool). Menüü saab sulgeda, liigutades näppu paremast äärest vasakule või vajutades uuesti menüüle. Menüüs kuvatakse järgnevaid elemente:

- *Home* – nupp, mis suunab maandumislehele
- *Profile* - nupp, mis kuvab kasutaja profiili nii, nagu see teistele kasutajatele paistab
- *About* – nupp, mis kuvab rakenduse tegija kontaktandmeid
- *Log out* – nupp, mis kustutab kasutaja sessioonivõtme ning lõpetab sessiooni



Joonis 6. Rakenduse menüü

5.2 Maandumisleht

Mobiilirakenduse maandumisleheks nimetame vaadet, kuhu kasutaja suunatakse, kui tal on aktiivne sessionivõti. Maandumislehe kuva on defineeritud lehel *mainscreen.html*. Maandumislehe eesmärgiks on kuvada kõiki kasutajaid, kes on avaldanud soovi minna lõunale. Süsteem kuvab ainult lõunaid, mis ei ole kuupäevaga minevikus ja mille soovi avaldanud kasutaja ei ole leidnud endale kaaslast. Maandumisleht koosneb *gridview*-st, mis genereeritakse asünkroonselt vaate avamisel. Kui kasutaja suunatakse maandumislehele, saadab rakendus *get* päringu rakenduse veebiteenusele, mis on paigaldatud Microsoft Azure'i. *Get* päring saadetakse *ActiveProfiles* kontrolleri, mis tagastab *JSON* kujul kõikide aktiivsete lõunasöögide andmed koos lõunasöögisoovi lisanud kasutaja infoga. Maandumislehel kuvatakse iga kirje kohta, mille veebiteenus tagastab, üks rida. Iga rea kohta kuvatakse kasutaja Facebooki profiilipilti, lõuna kuupäeva koos kellaaajaga, kasutaja nime, kasutaja vanust ning ametikohta (vt joonis 5).

5.3 Profiili kuva

Maandumislehel *gridview* kirjele vajutades suunatakse kasutaja konkreetse lõunasooviga seotud kasutaja profiililehele. Profiililehe vaade on defineeritud *profile.html* lehel. Profiili vaates kuvatakse kasutaja profiilipilti, nime, vanust, ametit, kasutaja teavet ning võtmesõnu. Andmed küsitakse kasutades *Get* päringut rakenduse veebiteenuselt. Parameetrina antakse kaasa valitud lõunasöögi identifikaator ning kuvatava kasutaja identifikaator, mille alusel *ActiveProfiles* kontrolleri tagastab konkreetse lõunasöögiga seotud kasutaja info. Veebiteenusest tagastatud päringu *JSON*'i andmemudel väärtustatakse profiili kuva kontrolleri skoobi muutujaks. Skoobi muutuja andmemudeli põhjal väärtustatakse omakorda profiili kuva kõik väljad. Kui kasutaja soovib antud isikuga minna lõunatama, vajutab ta üleval paremal nurgas olevale päise nupule ning rakendus saadab *post* päringu rakenduse veebiteenuse kontrolleri *OneMealController*, mis tekitab andmebaasi uue lõunasöögiavalduse kirje (vt joonis 6 allpool).

Aktiivsete kutsete vaatest vajutades kasutaja *gridview* kirjele suunatakse kasutaja profiili kuva vaatesse. Kasutajal on võimalik aktiivne kutse vastu võtta vajutades üleval paremal päises olevale nupule „*Send*“. Nupuvajutuse peale tehakse *post* päring rakenduse pilves olevasse veebiteenusesse. Veebiteenus kustutab ära kõik algse lõunasooviga seotud kutsed ning lisab lõunasöögiavaldusele teise osapoole identifikaatori. Seejärel liigutatakse lõunasöögi kirje mõlema osapoole arhiivi.

Arhiivis olevate lõunate *gridview* kirjele vajutades suunatakse kasutaja profiili kuva vaatesse. Kasutajal on võimalik vaadata infot kasutajatest, kellega ta on käinud lõunal või on tulevikus minemas lõunale.

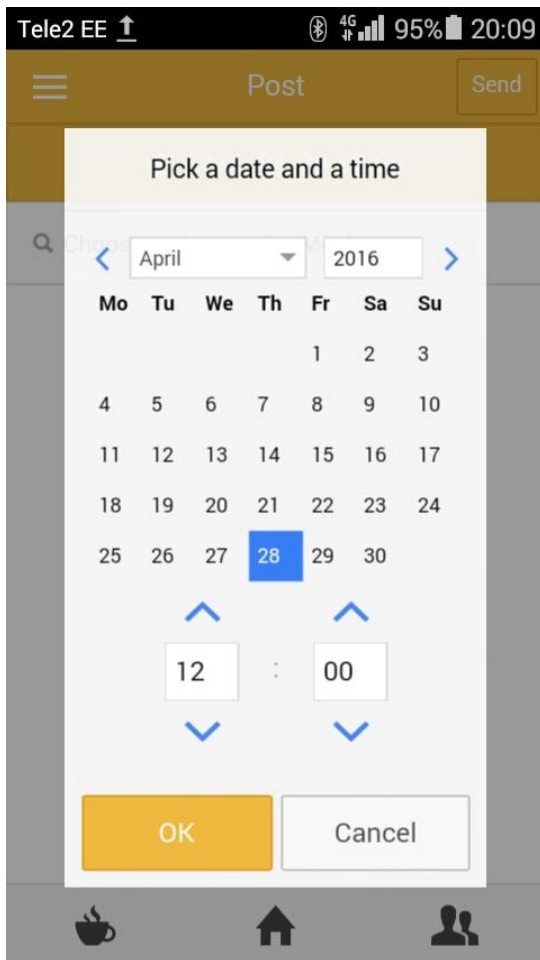


Joonis 6. Profili vaade kasutaja lõunale kutsumiseks

5.4 Lõuna postitamine

Lõuna postitamise vaade on defineeritud lehel *post.html* ning antud vaates on kasutajal võimalik postitada oma lõunasöögikuulutus, mida kuvatakse kõikidele 30 km raadiuses olevate kasutajate maandumislehele. Leht võimaldab kasutajatel valida kuupäeva ja kellaega ning sisestada lõuna asukohta. Kuupäeva valimiseks peab kasutaja vajutama vaikumisi kuvatud kuupäevale, milleks on käesolev kuupäev kell 12:00. Nupuvajutuse peale kuvatakse kasutajale modaali dialoog (vt joonis 7 allpool), kus kasutaja saab muuta kuupäeva ja kellaega. Kuupäeva valik on realiseeritud kasutades Ionic-DateTime-picker teeki [18]. Uue kuupäeva valimisel muudab rakendus vaikumisi valitud kuupäeva kasutaja poolt valitud kuupäevaks. Kui kasutaja soovib oma kuulutuse kinnitada, siis päises olev parempoolne nupp „Send“ edastab kuulutuse süsteemi. Kinnitamise nupuvajutuse peale teeb mobiilirakenduse kontrollile *post* päringu rakenduse veebiteenuse kontrollile *ActiveProfiles*. *Post* päringule antakse kaasa parameetrina andmemudel, kus on defineeritud

valitud kuupäev ja kellaaeg, avalduse teinud kasutaja identifikaator ja kasutaja sisestatud söögikoht. Pärast päringu teostamist suunatakse kasutaja maandumislehele.



Joonis 7. Hüpikaken, mis võimaldab kasutajal täpset aega määrata

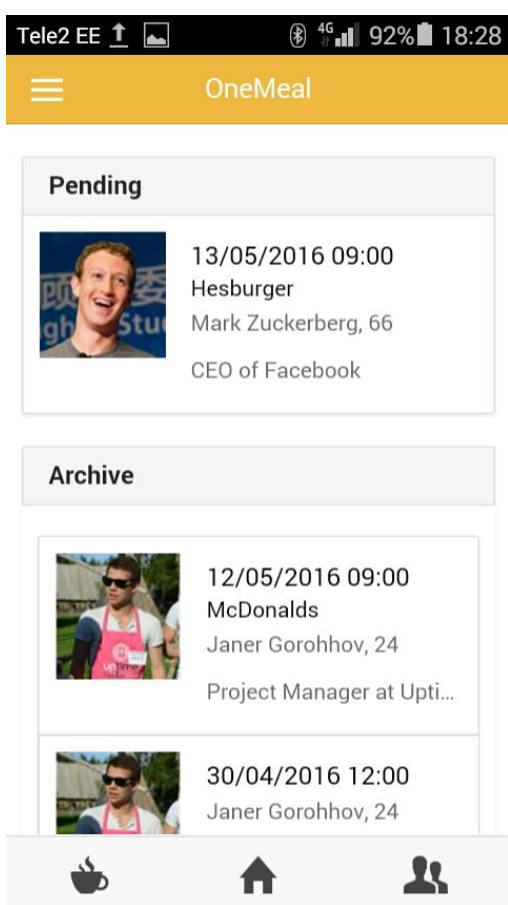
5.5 Aktiivsed kutsed ning lõunate ajalugu

Sisse logitud kasutaja aktiivsed kutsed ning lõunate ajalugu on defineeritud vaadena *history.html* lehel. Andmed saadakse teostades kaks päringut rakenduse veebiteenusele. Rakenduse kontrolleri *MyHistoryController* saadab kaks *get* päringut, millega päritakse kasutajale andmed.

Esimene päring saadetakse *PendingMealsController*le, andes kaasa parameetrina sisse logitud kasutaja identifikaatori. Veebiteenus teostab *LINQ* päringu rakenduse Microsoft Azure'i pilves olevasse andmebaasi. Päringus küsitakse andmebaasilt kõik kirjed, kus lähtekasutajaks on parameetrina kaasa antud kasutaja identifikaator, lõuna aksepteerimise tõeväärtuseks on väär ja lõuna toimumise aeg on tulevikus. Veebiteenus tagastab mobiilirakendusele *JSON*'i kujul kõik päringule vastavad lõunad koos kasutajainfoga.

Rakendus väärtustab vaates olevad muutujad tagastatud päringu andmetega ning kasutajale avaneb joonisel 8 kujutatud pilt.

Teine päring saadetakse ArchiveMealsControllerile, andes kaasa parameetrina sisse logitud kasutaja identifikaatori. Veebiteenus teostab *LINQ* päringu rakenduse Microsoft Azure'i pilves olevasse andmebaasi. Päringus küsitakse andmebaasist kõik aktsepteeritud lõunakuulutused, kus sisse logitud kasutaja on üks lõuna osapooltest. Veebiteenus tagastab mobiilirakendusele *JSON*'i kujul kõik päringule vastavad lõunad koos kasutajainfoga. Rakendus väärtustab vaates olevad muutujad tagastatud päringu andmetega ning kasutajale avaneb joonisel 5 kujutatud pilt.

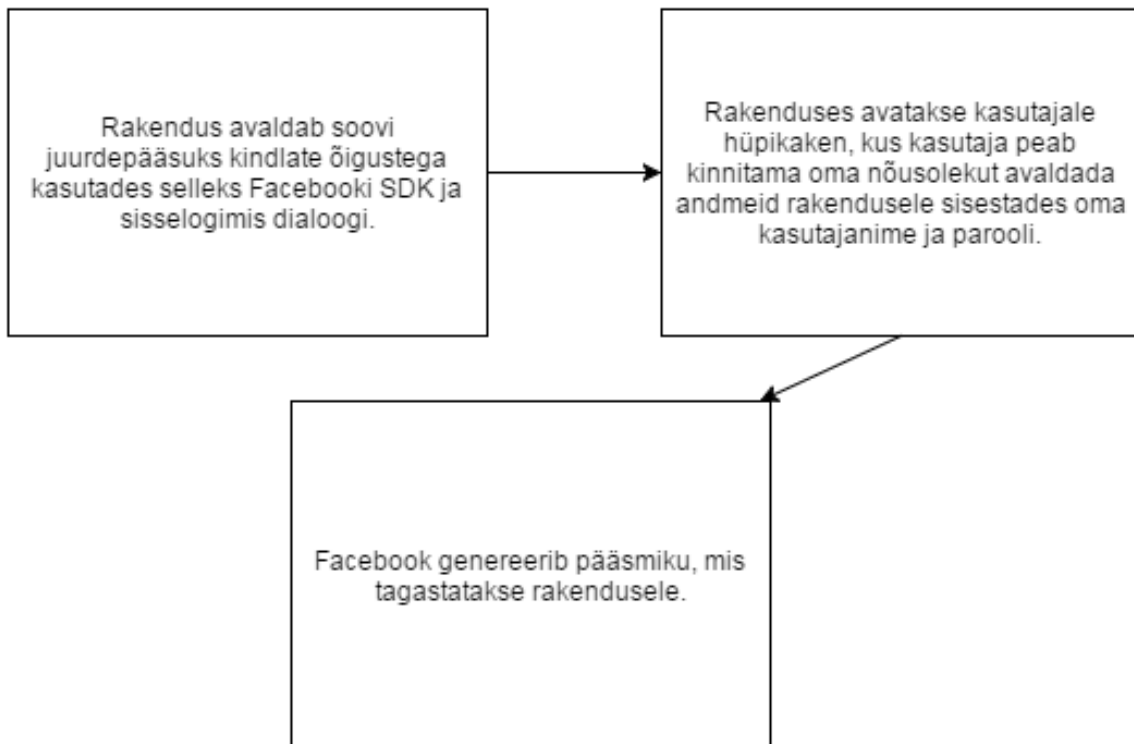


Joonis 8. Vaade aktiivsetest kutsetest ja aktsepteeritud lõunatest

Kasutajal on võimalik ka aktiivseid kutseid vastu võtta. Selleks peab ta vajutama *Pending* kirjele ning kasutajale kuvatakse profiilikuva, kus on võimalik kasutaja lõunasöögikutse aktsepteerida.

5.6 Isikustamine

Rakenduse isikustamine on lahendatud läbi veebibrauseri sessioonivõtmete. Rakenduse avamisel kontrollib süsteem, kas kasutaja veebibrauserisse on kirjutatud lokaalne sessiooni võti ja kasutaja identifikaator, mis saadakse Facebooki veebiteenuselt, kui luuakse viimasega ühendus. Facebooki veebiteenus tagastab esmakordsel edukal ühendumisel pääsmiku, mida süsteem kasutab võtmena, et kontrollida, kas kasutajal on eksisteeriv aktiivne sessioon (vt. joonis 9).



S

Joonis 95. Facebook integratsioon[19]

1). Ühe sessiooni pikkus on Facebooki poolt määratud 60 päeva.

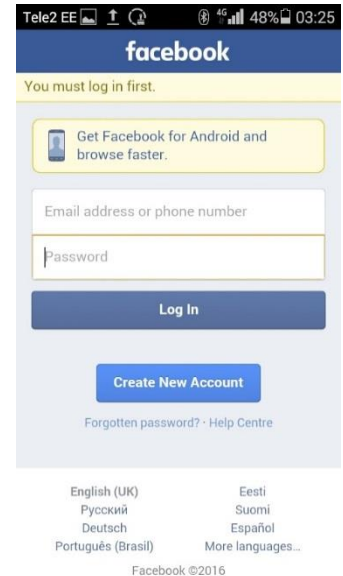


onemeal

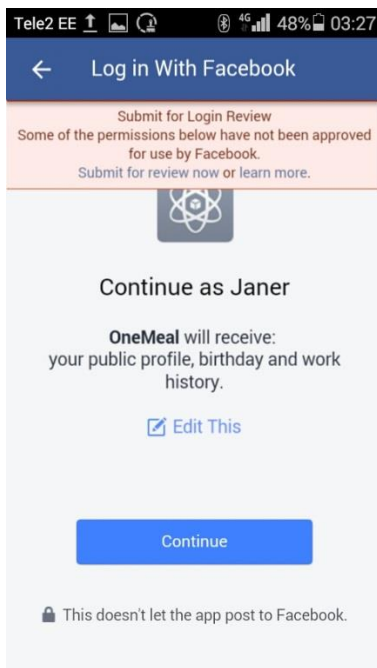
Login

Joonis 10. sisselogimise vaade

Kui kasutajal on juba aktiivne sessioonivõti, suunatakse kasutaja maandumislehele *mainscreen.html*. Vastasel juhul nõutakse kasutajalt, et ta looks uue sessioonivõtme kasutades selleks Facebooki integratsiooni. Uue sessioonivõtme loomine toimub *login.html* vaates (vt joonis 10). Vaates kuvatakse kasutajale rakenduse logo ning nupp, millele vajutades suunatakse Facebooki lehele, kus kuvatakse kasutajale sisselogimiskraan (vt joonis 11). Kasutaja peab sisestama oma Facebooki kasutajanime ja parooli ning seejärel suunatakse kasutaja õiguste kinnitamise vormile. Õiguste kinnitamise vormil peab kasutaja kinnitama, et mobiilirakendus võib pärida tema andmeid läbi Facebooki API (vt joonis 12).



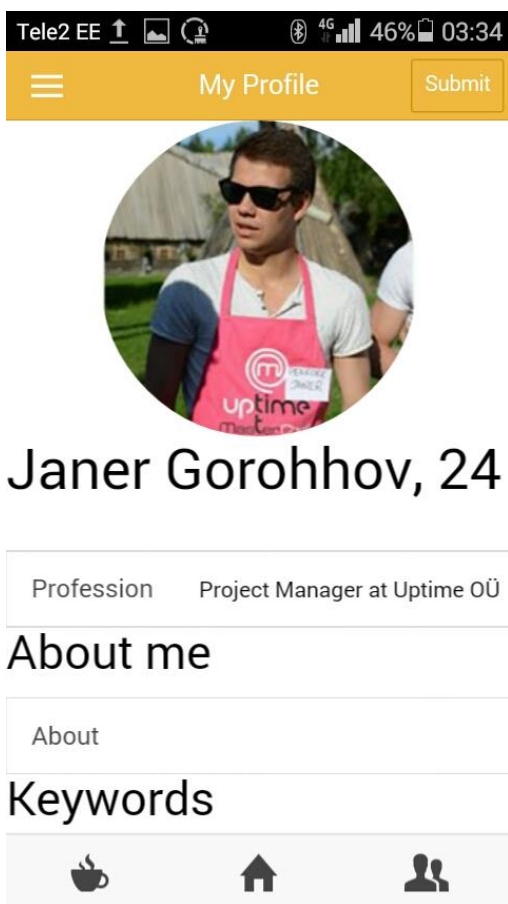
Joonis 6. Facebooki sisselogimiskraan



Joonis 12. Õiguste kinnitamise kuva

5.7 Minu profiil

Kasutajal on võimalik näha oma profiili vajutades menüüs nupule „Profile“. Nupuvajutuse peale suunatakse kasutaja *myprofile.html* vaatesse, kus süsteem teostab *get* päringu rakenduse veebiteenusele ning pärib kasutaja profiiliinfo. Päringu parameetrina antakse kaasa sisselogitud kasutaja identifikaator. Kasutajale kuvatakse tema profiilipilt, nimi, vanus, ametikoht, kirjeldus ning võtmesõnad. Kasutajal on võimalik muuta oma ametikohta, kirjeldust ja võtmesõnu muutes väärtust ja vajutades päises olevale nupule „Submit“. Nupuvajutuse peale tehakse *post* päring rakenduse veebiteenusele, mis uuendab andmebaasis kasutaja profiilirida muudetud andmetega.



Joonis 13. Minu profiili vaade

5.8 Rakenduse kirjeldus

Kasutajal on võimalik tutvuda rakenduse kirjeldusega ning rakenduse looja infoga *about.html* vaates. Vaade on staatiline ning ei päri andmeid veebiteenusest.

5.9 Väljalogimine

Kasutajal on võimalik logida välja rakendusest vajutades menüüs nupule „*Log out*“. Nupuvajutuse peale kustutatakse kasutaja rakenduse sessiooniajalugu ning suunatakse sisselogimise vaatesse.

6. Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli luua mobiilirakenduse prototüüp lõunatajatele, mida saaks kasutada, et üksikud lõunatajad saaksid leida endale kaaslase, kellega koos välja sööma minna. Teiseseks eesmärgiks oli luua prototüüp mobiiliplatvormist sõltumatuna, et seda oleks lihtsasti võimalik arendada edasi ka tulevikus teistele mobiiliplatvormidele. Antud töös kasutati mobiiliarenduse põhitõdesid, kus rakendus peab olema kasutaja jaoks lihtsasti arusaadav ning võimalikult minimaalse kujundusega.

Käesoleva bakalaureusetöö käigus valmis mobiilirakenduse prototüüp, mis võimaldab Facebooki konto olemasolul kutsuda teisi kasutajaid välja lõunale, lisada enda lõunasöögikuulutus ning vastata teiste kasutajate saadetud lõunasöögikutsetele. Kõige olulisema osana äriefunktsionaalsusest antud rakenduse juures realiseeriti mobiili asukohapõhine lõunasöögikuulutuste kuvamine, kus kasutajale kuvatakse ainult lõunasöögikuulutusi, mis asuvad maksimaalselt 30 km kaugusel kasutaja hetkeasukohast. See võimaldab kasutajatel näha enda läheduses olevaid lõunasöögikuulutusi ning filtreerib kutsed, mis kasutajat geograafilistel asukoha tõttu ei huvita. Rakenduse esialgne kujundus tehti primitiivne ning kasutati oranži värvi, mida seostatakse tervisliku toiduga ja toiduisu tekitamisega. Oranži värvi kasutatakse tihti ka toidu promotsiooniks.

Bakalaureusetöös püstitatud eesmärgid said täidetud ning töö kõigis valminud mobiilirakenduse prototüüp võimaldab lõunatajatel leida oma piirkonnast lõunakaaslasi.

Töö autor plaanib jätkata mobiilirakenduse funktsionaalsuse, võimaluste ja disaini edasiarendamist ning kindlasti lisada rakendusele ka järgneva:

- Lõunasöögikuulutustele kahe uue vaate lisamine
 - Kasutaja piirkonnas aktiivsed kuulutused, kus sorteeritakse kuulutused vastavalt ühistele huvidele
 - Kasutaja piirkonnas aktiivsed kuulutused, mis on edastatud kasutaja Facebooki sõprade poolt
- Kokkulepitud lõunasöögi tühistamine
- Lõunasöögikaaslase hindamine
- Teavituste saatmine enne lõuna algust
- Sisselogimine kasutades LinkedIni
- Kokkulepitud lõunasöögikohta jõudmise kohta teavituse saatmine partnerile
- Lõunate otsimine ja filtreerimisvõimalused

- Rakenduse seadistamine (kauguse määramine)

7. Kasutatud materjalid

- [1] Jobvite, "Jobvite job seeker nation study 2014",
<http://web.jobvite.com/rs/jobvite/images/2014%20Job%20Seeker%20Survey.pdf> Viimati külastatud 4. august 2016
- [2] NeverEatAlone kodulehekül, <https://nevereatalone.io/> Viimati külastatud 4. august 2016
- [3] Romain Dillet, "With never eat alone, never eat alone"
<http://techcrunch.com/2015/10/23/with-never-eat-alone-never-eat-alone/> Viimati külastatud 4. august 2016
- [4] LunchBuddy kodulehekül, <http://lunchbuddy.zcoder.io/> Viimati külastatud 4. august 2016
- [5] LunchMatcher kodulehekül, <http://www.lunchmatcher.com/> Viimati külastatud 4. august 2016
- [6] Apache Cordova Documentation,
<https://cordova.apache.org/docs/en/latest/guide/overview/index.html> Viimati külastatud 4. august 2016
- [7] Jen Looper, "What is a WebView?",
<http://developer.telerik.com/featured/what-is-a-webview/> Viimati külastatud 4. august 2016
- [8] Ionic Documentation, <http://ionicframework.com/docs/overview/> Viimati külastatud 4. august 2016
- [9] AngularJS Wikipedia <https://en.wikipedia.org/wiki/AngularJS> Viimati külastatud 4. august 2016
- [10] ASP.NET MVC Wikipedia https://en.wikipedia.org/wiki/ASP.NET_MVC Viimati külastatud 4. august 2016
- [11] Nikhil Kothari, Vandana Datye "Developing Microsoft ASP.NET Server Controls and Components", Microsoft Press, 2012
- [12] Microsoft, Common Language Specification [https://msdn.microsoft.com/en-us/library/12a7a7h3\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/12a7a7h3(v=vs.100).aspx) Viimati külastatud 4. august 2016
- [13] Academic Tutorials <http://www.academictutorials.com/microsoft.net/dotnet-framework.asp> Viimati külastatud 4. august 2016

- [14] Microsoft SQL Server Wikipedia,
https://en.wikipedia.org/wiki/Microsoft_SQL_Server Viimati külastatud 4. august 2016
- [15] Microsoft, LINQ to SQL [https://msdn.microsoft.com/en-us/library/bb386976\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/bb386976(v=vs.110).aspx) Viimati külastatud 4. august 2016
- [16] CSS Wikipedia https://en.wikipedia.org/wiki/Cascading_Style_Sheets
Viimati külastatud 4. august 2016
- [17] HTML5 Wikipedia, <https://en.wikipedia.org/wiki/HTML5> Viimati külastatud 4. august 2016
- [18] Kate Mihalikova, Ionicon DateTime-Picker Repository
<https://github.com/katemihalikova/ion-datetime-picker> Viimati külastatud 4. august 2016
- [19] Facebook, Access Tokens
<https://developers.facebook.com/docs/facebook-login/access-tokens> Viimati külastatud 4. august 2016
- [20] Margaret Rouse, <http://searchcloudcomputing.techtarget.com/definition/Windows-Azure> Viimati külastatud 4. august 2016

Lisad

I. Programmi lähtekood

1. Mobiilirakenduse lähtekoodi repositoorium – viimati uuendatud 12.05.2016.
https://github.com/janergoro/OneMeal_Mobile_Prototype
2. Veebiteenuse lähtekoodi repositoorium – viimati uuendatud 12.05.2016.
https://github.com/janergoro/OneMeal_Api_Prototype

II. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, **Janer Gorohhov**,
(*autori nimi*)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
Platvormist sõltumatu mobiilirakendus lõunatajatele,
(*lõputöö pealkiri*)

mille juhendaja on Pelle Jakovits,
(*juhendaja nimi*)

- 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
- 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **4.08.2016**