



LAUREA
AMMATTIKORKEAKOULU
Yhdessä enemmän

Mobiiliapplikaatioiden kehittäminen Phonegapilla

Stenroos, Tiina

2017 Laurea



Laurea-ammattikorkeakoulu

LAUREA
AMMATTIKORKEAKOULU

Yhdessä enemmän

Mobiiliapplikaatioiden kehittäminen Phonegapilla

Tiina Stenroos
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Kesäkuu, 2017

Tiina Stenroos

Mobiiliapplikaatioiden kehittäminen Phonegapilla

Vuosi 2017

Sivumäärä 26

Tämän opinnäytetyön aiheena on mobiiliapplikaatioiden kehittäminen Phonegapilla. Työ käsittää applikaatioiden kehittämisprosessin määrittelystä julkaisuun ja valottaa Phonegapin hyötyjä kehittämisen työkaluna. Työn tavoitteena oli tuottaa Laurea-ammattikorkeakoulun opiskelijoille suunnattu applikaatio, josta löytyisi toimintoja helpottamaan opiskelijoiden kouluarkea. Applikaation kehittämisprosessi toteutettiin vesiputousmallia ketterästi mukailen. Kehittäminen aloitettiin määrittelyllä, jonka pohjalta suunniteltiin ja toteutettiin applikaatiosta prototyyppi. Toimeksiantajalta saadun palautteen perusteella prototyyppiä kehitettiin haluttuun suuntaan. Saavutettavuuden kannalta oli tärkeää, että applikaatio on saatavilla kaikille käyttöjärjestelmille. Phonegapin hyötynä on sen mahdollisuus tuottaa eri alustoilla toimivia applikaatioita pienillä koodimuutoksilla. Applikaatio toteutettiin ensin Android -käyttöjärjestelmälle ja sitten iOS ja Windows 8 -käyttöjärjestelmille. Tuloksena oli hybridiapplikaatio, jolla voi hakea luokkatiloja, katsoa lukuvuoden aikataulut ja selata yksiköiden ruokalistoja. Toimeksiantaja on hyväksynyt applikaation ja se otetaan käyttöön Laurean opiskelijoille syksyllä 2017.

Tiina Stenroos

Developing mobile applications using Phonegap

Year	2017	Pages	26
------	------	-------	----

The thesis is about mobile application development using Phonegap. The thesis involves the process of developing applications starting from the software requirements specification to release. It highlights the benefits of using Phonegap as a tool for development. The purpose of the thesis was to produce an application for the students at the Laurea University of Applied Sciences. The application includes functions to ease the everyday life of students. The application development process was carried out with an agile adaptation of the waterfall model. The development was started by designing and creating a prototype application in accordance with the software requirements specification. Based on the feedback from the customer, the prototype was developed in the desired direction. For accessibility it was important that the application is available for all most common operating systems. Phonegap's benefit is its ability to produce applications on different platforms with small code changes. The application was first developed for Android operating system and then for iOS and Windows 8. The result was a hybrid application that allows the user to search for classrooms, look at the school's timetables and browse the menu of the units. The client has approved the application and it will be introduced to Laurea students in the autumn 2017.

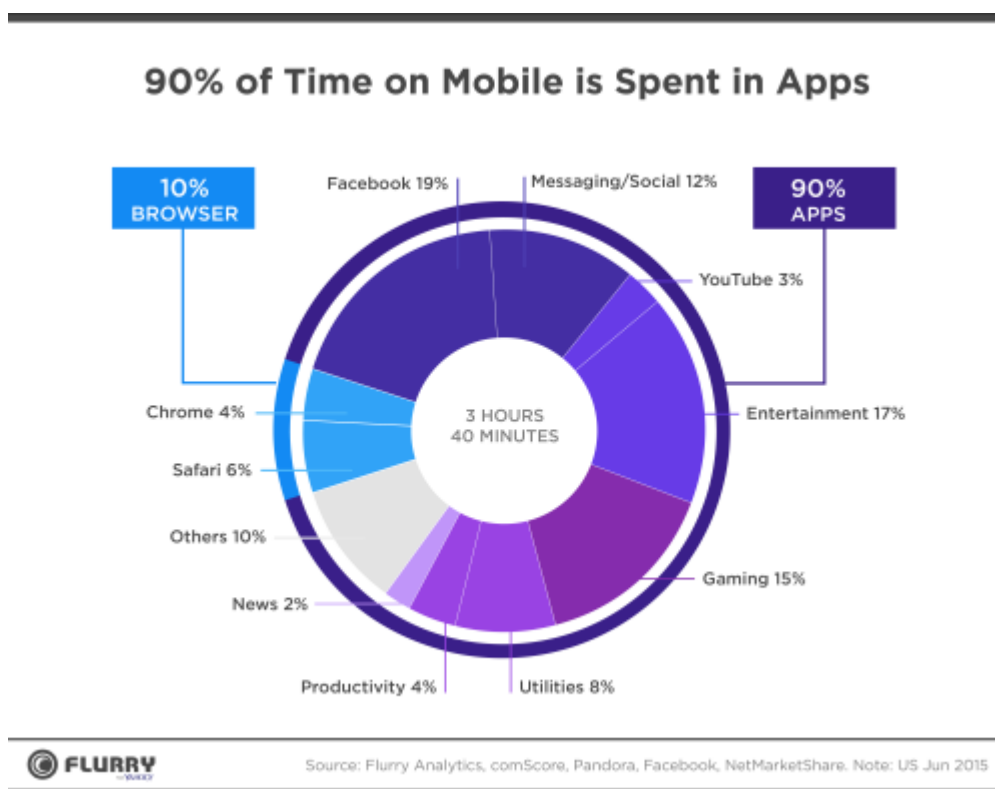
Keywords: Phonegap, Mobile application, Software engineering

Sisällys

1	Johdanto	6
2	Tavoite ja menetelmät.....	7
3	Käsitteet	8
4	Phonegap	9
	4.1 Vahvuudet ja haasteet kehittämisessä	9
	4.2 Phonegapin käyttöönotto	11
5	Applikaation kehittäminen	13
	5.1 Määrittely.....	13
	5.2 Suunnittelu	14
	5.3 Toteutus	16
	5.4 Testaus	20
	5.5 Julkaisu	21
	5.6 Päivittäminen	22
6	Yhteenveto ja loppupäätelmät	23
	Lähteet	24
	Kuviot..	25
	Taulukot	26

1 Johdanto

Älypuhelimien hyötynä on se, että se kulkee helposti käyttäjänsä mukana pienen kokonsa ansiosta, mutta kykenee kuitenkin periaatteessa samaan kuin pöytätietokone. Älypuhelimien käytön yleistymisen myötä mobiiliapplikaatioista on tullut jokapäiväinen ja monikäyttöinen apuväline. Flurry Analyticsin tekemän tutkimuksen mukaan mobiilikäyttäjiä on nyt enemmän kuin työpöytäkäyttäjiä ja suurin osa käyttöajasta kuluu applikaatioiden käyttämiseen (Kuva1).



Kuva 1: Flurry Analytics

Monella suurella amerikkalaisella yliopistolla, kuten Harvard ja Brown, on käytössä oma applikaationsa. Suomessa koulut eivät ole vielä ottaneet tällaista applikaatiota käyttöönsä. Applikaatio on kuitenkin nopeampi käyttää, kuin web-sivu, vaikka sivut olisi mobiilioptimoitu. Tämä johtuu siitä että applikaatio ladataan vain kerran ja internetyhteyttä käytetään vain kun halutaan lukea tietoa verkkosivuilta, joten käyttäjän ei tarvitse odottaa kun selain lataa kuvia ja sivun muotoiluja.

Saavuttaakseen mahdollisimman monet käyttäjät, palvelun pitää toimia yleisimmillä laitteilla. Mobiililaitemarkkinoilla on kuitenkin kova kilpailu ja erilaisia käyttöjärjestelmiä ja -versioita on useita. Tämä tuottaa ongelmia kehittäjälle, koska eri alustoille kehitetyt applikaatiot eivät ole keskenään yhteensopivia ja applikaatio täytyykin tehdä jokaiselle

järjestelmälle erikseen eri ohjelmointikielellä. Phonegap mahdollistaa hybridiapplikaatioiden tekemisen tavallisilla webkielillä: HTML, CSS ja javascript. HTML5 pohjaisen applikaation hyötynä on se, että koodia ei tarvitse kirjoittaa kuin kerran, jonka jälkeen se voidaan paketoita sopivaksi yleisimmille alustoille.

Työn toimeksiantaja on Laurea- ammattikorkeakoulu. Koulu on monialainen ja jakaantuu seitsemään yksikköön. Opiskelijoita Laurea-ammattikorkeakoulussa on noin 7800. Laureasta lähtee ulkomaille opiskelijavaihtoon noin 460 opiskelijaa vuosittain ja vastaavasti n. 400 ulkomaista opiskelijaa tulee Laureaan. (Kansainvälisyys opinnoissa.) Laurealla on käytössään mobiilioptimoidut websivut. Jos opiskelija kuitenkin haluaisi katsoa puhelimellaan esim. mitä ruokaa koulussa on tarjolla kyseisenä päivänä, tulisi ensin avata puhelimesta selain, hakea Laurean kotisivut, avata valikosta palvelut, hakea palveluiden alta opiskelijaravintolat ja valita sitten oman yksikkönsä ravintola. Tämä hakuprosessi käsittää viisi vaihetta, joista jokaiseen kuluu hakemisen lisäksi sivujen muotoilujen ja tiedon prosessoimiseen kuluva aika. Jos käyttäjä ei ole vierailut Laurean sivuilla ennen, voi tiedon etsimiseen kulua vielä enemmän aikaa.

2 Tavoite ja menetelmät

Opinnäytetyö on toiminnallinen suunnittelu- ja kehitystyö Laurea-ammattikorkeakoululle. Tavoitteena on kehittää koulun opiskelijoille applikaatio, joka sisältää opiskeljan arkea helpottavia toimintoja. Kehittämisessä käytetään Adobe Phonegapia, joka mahdollistaa applikaatioiden tekemisen eri käyttöjärjestelmille uudelleenkäyttämällä samaa koodia. Kehittäminen tapahtuu ketterästi vesiputousmallia mukailien. Kehittäminen aloitetaan määrittelyllä toimeksiantajan kanssa, josta siirrytään suunnitteluun. Applikaatiosta tehdään evoluutioprototyyppi, jota kehitetään haluttuun suuntaan, kunnes applikaatio on valmis. Testausta on tarkoitus tehdä koko kehitysprosessin ajan. Phonegap tarjoaa developer tool-ohjelman jolla käyttöliittymää voi testata mobiililaitteella ja nähdä muutokset reaaliajassa.

Työn tietoperustana käytetään pääosin Phonegapin virallisilta sivuilta löytyvää dokumentaatiota, jotka esittelevät sen käytön kaikilla tarjotuilla alustoilla ja antavat hyvän alun sovelluskehitykseen. Dokumentaatio sisältää ensimmäisen sovelluksen luomisen, kaikkien tarvittavien Cordova -työkalujen asentamisen, laajennukset, asetustiedostojen editoinnin, erityisten asioiden huomioon ottamisen kun applikaatiota kehitetään kaikille tarjotuille alustoille: Android, Blackberry 10, iOS, OS X, Ubuntu ja Windows.

Toimeksiantajan vaatimus oli, että applikaation täytyy olla saatavilla kaikille Laurean opiskelijoille ja sisältää samat ominaisuudet yksiköstä riippumatta, mikäli tiedot on päivitetty internetsivulle. Applikaation kehittämisessä on kiinnitetty huomiota tarkasti saavutettavuuden aspektiin ja että se olisi helppokäyttöinen kaikkien alojen opiskelijoille. Tämän vuoksi on

tärkeää, että applikaatio toimii kaikilla yleisimmillä käyttöjärjestelmillä: Android, iOS ja Windows. Applikaatio toimii kaikilla mobiililaitteilla, mutta se optimoidaan pääasiassa puhelimelle, joten käyttö tabletilla rajataan tässä työssä pois. Koska Laureassa on paljon kansainvälisiä opiskelijoita, olisi hyvä saada applikaatio myös englannin kieliseksi. Tämä ei kuitenkaan ollut vaatimuksena. Applikaation päätoiminnot rajattiin kolmeen päätoimintoon: tilahaku, lukuvuoden aikataulut ja ruokalistat.

Applikaatio tehdään nykyisten järjestelmien pohjalta. Pääosin applikaatio hakee tarvittavat tiedot Laurean websivuilta, mutta tilahaussa käytetään Asion -järjestelmää. Asio -järjestelmästä löytyvät opetuksen aikataulut, opintotarjonta ja tilavaraukset. Muita Laurean järjestelmiä ovat sähköinen oppimisympäristö Optima, Laurean intra Link ja SoleOPS opintojen suunnitteluun, seurantaan ja ilmoittautumiseen (Laurean järjestelmät.) Laurealla on suunnitelmissa korvata vanhat järjestelmät ja siirtyä uuteen Peppi -järjestelmään vuoden 2017 lopussa. Applikaatio voidaan päivittää helposti uuteen järjestelmään käyttämällä ohjelmarajapintoja. Peppi tarjoaa ohjelmarajapintojen käyttöön dokumentteja kehittämistä varten.

3 Käsitteet

Plugin

Plugin on ohjelmiston osa, joka lisää tietyn ominaisuuden olemassa olevaan ohjelmaan. (Cordova Plugins 2015)

API

Application programming interface (API), suomeksi ohjelmarajapinta, on järjestelmien välisen viestinnän käyttöliittymä. Se esimerkiksi voi mahdollistaa pääsyn käyttöjärjestelmän natiiviominaisuuksiin käyttämällä Javascriptia. Javascript kutsuu API:a ja palauttaa vastauksena saadun JSON-tiedoston. (Adobe Phonegap 2016)

Javascript

Javascript on web-ympäristössä käytettävä dynaaminen, oliopohjainen komentosarjakieli, jota käytetään lisäämään web-sivuille toiminnallisuutta. (W3S 2017)

SQLite

SQLite on kevyt avoimen lähdekoodin relaatiotietokantajärjestelmä, joka on upotettuna kaikkiin Android-laitteisiin. Se toimii myös iOS:illa, mutta Windows käyttöjärjestelmälle ei löydy enää tukea. (SQLite 2016)

CSS

Cascading Style Sheets (CSS) on muotoilukieli jolla muotoillaan www -sivujen ulkonäköä. CSS:ää käytetään useimmiten HTML:n ja XHTML:n kanssa. (W3S 2017)

HTML5

Hypertext Markup Language (HTML) on standardoitu www -sivujen muotoilukieli ja HTML5 on uusin versio tästä merkintäkielestä. (W3S 2017)

4 Phoneygap

Phoneygap on ilmainen, avoimen lähdekoodin työkalu hybridiapplikaatioiden kehittämiseen. Phoneygapin on alunperin kehittänyt Nitobi, mutta nyt se on Adoben omistuksessa. Phoneygap tarjoaa graafisen käyttöliittymän Cordova -ohjelmointiympäristöön erityisesti sellaisille ohjelmoijille, jotka eivät ole edistyneitä komentorivin käyttämisessä ja haluavat graafisen käyttöjärjestelmän projektien tekemiselle. Ohjelmointi Phoneygapilla tapahtuu käyttäen yleisimpiä webkieliä: HTML, CSS ja Javascript käyttäjän omassa kehitysympäristössä. (McWheter 2012)

4.1 Vahvuudet ja haasteet kehittämisessä

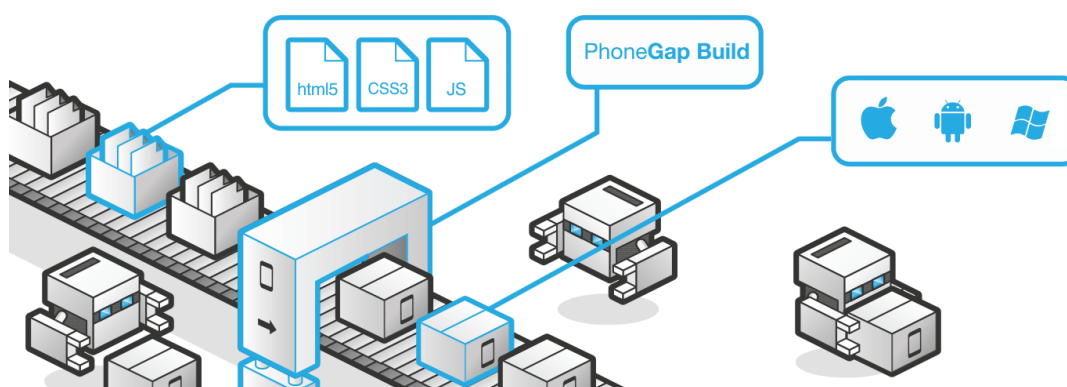
Applikaatioita voi tehdä kahdella eri menetelmällä. Natiivisti tai hybridiapplikaationa, jolloin applikaatio on web-sivujen ja natiivin sovelluksen välimuoto. Natiivisovellus tarkoittaa tietylle käyttöjärjestelmälle erikseen ohjelmoitua sovellusta. Natiivisovellusten kehittäminen mahdollistaa sen, että sovelluksen suorituskyky on paras mahdollinen, kun siltä vaaditaan suurta laskentakykyä. Natiivisovellukset pääsevät myös käsiksi laitteiden ominaisuuksiin ja rajapintoihin toisin kuin web-sivut. Eri alustoille ohjelmoitua applikaatioita eivät ole kuitenkaan keskenään yhteensopivia, joten on mietittävä mille käyttöjärjestelmälle sovellusta aletaan kehittää. (Kolme tapaa kehittää mobiilisovellus 2014) Android johtaa markkinoita 83% osuudella vuonna 2016 (Smartphone market 2016), mutta sovelluksen kehittäminen vain yhdelle käyttöjärjestelmälle sulkee kuitenkin suuren määrän käyttäjiä ulkopuolelle.

Mobile OS	Operating System	Software/IDEs	Programming Language
iOS	Mac only	Xcode	Objective C
Android	Windows/Mac/Linux	Eclipse/Java/Android Development Tool (ADT)	Java
BlackBerry	Windows mainly	Eclipse/JDE, Java	Java
Symbian	Windows/Mac/Linux	Carbide.c++	C++
WebOS	Windows/Mac/Linux	Eclipse/WebOS plugin	HTML/JavaScript/C++
Windows 7 Phone	Windows mainly	Visual Studio 2010	C#, .NET, Silverlight or WPF

Taulukko 1: Kehittämiskielet ja alustat (Ghatol 2012)

Taulukosta 1 voi havaita, että tehdäkseen applikaation, joka toimii sekä Androidilla ja iOS:illa, on ohjelmoijan osattava kieliä: Objective-C, C ja Java ja käytettävänä kehitysalustana Xcodea ja Eclipseä (Ghatol 2012). Sovellus on ohjelmitava alusta loppuun jokaiselle järjestelmälle erikseen. Tämä prosessi on kallista ja aikaa vievää. Jos haluttu sovellus on pääosin tietopohjainen, on hybridiapplikaatio usein parempi vaihtoehto.

Hybridiapplikaatio on HTML5- tekniikoilla toteutettu web-sovellus, joka ajetaan laitteissa omana erillisenä sovelluksenaan. Se yhdistää natiivin sovelluksen ja web-sivujen parhaat puolet. Hybridiapplikaatioiden tekemiseen voidaan käyttää Phonegapia. Natiivisovelluksista eroten hybridiapplikaatio tarvitsee ohjelmoida vain kerran, jonka jälkeen se paketoidaan sopivaksi kaikille käyttöjärjestelmille. Phonegap tarjoaa tähän pilvessä toimivan Phonegap buildin (kuva 3).



Kuva 2: Phonegap Build

Phonegapin käyttö on ilmaista ja sovellusten luominen tapahtuu ohjelmoijan omassa kehitysympäristössä. Jos sovelluksia halutaan kuitenkin julkaista yksityiseen käyttöön enemmän kuin yksi, veloittaa phonegap 9,99 dollaria kuukaudessa. Tällä hinnalla saa julkaista 25 sovellusta. (Phonegap Build 2013) Kilpailijalla Apperylla 3 applikaatiota saa 90 dollarin kuukausihintaan. (Appery 2017) Suurimpana hyötynä phonegapissa voidaan pitää sitä, että se on alustariippumaton ja mahdollistaa näin applikaatioiden kehittämisen pienellä työllä eri käyttäliittymille käyttämällä suurta osaa samasta koodista. Tämä on nopeaa ja kustannustehokasta verrattuna applikaatioiden kehittämiseen natiivisti.

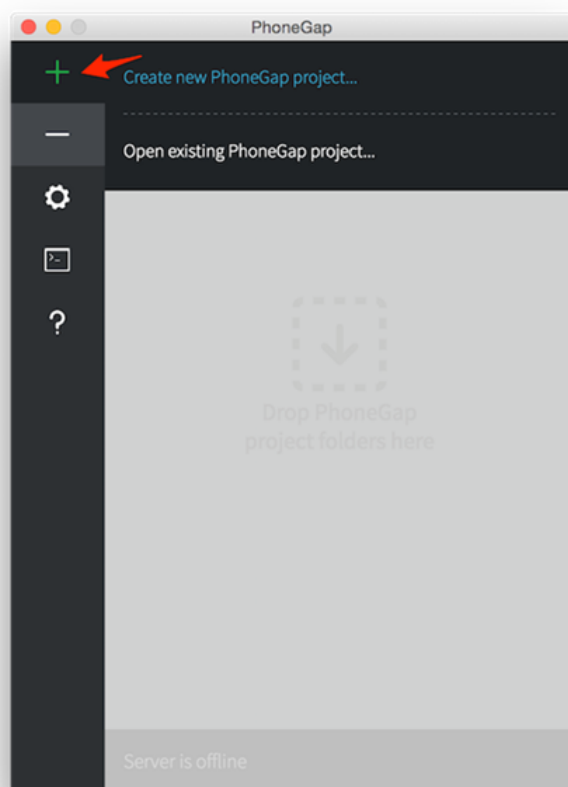
Hybridisovellukset voivat hyödyntää laitteiden toimintoja natiivisovellusten tapaan käyttäen javascriptiä. Phonegap tarjoaa kehittäjälle pohjia erilaisia sovelluksia varten esim. Push notification -mallin. Phonegap työpöytäsovellus sisältää myös valmiiksi koodia kosketusnäyttöä ja näytön kääntymisen hyödyntämistä varten. Graafinen käyttäliittymä ja dokumentaatio tekevät Phonegapin käytöstä helppoa, mutta se tarjoaa kuitenkin täyden applikaation kustomoimisen alusta loppuun ohjelmoijalle, toisin kuin ”drag and drop” tyyppiset sovelluskehitysympäristöt.

Eri valmistajien puhelimet eivät ole identtisiä, eivätkä kaikki mallit tue kaikkia natiiviominaisuuksia, joten sovelluksen toiminnassa voi olla eroja riippuen, mikä puhelinmalli käyttäjällä on. (Saman koodin toimiminen eri käyttäliittymillä vaatii myös välttämättömästi muutosten tekemistä ja joissain tapauksissa voi olla hankalaa identifioida, mitä muutoksia koodiin tulisi tehdä, jotta sovelluksen kaikki funktiot toimisivat samalla tavalla eri käyttäliittymien kesken.) Webkielillä tuotettu koodi ei ole myöskään yhtä tehokasta kuin natiivikielillä tuotettu, joten esim. Pelien tekemiseen Phonegap ei sovellu.

4.2 Phonegapin käyttöönotto

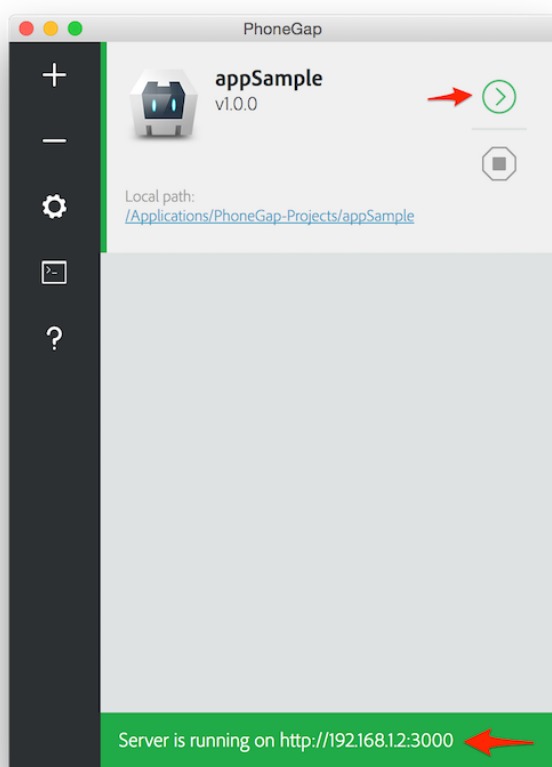
Phonegap tarjoaa selkeät käyttöohjeet ja dokumentaatioita kehitystyön aloittamiseksi. Sovelluskehitys aloitetaan asentamalla Phonegap työpöytäsovellus tietokoneelle Phonegapin omilta virallisilta sivuilta. Tämän jälkeen on hyvä ladata Phonegap kehittäjä -applikaatio puhelimeen applikaation testaamista varten mobiililaitteella. Applikaation voi ladata puhelimen omasta sovelluskaupasta ilmaiseksi. (Getting started 2015)

Applikaation tekeminen aloitetaan luomalla uusi projekti työpöytäsovelluksessa. Tämä tapahtuu klikkaamalla plus -ikonia sovelluksen vasemmassa yläkulmassa. (kuva 4) Tämän jälkeen käyttäjä voi valita erilaisia mallipohjia tai aloittaa täysin tyhjän projektin. Suositeltavaa on, ainakin aloittelijalle, valita ”hello world” - mallipohja. Seuraavaksi projektille valitaan tallennuspolku ja sille annetaan nimi, jonka jälkeen projekti luodaan.



Kuva 3: Phonegap työpöytäsovellus

Phonegap luo projektille kansion valittuun polkuun ja tekee käyttäjälle valmiiksi index.html-tiedoston, johon applikaation koodi kirjoitetaan. Jos käyttäjä valitsee projektille mallipohjan, on kansiossa lisäksi valmiina pluginit ja metatietoa, kuten applikaation skaalautuvuus laitteen mukaan. Ohjelmoija voi seurata applikaation kehittymistä selaimessa ja mobiililaitteessa. Kun projekti luodaan, Phonegap aloittaa pienen webserverin projektille. Serverin osoite näkyy työpöytäsovelluksen alareunassa vihreällä taustalla (kuva 5). Osoite syötetään Phonegap developer -applikaation puhelimesta tai selaimen osoitekenttään.



Kuva 4: Phonegap projekti

5 Applikaation kehittäminen

Mobiiliapplikaatioiden kehittäminen Phonegapilla muistuttaa web-sivujen kehitysprosessia. Applikaation kehittämisessä on noudatettu, ketterästi mukailen, perinteistä vesiputousmallia: määrittele, suunnittele, toteuta, testaa, ja ota käyttöön. Osana mallia on iterointi, joka tarkoittaa sitä, että työvaiheisiin voi palata. Iterointi on tärkeä ominaisuus mallissa ja tekee siitä sopivan moneen tilanteeseen. Uudemmissa projektimalleissa, vesiputousmalli on pohjana tavalla tai toisella esim. scrum-malli on ikäänkuin sarja yksittäisiä vesiputouksia. (Haikala 2011)

5.1 Määrittely

Sovellusta rakennettaessa on keskeistä aluksi kartoittaa millaisia toimintoja ja ominaisuuksia vaaditaan. Tällaista kartoitusta kutsutaan vaatimusmäärittelyksi. Vaatimusten määrittely muodostaa ohjelmistoprojektin kulmakiven, jonka varaan projektin kaikki myöhempi menestys rakentuu. Vaatimusten pohjalta lasketaan projektin työmäärä- ja kustannusarviot, laaditaan suunnitelmat ja resursoidaan ja aikataulutetaan hanke. Lopullisen ohjelmiston onnistuneisuutta arvioidaan vertaamalla sitä vaatimusmäärittelyyn. Vaatimusten määrittelyn tuotoksena syntyy vaatimusmäärittelydokumentaatio. vaatimusmäärittelyssä voidaan käyttää

esimerkiksi käyttötapauskaaviota kuvaamaan eri toimijoiden rooleja ja tietokannan käyttötappauksia. Silloin myös muiden, kuin ohjelman toteuttajien on helpompi ymmärtää, mitä ollaan tekemässä. (Haikala 2011)

Aluksi oli mietittävä, mitä olisivat ne opiskelijoille oleelliset toiminnot, joita applikaatioon sisällytettäisiin. Pidin itse tärkeänä tilojen löytämistä helposti sekä ruokalistojen ja aikataulujen selaamista. Keskustelin applikaation toiminnosta Laurean kehityspäällikön kanssa, joka piti näitä toimintoja hyvinä. Olin suunnittelut tekeväni applikaation Android-järjestelmälle, koska se on yleisin käyttäjärjestelmä. Laurean henkilökunta on kuitenkin siirtymässä windows ekosysteemistä Appleen, joten oli tärkeää, että applikaatio toimisi myös iOS:illa. Toimeksiantaja painotti, että applikaation tulee toimia samoin kaikkien yksikköjen kesken. Vaatimusmäärittelyssä keskityttiin pääosin toiminnallisten vaatimusten määrittelyyn. Laadullisen määrittelyn päätavoitteena oli applikaation ulkoasun selkeys ja yhteensopivuus Laurean muun visuaalisen ilmeen kanssa. Alla olevasta taulukossa 2 on listattu applikaation tärkeimmät toiminnalliset vaatimukset.

Tunnus	Vaatumuksen nimi
V1	Kielenvalinta
V2	Navigointi
V3	Tilojen hakeminen
V4	Lukuvuoden aikataulut
V5	Ruokalistat
V6	Toiminta kaikilla järjestelmillä

Taulukko 2: Vaatimukset

V1: Applikaatiossa on mahdollisuus valita kieleksi englanti tai suomi

V2: Käyttäjä pääsee jokaiselta sivulta pääsivulle

V3: Käyttäjä voi hakea, missä luokassa tilaisuus pidetään

V4: Käyttäjä voi nähdä tärkeimmät lukuvuoden päivämäärät

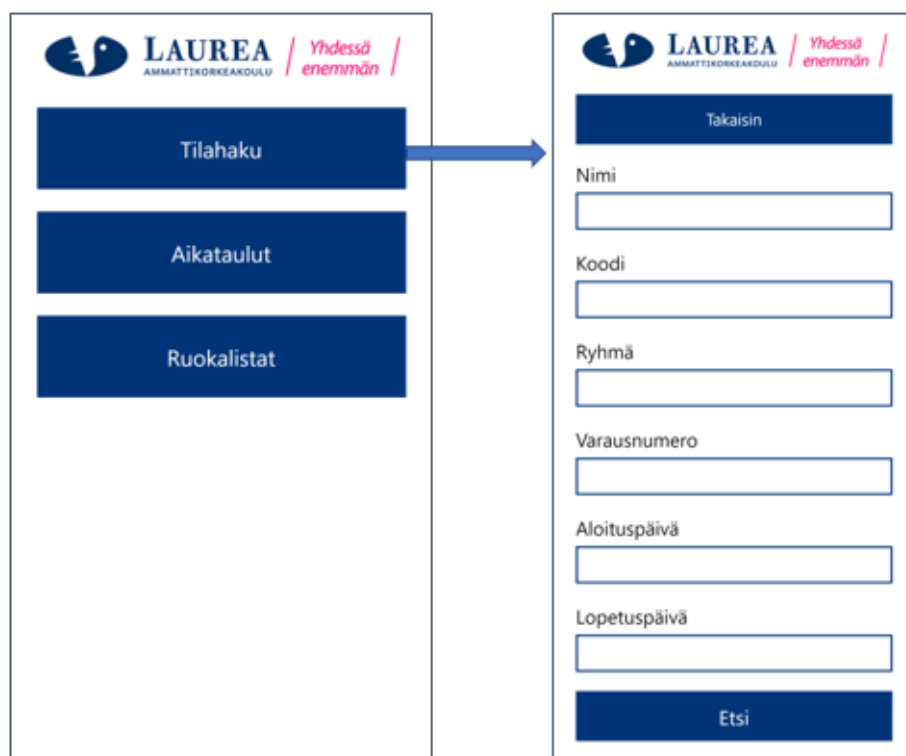
V5: Käyttäjä voi tarkastella jokaisen yksikön ruokalistoja

V6: Applikaatio toimii kaikilla yleisimmällä käyttöjärjestelmillä

5.2 Suunnittelu

Suunnittelu aloitettiin tekemällä projektisuunnitelma. Vaatimusmäärittelyn perusteella mietittiin mitä toimintoja applikaatiossa tulisi olla. Applikaation pitämiseksi yksinkertaisena toiminnot rajattiin kolmeen päätoimintoon. Tarkoituksena oli, että sovelluksen ulkoasu on mahdollisimman selkeä ja yhteneväinen Laurean muiden palveluiden kanssa. Laurean ulkoasuun liittyvät ohjeet on kuvattu tarkasti brändikirjaan. Logon tulee näkyä selvästi kaikissa Laurean julkaisuissa. Applikaatiossa logo on ylätunnisteessa ja näkyy jokaisella

sivulla. Valikkonäppäinten väriksi valittiin sama sininen väri, joka logossa on. Tausta haluttiin pitää valkoisena selkeyden vuoksi. Tämän suunnittelun pohjalta tehtiin rautalankamalli. Mallintamiseen käytettiin Adobe experience design CC- ohjelmaa, joka mahdollistaa käyttöliittymäprototyyppien tekemisen lisäämällä toiminnallisuutta suoraan rautalankamalliin. Ohjelma on ladattavissa ilmaiseksi windows 10- käyttöjärjestelmälle.



Kuva 5: Rautalankamalli

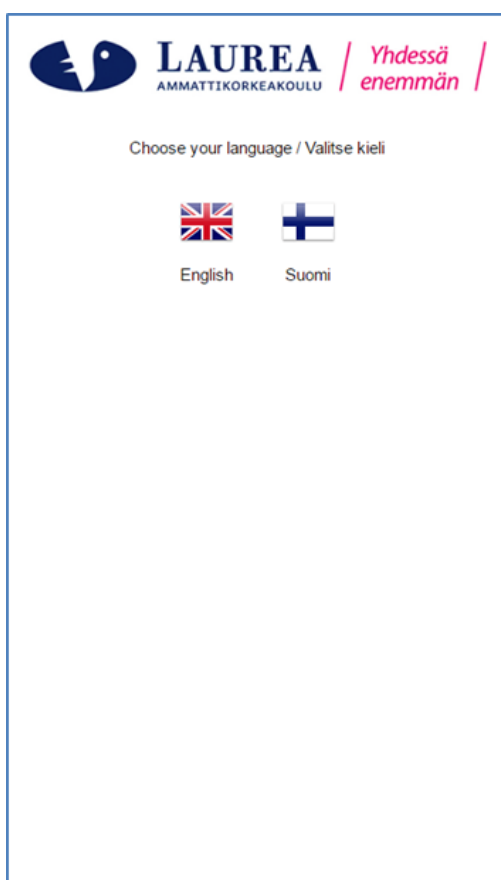
Tämän jälkeen ohjelmoitiin rautalankamallin mukainen prototyyppi (kuva 6). Prototyyppejä voidaan rakentaa kahden päätyypin mukaisesti: evoluutioprototyyppinä tai Kertakäyttöisenä prototyyppinä. Kertakäyttöinen prototyyppi on pelkkä mallinnus järjestelmästä, jonka jälkeen kehitys aloitetaan alusta. Se ei ole sidottu mihinkään malliin ja sen voi tehdä kuka vain missä vaiheessa projektia tahansa. Evoluutio prototyyppissä tuotetta kehitetään vaihe vaiheelta, kunnes se on valmis. Käytännössä tämä voi tarkoittaa esim. että ohjelmasta kehitetään ensin näkyvimät piirteet ja kehitystä jatketaan asiakkaalta saadun palautteen mukaisesti. Jossain vaiheessa sekä asiakas, että ohjelmoija ovat yhtä mieltä siitä, että ohjelma on tarpeeksi hyvä ja se hiotaan valmiiksi ja julkaistaan. (Haikala 2011)

Prototyyppi esiteltiin toimeksiantajalle ja siitä saatiin palautetta ja kehitysehdotuksia, jonka pohjalta prototyyppiä alettiin kehittämään suunnittelun mukaiseksi. Aikataulut vaihdettiin

lukuvuoden aikatauluksi lopullisessa versiossa, jotta käyttäjä ei sekoita sivua esim. Joukko-liikenneaikatauluihin. Tilahakuun toivottiin myös mahdollisuutta saada monipuolisempi haku.

5.3 Toteutus

Laureassa on paljon kansainvälisiä opiskelijoita ja tämä haluttiin ottaa applikaatiossa huomioon. Applikaatio aukeaa puhelimesta kielenvalintaan, josta käyttäjä valitsee joko englannin- tai suomenkielen (kuva 7). Valintaa ei tarvitse tehdä ensimmäisen käyttökerran jälkeen uudestaan, sillä valinta tallentuu Androidissa ja iOS- käyttöjärjestelmässä SQL - liteen. Windows- puhelinta käytettäessä valinta tallentuu puhelimen local storageen.



Kuva 6: Kielenvalinta

Kielenvalinnan jälkeen käyttäjä siirtyy päävalikkoon (kuva 8), josta löytyy kolme päätoimintoa: tilahaku, lukuvuoden aikataulut ja ruokalistat. Tämän lisäksi etusivulla on "muuta kieli"- näppäin, josta painamalla, käyttäjä pääsee takaisin kielivalikkoon. Jokaisella sivulla on myös takaisin -näppäin, joka vie käyttäjän päävalikkoon. Siirtyminen takaisin edelliselle sivulle toimii kuitenkin myös puhelimen omalla navigoinilla. Kun käyttäjä painaa päävalikossa mitä tahansa valikon näppäintä, applikaatio piilottaa päävalikon näppäimet

lisäämällä niihin CSS:ssä “display:none” komennon. Tämän jälkeen applikaatio näyttää kyseisen valikon muotoilut.



Kuva 7: Etusivu

Tilahaussa käyttäjä voi löytää nopeasti, missä luokkatilassa opintojakso pidetään. Tilahaku sivulla käyttäjä näkee lomakkeen, joka vastaa asiosta löytyvää tilahakulomaketta, mutta on tehty sopivammaksi mobiililaitteelle muokkaamalla ulkoasua ja poistamalla kenttiä. (Kuva 9). Painamalla etsi -näppäintä, applikaatio lähettää pyynnön Asioon täytetyn hakulomakkeen tiedoilla, jonka jälkeen applikaatio käsittelee vastauksena saadun lähdekoodin. Tämä lähdekoodi iteroidaan ja jokaiselle tulokselle lisätään oma laatikkonsa. Applikaation suorittaessa funktiota, käyttäjälle näytetään latausanimaatio. Tarkennettu haku-näppäin vie käyttäjän tilat.laurea.fi -sivulle, josta löytyy kokonainen tilahaku -lomake.

https://tilat.laurea.fi/asio/kale

3

LAUREA AMMATTIKORKEAKOULU | Yhdessä enemmän |

Takaisin

Nimi
Professional Communication in English

Koodi
R0023

Ryhmä
HLA1611

Varausnumero
336415

Aloituspäivä
02.10.2016

Lopetuspäivä
15.10.2016

Etsi

Avaa tarkennettu haku

Varauksen haku

Varausnumero

Päivä

Muutoksen raportointi

Tilaisuuden/kurssin nimi

Kurssin tulos

Päätyhtymä

Ryhmä/Ohjelmia

Aika

Vapaustyyppi

Käyttötyyppi

Varsija

Vahvistusta

Vapausnumerot

Vapausnumerot

Ilmoitetaan

• Vapausnumerolla haettaessa hakua ei rajoiteta pöytämuotoon
 • Muiden päivinä voi käyttää vapaan haku tyylinä ajankohde
 • Ensimmäisen päivämäärän voi valita myös vapausnumerosta kalenterista

Kuva 8: Tilahaku

Lukuvuoden aikataulut sivulla (kuva 10), käyttäjä näkee lukuvuodelle tärkeitä päivämääriä, kuten lukuvuoden alkamis- ja päättymisajat ja uusintatilaisuuteen ilmoittautumisen ajankohdat. Sivulla on kaksi haitarivalikkoa, jotta sivulla oleva tieto olisi luettavammassa muodossa. Kun käyttäjä avaa sivun, applikaatio lähettää pyynnön laurean websivulle ja prosessoi vastauksena saadun lähdekoodin. Applikaatio muuttaa lähdekoodista fonttien koon, lisää haitarivalikot ja vaihtaa fonttien väriä. Tämä tekee sivusta sopivan mobiililaitteelle.

Lukuvuoden aikataulu

Lukuvuoden aikataulusta löydät

- lukuvuoden alkamis- ja päättymispäivämäärän,
- jälki-ilmoittautumisajankohdat,
- kesäopinnot ilmoittautumisaikataulun,
- välitarjottimen opintojen ilmoittautumisaikataulun,
- uusintatilaisuudet ja niihin ilmoittautumisen aikataulun,
- valmistumiseen liittyviä aikatauluja, kuten tutkintotodistushakemuksen jättämispäivämäärät ja valmistumisjuhlien päivämäärät.

⊖ Lukuvuosi 2016 - 2017

Kevätlukukausi 2017

9.1.2017 - 16.6.2017

Kevätlukukauden 2017 läsnäolotiedon muutos
1.12.2016 - 31.12.2016

Opintojaksotarjonnan aikataulu

Kevätlukukausi 2017

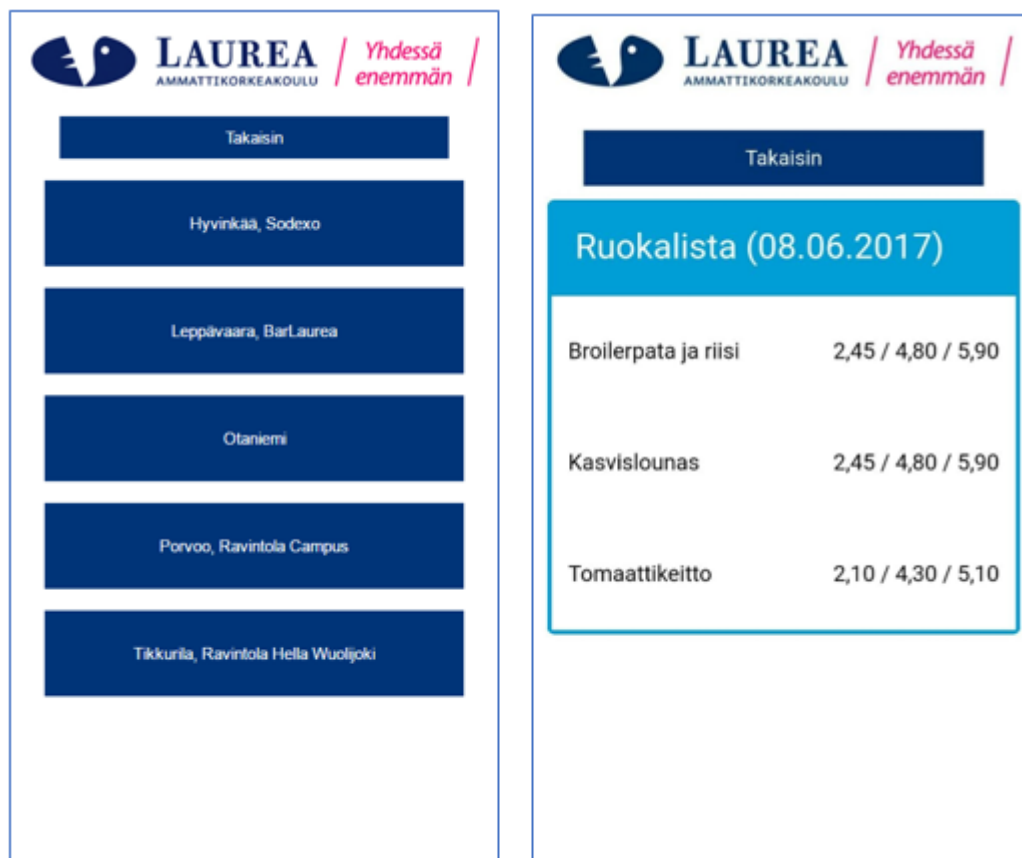
Jälki-ilmoittautuminen opintojaksolle
12.12.2016-1.1.2017

Opintojaksolle osallistumisen vahvistaminen
2.1.2017 alkavalla viikolla.

FUAS-kesäopinnot
15.5.-31.8.2017

Kuva 9: Lukuvuoden aikataulut

Ruokalistat -painikkeen takana on lista kaikista Laurean ruokaloista ja painamalla haluamaansa kampusta, pääsee käyttäjä tämän kampuksen ruokalistaan (kuva 11). Listan näyttötapa riippuu siitä onko kyseisellä ravintolla käytössään api. Jos api on käytössä, applikaatio tekee pyynnön apista kyseiselle päivälle. Applikaatio käsittelee vastauksena saadun json:in ja tulos näytetään käyttäjälle. Jos ravintolalla ei ole käytössään apia, applikaatio toimii samoin, kuin muiden päätoimintojen sivujen kohdalla: applikaatio käsittelee sivuilta saadun lähdekoodin ja tulos muutetaan mobiililaitteelle sopivaksi. Jos applikaatio ei saa oikeuksia käyttää lähdekoodia, sivu avataan mobiililaitteessa selaimen.



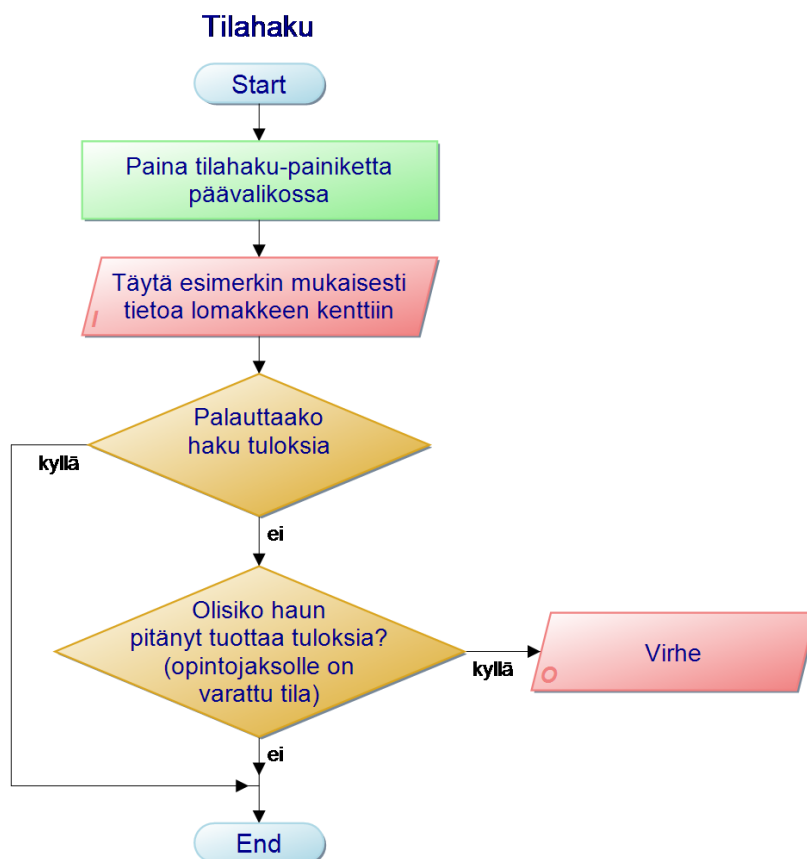
Kuva 10: Ruokalistat

5.4 Testaus

Testauksella varmistetaan, että kaikki sovelluksen ominaisuudet toimivat suunnitellun mukaisesti ja sovellus täyttää asiakkaan tarpeet. Testausmallit voidaan jakaa staattisiin ja dynaamisiin malleihin. Staattisessa testausmenetelmässä sovellusta ei varsinaisesti käytetä lainkaan vaan tutkitaan arkkitehtuurisuunnitelun ja koodiarviointien näkökulmasta. Dynaaminen testaus puolestaan on staattisen testauksen vastakohta. Sovellusta käytetään ja sen reaktioita seurataan. Tässä opinnäytetyössä käytetään testaukseen dynaamista testausmenetelmää ja tarkemmin mustalaatikkotestausta. (Kasurinen 2015)

Mustalaatikkotestaus on testauksen perinteisin muoto. Testaus tapahtuu siten, että sovellukselle annetaan syötteitä ja testaaaja seuraa miten ohjelma reagoi tähän, tarkastelematta mitä ohjelman sisällä tapahtuu. Testaaaja varmistaa, että syntyvä lopputulos vastaa haluttua lopputulosta ja raportoi virhetilanteissa sovelluksen ohjelmoinnista vastaavalle henkilölle. Testaus voidaan suorittaa esimerkiksi kokeilemalla erilaisia käyttötapauksia. Lisäksi voidaan tarkastaa järjestelmän tapa reagoida virheisiin tai käyttäjän antamiin haitallisiin syötteisiin. (Kasurinen 2015) Tässä tapauksessa testaaajalle annettiin alemman kuvan mukaisia käyttötapauksia. Virhetilanteeseen päädyttyä testaaaja raportoi

virheen ohjelmoijalle, joka teki tarvittavat muutokset applikaatioon ja käyttötapaus toistettiin, kunnes sen läpikäyminen suoritui ilman virheitä.



Kuva 11: Käyttötapaus

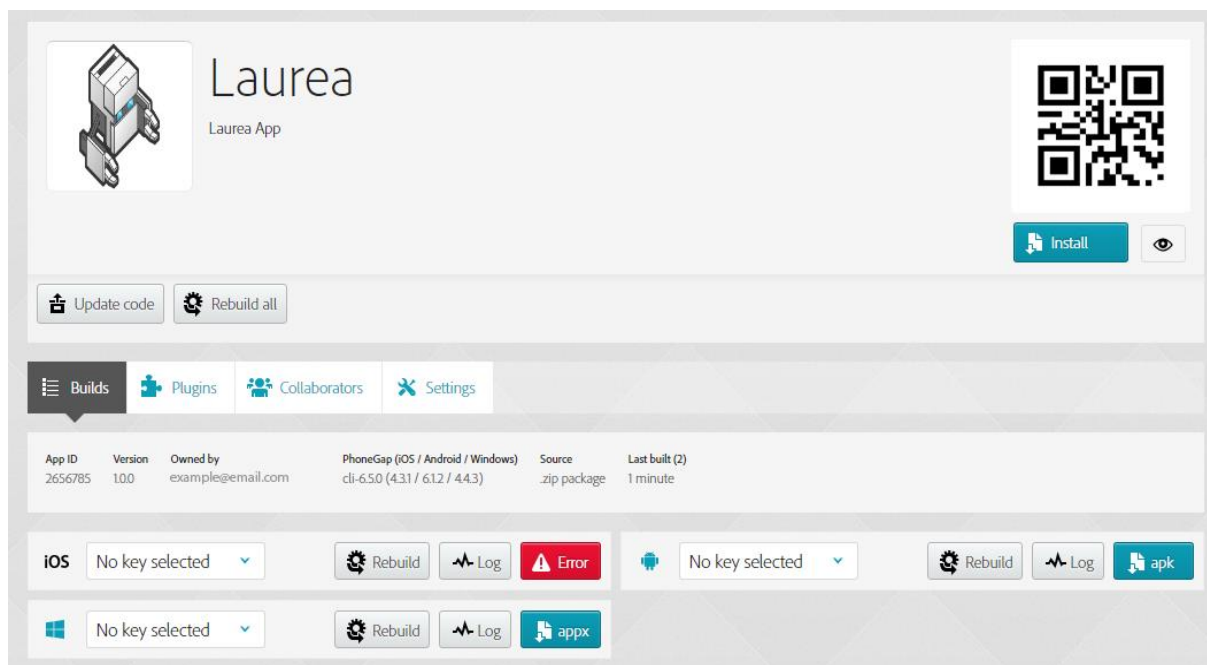
Applikaatio ohjelmoitiin ensin Android-käyttöjärjestelmälle ja sitä testattiin kunnes kaikki virheet oli saatu karsittua. Tämän jälkeen sama testausprosessi suoritettiin iOS- ja Windows 8-natiivilaitteella, joihin oli asennettuna Phoneyap developer tool.

5.5 Julkaisu

Kun applikaatio on valmis, se halutaan ottaa käyttöön. Tähän on kaksi tapaa: se voidaan manuaalisesti siirtää puhelimeen tai jakaa kauppapaikalla. Molemmissa tapauksissa koodi siirretään ensin Phoneyap Buildiin.

Phoneyap Build on Adoben pilvessä toimiva paketointipalvelu, joka pakatoi koodin valmiiksi applikaatioksi. Yksityiseen käyttöön applikaatioita voidaan tehdä yksi kappale ilmaiseksi. Maksullinen ohjelma kustantaa tekijälle 9,99 euroa kuukaudessa ja oikeuttaa julkaisemaan 25 applikaatiota. Kun Build on kääntänyt koodin sopivaksi kaikille järjestelmille, se voidaan

laittaa yleiseen jakeluun kauppapaikalle tai asentaa manuaalisesti puhelimeen lataamalla kullekin järjestelmälle generoidut tiedostot. Nopeampi vaihtoehto ladata tiedostot, on skannata puhelimella Phonegapin luoma QR-koodi, joka tuo tarvittavat tiedostot suoraan puhelimeen.



Kuva 12: Phonegap Build

Applikaation tekemiseen iOS:ille tarvitaan sertifikaatit. Nämä on mahdollista saada vain, jos sitoutuu Applen omaan kehittäjäohjelmaan, joka maksaa 99 euroa vuodessa. Tämä oikeuttaa kehittämään applikaatioita ja lisäämään ne applen omalle kauppapaikalle. Ilman kehittäjäohjelmaa Phonegap ei saa tarvittavia tiedostoja, jotta applikaatiota iOS:ille voitaisiin luoda ja tuloksena on virheilmoitus (kuva 13).

5.6 Päivittäminen

Ohjelmistokehityksen viimeinen vaihe on ylläpito. Se on vaihe, joka jatkuu koko sovelluksen elinkaaren ajan. Phonegap Build sisältää myös työkalut ylläpitoon. Jos applikaatioon halutaan tehdä muutoksia, voidaan uusi koodi ladata Buildiin painamalla "update code"-näppäintä. Tämä päivittää automaattisesti uuden version käyttäjien puhelimiin. Laureaan ollaan suunnittelemassa Peppijärjestelmää, joka korvaa mm. Asion, jossa nykyinen tilahaku on. Peppi käyttää toteuksessaan API:a. Tämä tarkoittaa sitä muutosten tekeminen applikaatioon uudessa järjestelmässä on helpompaa kuin nyt. Tällä hetkellä API on käytössä vain muutamassa Laurean ravintolassa. Muissa tapauksissa applikaation täytyy lähettää pyyntö sivulle ja käsitellä vastauksena saatu kokonainen lähdekoodi ja tehdä siihen muutoksia.

Validilla API-avaimella, kehittäjä saa kuitenkin JSON-formaatilla vastauksena listan kaikista varauksista; tilaisuuksista ja luokista. API:n kutsumiseen käytetään Phonegapin tukemaan Javascript-ohjelmointikieltä. Peppi on käytössä mm. Tampereen ammattikorkeakoulussa, jonka sivuilta löytyy yksityiskohtaiset ohjeet API:n käyttöön (Varaukset) .

6 Yhteenveto ja loppupäätelmät

Opinnäytetyön tavoitteena oli kehittää Laurean opiskelijoille applikaatio, joka sisältää opiskelijoille oleellisia toimintoja helpottamaan kouluarkea. Applikaation kehitys aloitettiin määrittelyllä yhdessä toimeksiantajan kanssa. Määrittelyn pohjalta tehtiin rautalankamalli ja prototyyppi. Prototyyppi esiteltiin toimeksiantajalle ja saadun palautteen perusteella, sitä lähdettiin kehittämään lopulliseen muotoonsa.

Applikaatio toteutettiin käyttäen Phonegapia, joka on tehokas ja helppokäyttöinen työkalu kun halutaan tehdä applikaatioita useammalle kuin yhdelle käyttöjärjestelmälle. Ohjelmointi toteutetaan käyttäjän omassa ohjelmointiympäristössä käyttäen yleisiä webkieliä: HTML, CSS; Javascript. Kun applikaatio on valmis se ladataan Phonegap Buildiin, joka paketoi lopullisen koodin ja tekee sen yhteensopivaksi kaikille käyttöjärjestelmille.

Lopullisena vaiheena oli applikaation testaus kaikilla käyttöjärjestelmillä. Applikaation saaminen toimimaan kaikilla kolmella järjestelmällä ei ollut täysin ongelmaton. Applikaatio tehtiin aluksi Androidille, jonka jälkeen sitä testattiin iOS:illa ja Windowsilla. Se ei toiminut aluksi kummallakaan käyttöjärjestelmällä johtuen suurimmaksi osaksi siitä, ettei tietokantaan saatu yhteyttä ja siksi kielen vaihtaminen applikaatiossa ei onnistunut. Koodiin tehtiin muutoksia kunnes järjestelmä saatiin toimimaan. Windows-puhelimen kohdalla huomattiin ettei SQL-lite tue enää Windowsia, joten kielivalinta täytyi tallentaa local storageen. Tämän korjaaminen onnistui niin, että applikaatio tunnistaa avautuessaan ensin, mikä järjestelmä käyttäjän laitteessa on ja jos käyttäjällä on Windows-puhelin, applikaatio tallentaa valinnan local storageen. Testaukseen ja virheettömyyden varmistamiseen kului lopulta suurin osa kehittämiseen kuluneesta ajasta. Projektisuunnitelmaa tehtäessä olisi tullut miettiä tarkemmin projektin aikataulua ja eri vaiheisiin käytettävää aikaa.

Opinnäytetyön tuloksena oli applikaatio, jolla käyttäjä pystyy löytämään, missä luokkatilassa oppitunti pidetään, tarkastelemaan lukuvuoden tärkeitä päivämääriä ja selaamaan kunkin yksikön ruokalistoja. Applikaatio on saatavilla sekä suomeksi että englanniksi ja se toimii kaikilla kolmella yleisimmällä käyttöjärjestelmällä: Androidilla, iOS:illa ja Windowsilla. Lopullinen applikaatio onnistui vaatimusten mukaisesti ja toimeksiantaja on hyväksynyt sen. Applikaatio otetaan käyttöön opiskelijoille syksyllä 2017.

Lähteet

Kirjalliset lähteet

Ghatol, R., Patel, Y. 2012. Beginning PhoneGap. Apress

Haikala, I., Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt. Talentum

Kasurinen, J. 2015 Ohjelmistotestauksen käsikirja. Docendo

McWheter, J. 2012 Professional Mobile Application Development.

Sähköiset lähteet

Adobe Phonegap 2016 Viitattu 21.05.2017

<http://phonegap.com/products/>

Appery. 2017 Viitattu 03.08.2017

<https://appery.io/pricing>

Cordova Plugins 2015 Viitattu 11.02.2017

<https://cordova.apache.org/plugins/>

Getting started. Adobe 2015 Viitattu 11.02.2017

<http://docs.phonegap.com/phonegap-build/faq/#what-is-build>

Gibbs, S. 2016. Mobile web browsing overtakes desktop for the first time. The Guardian. Viitattu 10.02.2017

<https://www.theguardian.com/technology/2016/nov/02/mobile-web-browsing-desktop-smartphones-tablets>

Kansainvälisyys opinnoissa. Laurea. Viitattu 30.05.2017

<https://www.laurea.fi/opiskelu-ja-hakeminen/opintojen-kulku/kansainvalisyys-opinnoissa>

Kolme tapaa kehittää mobiilisovellus. W3 2014 Viitattu 03.08.2017

<https://w3.fi/kolme-tapaa-kehittaa-mobiilisovellus/>

Järjestelmät Laureassa. Viitattu 03.08.2017

https://www.laurea.fi/dokumentit/Documents/Opiskelijan%20ohje%20Laurean%20j%C3%A4rjestelmiin_SU.pdf

Phonegap Build 2013 Viitattu 07.06.2017

<https://build.phonegap.com/>

Smartphone market 2016 Viitattu 03.08.2017

<http://www.idc.com/promo/smartphone-market-share/os>

Varaukset. TAMK. Viitattu 08.06.2017

<http://avoindata.tamk.fi/en/rajapinnat/varaukset/>

W3S 2017 Viitattu 11.02.2017

<https://www.w3schools.com/>

Kuviot

Kuva 1: Flurry Analytics.....	6
Kuva 2: Phonegap Build	10
Kuva 3: Phonegap työpöytäsovellus	12
Kuva 4: Phonegap projekti	13
Kuva 5: Rautalankamalli	15
Kuva 6: Kielenvalinta.....	16
Kuva 7: Etusivu	17
Kuva 8: Tilahaku	18
Kuva 9: Lukuvuoden aikataulut.....	19
Kuva 10: Ruokalistat.....	20
Kuva 11: Käyttötapaus	21
Kuva 12: Phonegap Build	22

Taulukot

Taulukko 1: Kehittämiskielet ja alustat	10
Taulukko 2: Vaatimukset	14